

Computational Intelligence

A Logical Approach

David Poole
Alan Mackworth
Randy Goebel

New York Oxford
Oxford University Press
1998

Oxford University Press
Oxford New York
Athens Auckland Bancock Bogotá Bombay Buenos Aires
Calcutta Cape Town Dar es Salaam Delhi Florence Hong Kong
Istanbul Karachi Kuala Lumpur Madras Madrid Melbourne
Mexico City Nairobi Paris Singapore Taipei Tokyo Toronto Warsaw
and associated companies in
Berlin Ibadan

Copyright © 1998 by Oxford University Press, Inc.

Published by Oxford University Press, Inc.,
198 Madison Avenue, New York, New York, 10016
<http://www.oup-usa.org>
1-800-334-4249

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of Oxford University Press

The authors and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research and testing of the theories and programs to determine their effectiveness. The authors and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

Library of Congress Cataloging-in-Publication Data

Poole, David (David Lynton), 1958–
Computational intelligence: a logical approach /
David Poole, Alan Mackworth, Randy Goebel.
p. cm.
Includes bibliographical references and index.
ISBN 0-19-510270-3 (cloth)
1. Artificial Intelligence,
I. Mackworth, Alan K. II. Goebel, Randy G. III. Title.
Q335.P657 1997
006.3–dc21

97-9075
CIP

Printing (last digit): 9 8 7 6 5 4 3 2 1
Printed in the United States of America
on acid-free paper

Contents

Preface	xv
1 Computational Intelligence and Knowledge	1
1.1 What Is Computational Intelligence?	1
1.2 Agents in the World	7
1.3 Representation and Reasoning	9
1.4 Applications	11
1.5 Overview	19
1.6 References and Further Reading	20
1.7 Exercises	21
2 A Representation and Reasoning System	23
2.1 Introduction	23
2.2 Representation and Reasoning Systems	23
2.3 Simplifying Assumptions of the Initial RRS	27
2.4 Datalog	29
2.5 Semantics	31
2.6 Questions and Answers	40
2.7 Proofs	46
2.8 Extending the Language with Function Symbols	58
2.9 References and Further Reading	63
2.10 Exercises	63
3 Using Definite Knowledge	69
3.1 Introduction	69
3.2 Case Study: House Wiring	70
3.3 Databases and Recursion	75
3.4 Verification and Limitations	79
3.5 Case Study: Representing Abstract Concepts	81
3.6 Case Study: Representing Regulatory Knowledge	86
3.7 Applications in Natural Language Processing	91
3.8 References and Further Reading	104

3.9	Exercises	104
4	Searching	113
4.1	Why Search?	113
4.2	Graph Searching	114
4.3	A Generic Searching Algorithm	119
4.4	Blind Search Strategies	125
4.5	Heuristic Search	132
4.6	Refinements to Search Strategies	138
4.7	Constraint Satisfaction Problems	147
4.8	References and Further Reading	163
4.9	Exercises	163
5	Representing Knowledge	169
5.1	Introduction	169
5.2	Defining a Solution	170
5.3	Choosing a Representation Language	174
5.4	Mapping from Problem to Representation	180
5.5	Choosing an Inference Procedure	192
5.6	References and Further Reading	195
5.7	Exercises	196
6	Knowledge Engineering	199
6.1	Introduction	199
6.2	Knowledge-Based System Architecture	200
6.3	Meta-Interpreters	202
6.4	Querying the User	212
6.5	Explanation	217
6.6	Debugging Knowledge Bases	221
6.7	A Meta-Interpreter with Search	226
6.8	Unification	230
6.9	References and Further Reading	233
6.10	Exercises	233
7	Beyond Definite Knowledge	235
7.1	Introduction	235
7.2	Equality	235
7.3	Integrity Constraints	241
7.4	Complete Knowledge Assumption	248
7.5	Disjunctive Knowledge	256
7.6	Explicit Quantification	268
7.7	First-Order Predicate Calculus	270

7.8	Modal Logic	275
7.9	References and Further Reading	277
7.10	Exercises	278
8	Actions and Planning	281
8.1	Introduction	281
8.2	Representations of Actions and Change	287
8.3	Reasoning with World Representations	298
8.4	References and Further Reading	315
8.5	Exercises	316
9	Assumption-Based Reasoning	319
9.1	Introduction	319
9.2	An Assumption-Based Reasoning Framework	321
9.3	Default Reasoning	323
9.4	Abduction	332
9.5	Evidential and Causal Reasoning	335
9.6	Algorithms for Assumption-Based Reasoning	339
9.7	References and Further Reading	342
9.8	Exercises	343
10	Using Uncertain Knowledge	345
10.1	Introduction	345
10.2	Probability	346
10.3	Independence Assumptions	361
10.4	Making Decisions Under Uncertainty	381
10.5	References and Further Reading	394
10.6	Exercises	395
11	Learning	397
11.1	Introduction	397
11.2	Learning as Choosing the Best Representation	403
11.3	Case-Based Reasoning	414
11.4	Learning as Refining the Hypothesis Space	416
11.5	Learning Under Uncertainty	424
11.6	Explanation-Based Learning	433
11.7	References and Further Reading	437
11.8	Exercises	438

12 Building Situated Robots	443
12.1 Introduction	443
12.2 Robotic Systems	444
12.3 The Agent Function	446
12.4 Designing Robots	447
12.5 Uses of Agent Models	449
12.6 Robot Architectures	450
12.7 Implementing a Controller	451
12.8 Robots Modeling the World	457
12.9 Reasoning in Situated Robots	458
12.10 References and Further Reading	459
12.11 Exercises	460
Appendix A Glossary	461
Appendix B The Prolog Programming Language	477
B.1 Introduction	477
B.2 Interacting with Prolog	478
B.3 Syntax	479
B.4 Arithmetic	481
B.5 Database Relations	483
B.6 Returning All Answers	485
B.7 Input and Output	487
B.8 Controlling Search	488
Appendix C Some More Implemented Systems	491
C.1 Bottom-Up Interpreters	491
C.2 Top-Down Interpreters	498
C.3 Constraint Satisfaction Problem Solver	507
C.4 Neural Network Learner	511
C.5 Partial-Order Planner	515
C.6 Implementing Belief Networks	521
C.7 Robot Controller	529
Bibliography	533
Index	549

List of Figures

1.1	An agent as a black box	8
1.2	An environment for the delivery robot	14
1.3	An electrical environment for the diagnostic assistant	16
2.1	The role of semantics in a representation and reasoning system . . .	26
2.2	Truth table defining \wedge and \leftarrow	35
2.3	Clauses provided by the user	44
2.4	Bottom-up proof procedure for computing consequences of KB . . .	47
2.5	Top-down definite clause interpreter, without variables	51
2.6	Top-down definite clause interpreter, with variables	56
3.1	An electrical environment with individuals named	71
3.2	Sample database	76
3.3	Example database of student record information	87
3.4	Example database relations about the university	88
3.5	A context-free grammar for a very restricted subset of English . . .	96
3.6	A simple dictionary	97
3.7	Grammar for output of canned English	99
3.8	A grammar that enforces number agreement and builds a parse tree .	100
3.9	A grammar that constructs a query	102
3.10	A dictionary for constructing a query	103
4.1	The delivery robot domain with interesting locations labeled	115
4.2	A graph to search for the delivery robot domain	117
4.3	A search graph for an SLD derivation	118
4.4	Problem solving by graph searching	120
4.5	A graph, with cycles, for the delivery robot domain	127
4.6	Summary of search strategies	138
4.7	Iterative deepening searcher	142
4.8	Arc consistency algorithm AC-3	153
4.9	Domain-consistent constraint network for the scheduling problem .	154
4.10	A foothill, plateau, and ridge on a contour map	159

4.11	Simulated annealing algorithm	160
4.12	A grid searching problem	165
4.13	A crossword puzzle to be solved with six words	167
5.1	The knowledge representation framework	170
5.2	Solution quality as a function of time for an anytime algorithm . . .	174
5.3	Lattice of sublogics	176
5.4	A hierarchy of representations	178
5.5	A flat semantic network	186
5.6	A structured semantic network	189
6.1	Knowledge-based system architecture	200
6.2	The non-ground representation for the base language	204
6.3	The vanilla definite clause meta-interpreter	205
6.4	A base-level knowledge base for house wiring	206
6.5	A meta-interpreter that uses built-in calls and disjunction	208
6.6	A meta-interpreter for depth-bounded search	209
6.7	A meta-interpreter that collects delayed goals	211
6.8	An ask-the-user interpreter in pseudo-code	213
6.9	A meta-interpreter that builds a proof tree	218
6.10	A meta-interpreter that collects ancestor rules for WHY questions . .	221
6.11	An algorithm for debugging incorrect answers	223
6.12	An algorithm for debugging missing answers	225
6.13	Meta-interpreter with search	228
6.14	Pseudocode for unification using the ground representation	232
7.1	Two chairs	236
7.2	Bottom-up proof procedure for computing conflicts of T	246
7.3	A meta-interpreter to find conflicts	248
7.4	Bottom-up negation as failure proof procedure	254
7.5	Truth table for the clausal operators	259
7.6	Bottom-up proof procedure for computing prime implicates of KB .	261
7.7	Meta-interpreter for general clauses	263
7.8	A proof tree for Example 7.32	265
7.9	Theorem prover with answer extraction	267
7.10	Truth table for predicate calculus operators	272
7.11	Modal logic, their constraints on R , and their axioms	276
7.12	Axioms for L	277
8.1	The delivery robot domain with objects	285
8.2	A simple STRIPS planner that uses STRIPS representation	302
8.3	Finding weakest preconditions using the STRIPS representation . .	306

8.4	A regression planner	307
8.5	Partial-order planner	311
8.6	Threat handler for the partial-order planner	313
8.7	Partial elaboration of the partial-ordered planning example	314
9.1	Diagrammatic representation of the defaults from Example 9.3	325
9.2	A diagrammatic representation of Example 9.5	328
9.3	Causal network for Example 9.8	336
10.1	Belief network for the electrical domain of Figure 3.1	365
10.2	Belief network for report of leaving of Example 10.16	370
10.3	Belief network for aching limbs of Example 10.17	374
10.4	Decision tree for delivery robot decision of Example 10.20	383
10.5	Decision network for delivery robot decision	387
10.6	Decision network for the alarm problem	388
10.7	Value for alarm decision network	388
10.8	Possible money-utility tradeoff from Example 10.29	394
10.9	Belief network for overhead projector	396
11.1	The learning architecture	398
11.2	Some classification data for learning user preferences	400
11.3	Simple decision tree	404
11.4	Simple decision tree learning program for binary attributes	406
11.5	The sigmoid or logistic function	410
11.6	Simple neural network with one hidden layer	411
11.7	Simulation of the neural net learning algorithm	413
11.8	Candidate elimination algorithm	419
11.9	Posterior distributions of <i>probA</i> based on different samples	425
11.10	Belief network corresponding to a naive Bayesian classifier	430
11.11	A meta-interpreter for explanation-based learning	434
11.12	Example background KB for explanation-based learning	436
12.1	A robotic system and its components	445
12.2	A hierarchical robotic system architecture	451
12.3	A hierarchical decomposition of the delivery robot	454
12.4	A simulation of the robot carrying out the plan of Example 12.4	457
B.1	Syntactic translation between this book and standard Prolog	479
C.1	Forward-chaining assumption-based reasoner	495

1 Introduction to Computational Intelligence 1.1 Computational Intelligence Paradigms . . . 1.1.1 Artificial Neural Networks . . .

In this context, the book includes artificial neural networks, evolutionary computation, swarm intelligence, artificial immune systems, and fuzzy systems, which are respectively models of the following natural systems: biological neural networks, evolution, swarm behavior of social organisms, natural immune systems, and human thinking processes. The expression computational intelligence (CI) usually refers to the ability of a computer to learn a specific task from data or experimental observation. Even though it is commonly considered a synonym of soft computing, there is still no commonly accepted definition of computational intelligence. Generally, computational intelligence is a set of nature-inspired computational methodologies and approaches to address complex real-world problems to which mathematical or traditional modelling can be Computational Intelligence (Soft Computing) is a new concept for advanced information processing. The objective of CI approaches is to realize a new approach for analyzing and create flexible information processing of humans such as sensing, understanding, learning, recognizing and thinking. The ANN simulates physiological features of the human brain, and has been applied for non-linear mapping by numerical approach. Computational Intelligence (CI) is the theory, design, application and development of biologically and linguistically motivated computational paradigms. Traditionally the three main pillars of CI have been Neural Networks, Fuzzy Systems and Evolutionary Computation. However, in time many nature inspired computing paradigms have evolved.