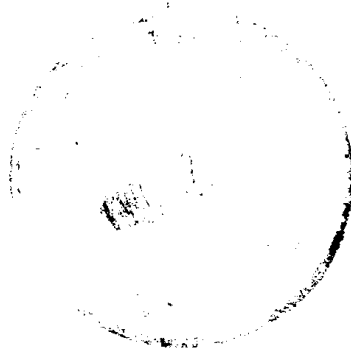


73.82
5297

ARTIFICIAL INTELLIGENCE

An Engineering Approach

Robert J. Schalkoff
Clemson University



9250119

McGraw-Hill Publishing Company

New York St. Louis San Francisco Auckland Bogotá Caracas Hamburg
Lisbon London Madrid Mexico Milan Montreal New Delhi Oklahoma City
Paris San Juan São Paulo Singapore Sydney Tokyo Toronto

9250119

CREDITS

- Quote, p. 1: From THE LORAX, by Dr. Seuss. Copyright ©1971 by Dr. Seuss and A. S. Geisel. Reprinted by permission of Random House, Inc.
- Fig. 7.2: Roads, C., *Foundations of Computer Music*, MIT Press, copyright 1979.
- Fig. 13.2, 13.3: Harmon, P., and King, D., *Expert Systems*, copyright 1985, John Wiley and Sons, reprinted by permission.
- Fig. 13.4: Barker, V., and O'Connor, D., "Expert Systems for Configuration at Digital. XCON and Beyond," *Comm. of the ACM*, March 1989.
- Fig. 13.15: Hayes-Roth, F., "Rule-Based Systems," in *Encyclopedia of Artificial Intelligence* (Shapiro, S., ed.), copyright 1987, John Wiley and Sons, reprinted by permission.
- Figs. 13.16, 13.17: copyright 1984, IEEE.
- Figs. 13.19, 13.20, 16.10: copyright 1986, Academic Press.
- Fig. 15.1: copyright 1983, SIAM.
- Fig. 16.3: copyright 1987, IEEE.
- Fig. 16.12: Clancy, W. J., "From Guidon to Neomycin," *AI Magazine*, August 1986, page 53, copyright 1986, American Association for Artificial Intelligence.
- Fig. 18.2: Carbonell, J., and Langley, P., "Machine Learning," in *Encyclopedia of Artificial Intelligence* (Shapiro, S., ed.), copyright 1987, John Wiley and Sons, reprinted by permission.
- Fig. 19.1: copyright 1987, IEEE.

ARTIFICIAL INTELLIGENCE An Engineering Approach

Copyright ©1990 by McGraw-Hill, Inc. All rights reserved.
Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

2 3 4 5 6 7 8 9 0 DOC DOC 9 5 4 3 2 1 0

ISBN 0-07-055084-0

This book was set in Times Roman by Publication Services.
The editor was David M. Shapiro;
the production supervisor was Denise L. Puryear.
The cover was designed by Albert M. Cetta.
Project supervision was done by Publication Services.
R. R. Donnelley & Sons Company was printer and binder.

Library of Congress Cataloging-in-Publication Data

Schalkoff, Robert J.
Artificial intelligence: an engineering approach / Robert J. Schalkoff
p. cm.
Includes bibliographical references.
ISBN 0-07-055084-0
1. Artificial intelligence 2. Artificial intelligence--Data processing. 3. Prolog (Computer program language) I. Title.
Q335.S37 1990
006.3--dc20 89-13682

Also Available from McGraw-Hill

Schaum's Outline Series in Computers

Most outlines include basic theory, definitions, and hundreds of solved problems and supplementary problems with answers.

Titles on the Current List Include

Advanced Structured Cobol
Boolean Algebra
Computer Graphics
Computer Science
Computers and Business
Computers and Programming
Data Processing
Data Structures
Digital Principles, 2d Edition
Discrete Mathematics
Essential Computer Mathematics
Mathematical Handbook of Formulas & Tables
Matrix Operations
Microprocessor Fundamentals, 2d Edition
Programming with Advanced Structured Cobol
Programming with Assembly Language
Programming with Basic, 3d edition
Programming with C
Programming with Fortran
Programming with Pascal
Programming with Structured Cobol

0748/30

Schaum's Solved Problems Books

Each title in this series is a complete and expert source of solved problems containing thousands of problems with marked out solutions.

Related Titles on the Current List Include

3000 Solved Problems in Calculus
2500 Solved Problems in Differential Equations
3000 Solved Problems in Linear Algebra
2000 Solved Problems in Numerical Analysis

Available at your college bookstore. A complete listing of Schaum titles may be obtained by writing to: Schaum Division
McGraw-Hill Publishing Co.
Princeton Road, S-1
Hightstown, NJ 08520

ABOUT THE AUTHOR

Robert J. Schalkoff received the Ph.D. degree in Electrical Engineering from the University of Virginia. Currently, he is a Professor of Electrical and Computer Engineering at Clemson University in Clemson, South Carolina. His primary research interests are in artificial intelligence and computer vision with special emphasis on model-based image understanding, motion and stereo vision, and associated computer architectures. He is also the author of *Digital Image Processing and Computer Vision* (John Wiley and Sons, 1989).

PREFACE

"All men by nature desire knowledge."

Metaphysics, bk. I, ch. I

ARISTOTLE

384–322 B.C.

This book explores the theoretical and practical underpinnings of Artificial Intelligence (AI) and provides an introductory-level "hands-on," engineering-oriented perspective on AI technology. No previous experience in this area is assumed, although readers are expected to have minimal familiarity with discrete mathematics and some exposure to modern programming languages, such as Pascal. Since this is a teaching text dedicated to those new to the field, efforts have been made to provide the reader a "soft start."

Artificial Intelligence may be the Rodney Dangerfield of scientific disciplines. Only recently has AI begun to receive serious, widespread attention (respect?). This is due partially to the fact that many visible and important AI successes exist. These include expert systems, natural-language and speech-understanding systems, and computer vision systems. Although many issues in AI are still "open" or unresolved, most optimistic observers predict new successes are just beyond the horizon. Indeed, the expectations from Artificial Intelligence may be at an all-time peak. Perhaps we should modify the common expression "knowledge is power" to "representation and autonomous manipulation of knowledge is *real* power."

The book presents topics that are generally considered "core" AI concepts; it does not attempt to be an encyclopedia or bibliographic summary of all AI-related research work ever produced. Emphasis on the conceptual approach explains the organization of the text (i.e., the partitioning of topics). In addition, the book attempts to make the important distinctions between theory and practice that are often major factors in the choice and implementation of AI algorithms.

The area of Artificial Intelligence is becoming more important in both undergraduate and graduate curricula in Computer Science and Engineering. In fact, it represents a major elective area in the IEEE Model Curriculum. There is little doubt

that AI techniques will become more commonplace in other engineering and scientific disciplines. In this text, extensive use of detailed PROLOG and LISP examples of AI implementation allows the design of an AI system to be treated with an engineering approach and in a "symbolically quantitative" manner. In addition, the AI theory presented is merged with practical aspects of software engineering.

Although the primary objective of the text is to provide a teaching tool, it is quite likely that practicing engineers and scientists will also find the clear, concept-based treatment useful in updating their backgrounds. Hopefully, this will provide the fundamental conceptual and algorithmic tools necessary to confront the industrial application-driven challenges of this rapidly expanding and emerging field.

Here are two things a reader should consider before entering into a study of AI.

1. AI embodies an *inexact form of computation*, which appears, at best, to be based on an *ambiguous model* (similar to human reasoning). This contrasts with numerical computation, where algorithm behavior typically is explicitly developed a priori and (numerical) answers are known to many decimal places. Some aspects of AI are therefore close to "exploratory programming."
2. AI is a *rapidly maturing discipline* with a strong *interdisciplinary nature*. Algorithms and data structures that were in vogue yesterday are now considered obsolete. Today's algorithms are under scrutiny, and tomorrow's algorithms are on the horizon.

In research, it has been found that a theoretical understanding of AI concepts (e.g., unification, first-order predicate logic), while desirable, is not sufficient. A capability to develop, analyze, and implement knowledge databases; design, analyze, and modify inference engines; develop large-scale systems; and interface AI systems to the "real," or outside, world is also necessary. As a consequence, both theoretical and practical aspects are explored in this text using straightforward examples in the LISP and PROLOG programming languages. The Clocksin and Mellish syntax is used for PROLOG. The LISP constructs used are those typically found in a "pure" LISP dialect. For the most part, they are a subset of those available in Franz or Common LISP. At this point in history, it is *not clear whether PROLOG or LISP will become the AI language of choice*; each has its strengths, weaknesses, and preferred applications. Since both languages are currently enjoying popularity, and often the representation or manipulation desired is more naturally suited for one, both are included in this book. The use of these languages as an integral part of the presentation helps make illustration of concepts unambiguous and concise.

It is possible to use this text while de-emphasizing programming aspects. In the author's opinion, however, the hands-on, or practical, experience gained by developing and modifying working systems in AI languages is an invaluable aspect of any AI course. Learning the jargon of AI is one thing; development of an ability to appreciate and experiment with the underlying concepts is a higher goal. Many of the ideas in AI are in fact fully understood only when the complexities and limitations of underlying data structures and inference mechanisms themselves are familiar to the student through hands-on experience. Table P.1 summarizes options for the use of this text in courses with differing orientations.

Clearly, there is more material than can be covered in one semester--the choice of specific topics is left to individual instructors. Given students with significant logic, LISP, and PROLOG background, it is possible to cover many of the later chapters on advanced topics (e.g., learning, temporal reasoning, architectures) in a single semester. Alternatively, Chapters 1 through 13 provide the basis for a comprehensive one-semester introduction, which then could be followed by a second course based Chapters 14 through 19, supplemented with readings from the current AI literature and possibly a comprehensive project. Many of the problems at the end of each chapter, combined with the implementation examples in the text, serve as good starting points for projects or extended self-study.

TABLE P.1
Suggested use of text in courses with different orientations/objectives

Orientation/Objective	Chapters/Sequence
2-semester AI course (with or without projects)	Ch. 1-13 (first semester) Ch. 13-19 (second semester) Supplement with outside reading/programming/ project
1-semester course with programming emphasis	Ch. 1-5, 7-12; selected portions of Ch. 13, 14
1-semester course without programming emphasis	Ch. 1, 2, 4 (lightly), 5, 8 (lightly), 11-14 (without LISP examples)
1-semester course with project	Ch. 1-5, 7-12; selected portions of Ch. 13, 14, with projects chosen from Ch. 13-19
1-semester course with PROLOG and LISP programming background required	Ch. 1, 2, 5, 7, 11-15; selected portions of Ch. 16-19

It is difficult to specify the intended audience of this text on the basis of the descriptors "graduate" or "undergraduate." It is accurate to term the material "introductory," that is, suitable for a first course in AI. This first exposure may occur anywhere from the junior level undergraduate to first year graduate level.

The book is structured in the following order:

1. Introduction to logic, representation, and AI languages.
2. AI concepts and examples in PROLOG.
3. AI concepts and examples in LISP.
4. Advanced AI concepts.

This structure has suited teaching needs well, since some underlying theory may be presented while the students are becoming familiar with the particular PROLOG or LISP development systems to be used. Additionally, the first examples are shown in PROLOG since this language has a built-in unification mechanism; this allows concentration on *problem specification* or representation, without worrying about the solution procedure. It is also a natural extension of the logic-based representation considered early in Chapter 2. Having developed some dexterity in setting up AI

(symbolic manipulation) problems, students are then ready to tackle LISP and explicitly develop their own representations and solution procedures.

There is no single, unique, or generally agreed upon set of algorithms for AI. In this book many of the algorithms shown or developed are shown in *skeletal form*. They constitute a general approach, and possible variants, options, or extensions are indicated.

This book reflects the contribution of numerous individuals. Comments and questions from many students led to subsequent revisions that improved overall quality. The efforts of students in solving early versions of the problems and debugging and revising LISP and PROLOG code are also reflected.

An accompanying instructor's manual is available from the publisher. It contains problem solutions as well as IBM-format diskettes containing all the PROLOG and LISP examples shown in the text. Both Common and Franz LISP versions are provided.

REFERENCES

- Leitch, R. R., "Artificial Intelligence: The Emergence of a Formal Basis for Engineering," *IEE Proc. (Part D: Control Theory and Application)*, Institution of Electrical Engineers (London), Vol. 132, No. 4, July 1987, p. 216.
- "Model Program in Computer Science and Engineering" (Model Program Committee, Educational Activities Board, IEEE), *IEEE Computer Society*, December 1983.

ACKNOWLEDGMENTS

I would like to thank my wife Leslie for her support throughout this project. The quality and accuracy of the text reflects her careful word processing, editing, and proofreading assistance.

McGraw-Hill and the author would like to thank the following reviewers for their many helpful comments and suggestions: Seth Hutchinson, Purdue University; Robert McCoard, California State University-Northridge; Dennis E. Ray, Old Dominion University; Kathleen Swigger, North Texas State University.

Robert J. Schalkoff

"It is impossible for a man who takes a survey of what is already known, not to see what an immensity in every branch of science yet remains to be discovered."

THOMAS JEFFERSON

Monticello, June 18, 1799

CONTENTS

Preface	xxi
1 Introduction	1
Background and History	1
What is AI?	2
Emulating Human Cognitive Processes	6
Why Is AI Difficult?	7
Central Themes in AI	8
Overview of Concepts	8
To Search or Not to Search (The Knowledge-Search Tradeoff)	9
Mathematical Preliminaries	10
Stored Knowledge, Descriptions, and Relations	10
Semantic Nets	13
Representations That Attempt to Convey "Meaning"	16
The Role of Knowledge Representations and Computational Models in AI	17
Why Models?	17
Representation Characteristics	22
An Abstract View of Modeling	23
The Frame Problem in AI	24
Elementary Knowledge Representation Approaches	26
Procedural versus Declarative Schemes	26
Selection of Representation Primitives	27
State Space Representations and Control	29
Characteristics of AI Algorithms	31
Topical Outline	31
Ancillary References	32
References	33
Problems	34

2	Symbolic (Computational) Logic	37
	Introduction	37
	The Use of Logic in AI	38
	Relation to "Classical" Logics and Extensions	38
	Inference	38
	The Vocabulary of Logic	39
	The Mathematics of Logic	39
	Statements	39
	Analysis of Compound Statements Using Simple Logical Connectives	41
	Implication	41
	A Further Look at Implication	43
	Deduction Using Propositional Logic	45
	Predicate Logic	48
	Introduction	48
	Predicates and Variables	48
	Quantified Predicates, Equivalences, and Manipulations	50
	The Language of First Order Logic	53
	Manipulation of WFFS	55
	Resolution (Proof by Refutation)	55
	Unification	57
	Extending Modus Ponens to Statements with Variables (and Unification)	63
	Summary and Limitations of Logic-Based Representations	65
	References	66
	Problems	66
 3	 AI Programming Languages and Development Systems—"The Tools of the Trade"	 73
	Programming Languages Overview	73
	Languages for AI Implementation	74
	"Conventional" and Symbolic Computing	74
	Coupled Computing Systems	75
	Manipulable Representations and Data Structures	76
	The LISP and PROLOG Distinction	77
	The Distinction (What's Different?)	77
	The PROLOG Programming Concept as Description	78
	The LISP Programming Concept	78
	Data Directed Processing	79
	AI Development Environments and Systems	79
	Desirable AI Software Capabilities	79
	Desirable AI Software Environments	80
	Programming Language Classifications	81
	Introduction and Definitions	81
	Examples of Pascal, LISP and PROLOG	82
	Control Structures	84
	Alternative Languages	85
	Recursion	85
	Implementation of Recursion	86
	Tail Recursion	87

Variables and Scoping (Programming Point of View)	88
Variable Scope	88
Binding	88
Pascal Example	89
LISP and PROLOG Variable Scoping	89
The Binding Environment	90
Dynamic and Lexical Scoping of LISP Variables	90
References	90
4 Fundamentals of PROLOG	91
PROLOG Background	91
Introduction	91
Unification—the Basis of PROLOG	92
PROLOG “Standards,” Compatibility and C + M	92
Logical Connections and PROLOG Syntax	93
Clauses, Facts, Goals, and Rules	93
Clauses	93
Facts	93
Goals	94
Rules	94
PROLOG and Modus Ponens	96
More Comprehensive Examples	96
The Syntax and Characteristics of PROLOG	98
PROLOG Syntax Basics	99
Lists in PROLOG	101
Goals and the PROLOG Unification Mechanism	103
Potential Unification Circularities with Symmetric and Transitive Relations	107
Constraints Using Variables in PROLOG	109
Creating, Modifying and Examining PROLOG Databases (Database Management)	109
Consult	109
Reconsult	111
Retract and Listing	112
A Detailed Look at the Unification Process and Search	112
References	115
Problems	115
5 Constraint Satisfaction Problems and PROLOG	
Application to Image Labeling	124
Constraint Satisfaction Problems	124
Constraint Satisfaction Problem Formulations	125
Relation to Numerical Constraint Satisfaction Problems	125
Interpretation as Search	126
Region Labeling and Segmentation in Image Processing	127
General Labeling Formulation	127
Image Labeling Example	128
Constraint Determination	129
PROLOG Implementation of Constraint Satisfaction	130
The Positive Approach	130
The Negative Approach (Coding What Is Not Allowed in PROLOG)	135

References	139
Problems	139
6 An AI Application Using PROLOG: Electric Power System Protection Analysis	144
Overview	144
Objectives	144
Simple Power Protection Background	145
Problem Formulation	146
Overall Goals	146
PROLOG Description of System Topology	147
Breaker Backup	148
An Extended Example of PROLOG Description Derivation	148
General Network Topology Description	149
Significant Rules in Description	151
Connectivity Rules	151
Locating Corresponding Breakers via Rules	151
Generation Rules	151
Review of the PROLOG Cut Predicate	152
Backup Breaker Description	152
Determination of Fault Locations through Electrical Isolation	155
Rules for Interpretation and Display	155
Overall Goal (Incorporated in Database, for Simplicity)	156
Sample Results	158
Conclusions	158
References	159
Problems	160
7 PROLOG Application to Structural Object Description, Formal Languages, and Parsing	163
Introduction	163
Parsing and Applications	164
Compilation of High Level Languages	164
Understanding of Natural Language	165
Description and Interpretation of Image (Visual) Data	168
Syntactic Pattern Classification	168
Formal Language Theory	170
Formal Definitions	170
Example of a 2-D Language Application—A Line Drawing	
Description Grammar [Shaw 1970]	172
Sentence Generation and Parsing in PROLOG	174
PROLOG Sentence Generation	174
Context-Free Parsing In PROLOG	177
Graphical Constructs: The Derivation Tree	178
An Abstract View of the Parsing Problem	178
Top-Down versus Bottom-Up Parsing	179
Development of the G1 Parser	179
Parser/Generator Similarities	179
Desired Parser Input Structure	180

(Augmented) Transition Networks (ATN's)	185
Transition Networks	185
Augmented Transition Nets (ATNs)	188
Concluding Remarks	189
Parsing Computational Complexity	189
From Parsing to Meaning	190
References	190
Problems	191
 8 Fundamentals of LISP	 194
Introduction	194
Getting Started in LISP	194
Basic LISP Concepts and Examples (Eval, Lists, and Function Definitions)	196
The Eval Function	197
Elementary List Manipulation	197
Function Definition	200
(FranzLISP) Help	201
Simple LISP Assignment and I/O	201
LISP Syntax in General	203
The Basic Function Groups	204
Functions and Variables	204
Control Functions	205
List Manipulation (Construction) Functions (and Distinctions)	210
Properties and Property Lists	212
Scoping of LISP Variables	215
Extended Example of Recursion Using LISP	216
References	221
Problems	221
 9 List Structures—Rationale and Computer Representations	 228
Introduction	228
Why Graphs, Relations, and Lists?	229
Operations on Stored Representation(s) Desired	230
Trees	233
Computer Memory Model(s)	234
Computer Representation of Graphs and Relational Information	234
Conversion of Graphical Representations into Trees and Lists	236
The cons Cell and Building Lists	237
The cons Cell	238
Extensions Using the cons Cell	240
References	244
Problems	244
 10 Matching and Unification in LISP—A Prelude to Rule-based Inference	 248
Introduction	248
The Need for Matching	248
A Hierarchy of Matching Problems	249

The Simplest Matcher in LISP	250
Matching with "Don't Cares"	252
Matching with Variables and Bindings	255
Limited Matching Using <code>match-var</code> without Checking Prior Bindings	258
Matching with Checking and Adding of Bindings	260
The General Unifier	266
Matching of Lists with Lists of Lists	271
Constants Only (Case 4)	271
List Matching with Variables	272
Summary	274
References	275
Problems	275
 11 Rule-based Inference Systems Part I—Production Systems, Forward Chaining, and LISP Implementations	 279
Production Systems	279
Overview and Basic Structure	279
Features of Rule-based Production Systems	282
Theoretical Aspects of the Production System Structure and Inference Strategy	283
Significant Aspects of the Inference Engine (Inference Control Strategies)	283
Problem Solving, the State Space, and the Inference Engine	284
Production System Properties	284
Rule-based Chaining Inference Approaches	287
Rule-based Systems as Production Systems and Control Paradigms	287
Review of the Logical Basis for Forward Chaining	288
Implementation of a Simple Forward Chaining System in LISP	289
Overall System Structure	289
Database Structure	290
Rule Representation and Rulebase Structure	291
Forward Chaining Inference Engine Main Function "Chainforward"	295
Examples	296
Image Motion Description Example	300
Extensions to the Simple Forward Chaining Inference System	303
Rule Selection, Specificity and the Conflict Set	304
Forward Chaining with Variables in Rules	308
Examples of Conflict Resolution Strategies and Resulting Rule Firing Sequences	314
Summary	319
References	319
Problems	319
 12 Rule-based Inference Systems Part II—Backward Chaining with LISP Implementation	 325
Backward Chaining	325
The Basis for Backward Chaining	327
Hypothesis Generation and Application to Intelligent Systems	330
Abduction	331

LISP Implementations of Backward Chaining	334
Derivation of the Inference Engine	334
LISP Implementation of Basic Backward Chaining Algorithm	334
Extensions to the Backward Chaining Algorithm	342
Backward Chaining with Variables in Rules (Optional)	348
Backward versus Forward Chaining (Choosing "Directions")	352
References	353
Problems	354
 13 Inference Systems—A Further Look	 359
Expert Systems	359
Introduction and Examples	359
Expert System Limitations	363
Expert System Development	364
Expert Systems Summary	366
Reasoning with Uncertainty	367
Incorporating Uncertainty in the Inference Process	368
Types of Uncertainties and Their Propagation	369
Probability Review	372
Simple Example of Two-Antecedent Rule Using Probability	374
Probabilistic Extensions (Optional)	376
Heuristic Approaches Derived from Probabilistic Reasoning (Confidence Factors)	378
Many Valued Logics	381
LISP Implementations	386
Example of Numerical Confidence Propagation and Conflict Resolution in Forward Chaining	387
A Commercial Rule-based System Example: OPS5	390
The OPS5 Production System Programming Language	390
OPS5 Program Structure	392
Conflict Resolution in OPS5	395
OPS5 Examples	395
Additional Inference Approaches and Applications	406
Approaches Based upon Model Unification	406
Using a Forward Chaining System for Backward Chaining (Optional)	411
References	415
Problems	417
 14 Planning in AI Systems	 424
The Significance of Planning	424
Background	424
Planning Examples	425
Representational Issues in Planning	426
Representation, Operator Specification, and Search	428
Developing a Representation	428
Search and Potential Computational Complexity	429
Operators and Representations	430
Planning Difficulties (Local Actions and Global Objectives)	431

Planning Related to a Production System	432
The Non-Commutative Nature of Planning Systems	432
Plan Generation as a Control Problem	432
Blocks-World System Representation	434
Problem Specification, Assumptions and Operator Representations	434
Plan Generation Algorithms	436
Cycles in Plans	436
Forward State Propagation Using Generate-and-Test	436
Backward State Propagation	437
Advanced Planning Strategies	444
Generation of Expertise (Remembering a Plan)	444
The Triangle Table	445
Example of Plan and Corresponding Triangle Table Formation	446
LISP Implementation of Plan Generators	448
Forward State Propagation Using Generate-and-Test (Checkerboard)	448
Incorporating "Means-End" in the Operator Selection (Search) Strategy	463
Operator (and State) Evaluation Functions and Problem-Solving Heuristics	463
Forward State Propagation (Blocks-World) LISP Implementation	
Using Action Ordering (Means-End)	464
LISP Implementation of Backward State Propagation	476
Planning Extensions	481
Hierarchical Planning and Macro-Operators (Considering More Than One State Change)	481
Planning in a Changing Environment	481
Simultaneous Application of Operators and Parallel Planning	482
References	486
Problems	486
 15 Search and Computational Complexity in AI Systems	 493
Computational Complexity	493
Quantifying Computational Complexity	494
Search in AI Problems	496
A Formal Representation for Search Problems	496
Problem Decomposition and Searching with AND-OR Trees	499
Incorporating Search in AI	502
Search Computational Complexity and Trade-offs	503
Search Algorithms	505
Trial-and-Error Approaches	505
Breadth-First Search	506
Depth-First Search	507
Forward-Backward (Bidirectional) Search	507
Other Search Approaches (Including Heuristic Search Techniques)	507
References	508
Problems	509