

ARTIFICIAL INTELLIGENCE: A New Synthesis



Nils J. Nilsson

Artificial Intelligence

A New Synthesis

This page intentionally left blank

Artificial Intelligence

A New Synthesis

Nils J. Nilsson

Stanford University



**Morgan Kaufmann Publishers, Inc.
San Francisco, California**

Sponsoring Editor Michael B. Morgan
Director of Production and Manufacturing Yonie Overton
Production Editor Cheri Palmer
Assistant Editor Marilyn Alan
Cover Design Carrie English, canary studios
Text Design Detta Penna, Penna Design & Production
Composition and Illustrations Windfall Software, using ZzTEX
Copyeditor Robert Fiske
Proofreader Jennifer McClain
Indexer Valerie Robbins

Morgan Kaufmann Publishers, Inc.

Editorial and Sales Office
340 Pine Street, Sixth Floor
San Francisco, CA 94104-3205
USA
Telephone 415 / 392-2665
Facsimile 415 / 982-2665
Email mkp@mfp.com
WWW www.mfp.com
Order toll free 800 / 745-7323

Advice, Praise, Errors: Any correspondence related to this publication or intended for the author should be addressed to the Editorial and Sales Office of Morgan Kaufmann Publishers, Inc., Dept. AI APE. Please report any errors by email to aibugs@mfp.com. Please check the errata page at <http://www.mfp.com/nl5/clarified> to see if the bug has already been reported and fixed.

© 1998 by Morgan Kaufmann Publishers, Inc.
All rights reserved



Transferred to Digital Printing 2009

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—with the prior written permission of the publisher.

Library of Congress Cataloging-in-Publication Data

Nilsson, Nils J., (date)

Artificial Intelligence : a new synthesis / Nils J. Nilsson.

p. cm.

Includes bibliographical references and index.

ISBN 1-55860-467-7 (cloth). — ISBN 1-55860-535-5 (paper)

1. Artificial intelligence. I. Title.

Q335.N495 1998

006.3—dc21

97-47159

CIP

For Scott and Ryan

This page intentionally left blank

Contents

Preface	xix
---------	-----

1	<i>Introduction</i>	1
1.1	What Is AI?	1
1.2	Approaches to Artificial Intelligence	6
1.3	Brief History of AI	8
1.4	Plan of the Book	11
1.5	Additional Readings and Discussion	14
Exercises		17

I	Reactive Machines	19
----------	--------------------------	-----------

2	<i>Stimulus-Response Agents</i>	21
2.1	Perception and Action	21
2.1.1	Perception	24
2.1.2	Action	24
2.1.3	Boolean Algebra	25
2.1.4	Classes and Forms of Boolean Functions	26

2.2 Representing and Implementing Action Functions	27
2.2.1 Production Systems	27
2.2.2 Networks	29
2.2.3 The Subsumption Architecture	32
2.3 Additional Readings and Discussion	33
Exercises	34
3 Neural Networks	37
 3.1 Introduction	37
 3.2 Training Single TLUs	38
3.2.1 TLU Geometry	38
3.2.2 Augmented Vectors	39
3.2.3 Gradient Descent Methods	39
3.2.4 The Widrow-Hoff Procedure	41
3.2.5 The Generalized Delta Procedure	41
3.2.6 The Error-Correction Procedure	43
 3.3 Neural Networks	44
3.3.1 Motivation	44
3.3.2 Notation	45
3.3.3 The Backpropagation Method	46
3.3.4 Computing Weight Changes in the Final Layer	48
3.3.5 Computing Changes to the Weights in Intermediate Layers	48
 3.4 Generalization, Accuracy, and Overfitting	51
 3.5 Additional Readings and Discussion	54
Exercises	55
4 Machine Evolution	59
 4.1 Evolutionary Computation	59
 4.2 Genetic Programming	60
4.2.1 Program Representation in GP	60

4.2.2 The GP Process	62
4.2.3 Evolving a Wall-Following Robot	65
4.3 Additional Readings and Discussion	69
Exercises	69
5 State Machines	71
5.1 Representing the Environment by Feature Vectors	71
5.2 Elman Networks	73
5.3 Iconic Representations	74
5.4 Blackboard Systems	77
5.5 Additional Readings and Discussion	80
Exercises	80
6 Robot Vision	85
6.1 Introduction	85
6.2 Steering an Automobile	86
6.3 Two Stages of Robot Vision	88
6.4 Image Processing	91
6.4.1 Averaging	91
6.4.2 Edge Enhancement	93
6.4.3 Combining Edge Enhancement with Averaging	96
6.4.4 Region Finding	97
6.4.5 Using Image Attributes Other Than Intensity	101
6.5 Scene Analysis	102
6.5.1 Interpreting Lines and Curves in the Image	103
6.5.2 Model-Based Vision	106
6.6 Stereo Vision and Depth Information	108
6.7 Additional Readings and Discussion	110
Exercises	111

II Search in State Spaces 115

7 Agents That Plan	117
7.1 Memory Versus Computation	117
7.2 State-Space Graphs	118
7.3 Searching Explicit State Spaces	121
7.4 Feature-Based State Spaces	122
7.5 Graph Notation	124
7.6 Additional Readings and Discussion	125
Exercises	126
8 Uninformed Search	129
8.1 Formulating the State Space	129
8.2 Components of Implicit State-Space Graphs	130
8.3 Breadth-Fist Search	131
8.4 Depth-Fist or Backtracking Search	133
8.5 Iterative Deepening	135
8.6 Additional Readings and Discussion	136
Exercises	137
9 Heuristic Search	139
9.1 Using Evaluation Functions	139
9.2 A General Graph-Searching Algorithm	141
9.2.1 Algorithm A*	142
9.2.2 Admissibility of A*	145
9.2.3 The Consistency (or Monotone) Condition	150
9.2.4 Iterative-Deepening A*	153
9.2.5 Recursive Best-First Search	154

9.3	Heuristic Functions and Search Efficiency	155
9.4	Additional Readings and Discussion	160
	Exercises	160
10	Planning, Acting, and Learning	163
10.1	The Sense/Plan/Act Cycle	163
10.2	Approximate Search	165
10.2.1	Island-Driven Search	166
10.2.2	Hierarchical Search	167
10.2.3	Limited-Horizon Search	169
10.2.4	Cycles	170
10.2.5	Building Reactive Procedures	170
10.3	Learning Heuristic Functions	172
10.3.1	Explicit Graphs	172
10.3.2	Implicit Graphs	173
10.4	Rewards Instead of Goals	175
10.5	Additional Readings and Discussion	177
	Exercises	178
11	Alternative Search Formulations and Applications	181
11.1	Assignment Problems	181
11.2	Constructive Methods	183
11.3	Heuristic Repair	187
11.4	Function Optimization	189
	Exercises	192
12	Adversarial Search	195
12.1	Two-Agent Games	195
12.2	The Minimax Procedure	197

12.3	The Alpha-Beta Procedure	202
12.4	The Search Efficiency of the Alpha-Beta Procedure	207
12.5	Other Important Matters	208
12.6	Games of Chance	208
12.7	Learning Evaluation Functions	210
12.8	Additional Readings and Discussion	212
	Exercises	213

III Knowledge Representation and Reasoning 215

13	The Propositional Calculus	217
13.1	Using Constraints on Feature Values	217
13.2	The Language	219
13.3	Rules of Inference	220
13.4	Definition of Proof	221
13.5	Semantics	222
13.5.1	Interpretations	222
13.5.2	The Propositional Truth Table	223
13.5.3	Satisfiability and Models	224
13.5.4	Validity	224
13.5.5	Equivalence	225
13.5.6	Entailment	225
13.6	Soundness and Completeness	226
13.7	The PSAT Problem	227
13.8	Other Important Topics	228
13.8.1	Language Distinctions	228
13.8.2	Metatheorems	228
13.8.3	Associative Laws	229

13.8.4 Distributive Laws	229
--------------------------	-----

Exercises	229
------------------	-----

14 Resolution in the Propositional Calculus	231
--	------------

14.1 A New Rule of Inference: Resolution	231
---	-----

14.1.1 Clauses as wffs	231
------------------------	-----

14.1.2 Resolution on Clauses	231
------------------------------	-----

14.1.3 Soundness of Resolution	232
--------------------------------	-----

14.2 Converting Arbitrary wffs to Conjunctions of Clauses	232
--	-----

14.3 Resolution Refutations	233
------------------------------------	-----

14.4 Resolution Refutation Search Strategies	235
---	-----

14.4.1 Ordering Strategies	235
----------------------------	-----

14.4.2 Refinement Strategies	236
------------------------------	-----

14.5 Horn Clauses	237
--------------------------	-----

Exercises	238
------------------	-----

15 The Predicate Calculus	239
----------------------------------	------------

15.1 Motivation	239
------------------------	-----

15.2 The Language and Its Syntax	240
---	-----

15.3 Semantics	241
-----------------------	-----

15.3.1 Worlds	241
---------------	-----

15.3.2 Interpretations	242
------------------------	-----

15.3.3 Models and Related Notions	243
-----------------------------------	-----

15.3.4 Knowledge	244
------------------	-----

15.4 Quantification	245
----------------------------	-----

15.5 Semantics of Quantifiers	246
--------------------------------------	-----

15.5.1 Universal Quantifiers	246
------------------------------	-----

15.5.2 Existential Quantifiers	247
--------------------------------	-----

15.5.3 Useful Equivalences	247
----------------------------	-----

15.5.4 Rules of Inference	247
---------------------------	-----

15.6	Predicate Calculus as a Language for Representing Knowledge	248
15.6.1	Conceptualizations	248
15.6.2	Examples	248
15.7	Additional Readings and Discussion	250
	Exercises	250
16	Resolution in the Predicate Calculus	253
16.1	Unification	253
16.2	Predicate-Calculus Resolution	256
16.3	Completeness and Soundness	257
16.4	Converting Arbitrary wffs to Clause Form	257
16.5	Using Resolution to Prove Theorems	260
16.6	Answer Extraction	261
16.7	The Equality Predicate	262
16.8	Additional Readings and Discussion	265
	Exercises	265
17	Knowledge-Based Systems	269
17.1	Confronting the Real World	269
17.2	Reasoning Using Horn Clauses	270
17.3	Maintenance in Dynamic Knowledge Bases	275
17.4	Rule-Based Expert Systems	280
17.5	Rule Learning	286
17.5.1	Learning Propositional Calculus Rules	286
17.5.2	Learning First-Order Logic Rules	291
17.5.3	Explanation-Based Generalization	295
17.6	Additional Readings and Discussion	297
	Exercises	298

18	Representing Commonsense Knowledge	301
18.1	The Commonsense World	301
18.1.1	What Is Commonsense Knowledge?	301
18.1.2	Difficulties in Representing Commonsense Knowledge	303
18.1.3	The Importance of Commonsense Knowledge	304
18.1.4	Research Areas	305
18.2	Time	306
18.3	Knowledge Representation by Networks	308
18.3.1	Taxonomic Knowledge	308
18.3.2	Semantic Networks	309
18.3.3	Nonmonotonic Reasoning in Semantic Networks	309
18.3.4	Frames	312
18.4	Additional Readings and Discussion	313
	Exercises	314
19	Reasoning with Uncertain Information	317
19.1	Review of Probability Theory	317
19.1.1	Fundamental Ideas	317
19.1.2	Conditional Probabilities	320
19.2	Probabilistic Inference	323
19.2.1	A General Method	323
19.2.2	Conditional Independence	324
19.3	Bayes Networks	325
19.4	Patterns of Inference in Bayes Networks	328
19.5	Uncertain Evidence	329
19.6	D-Separation	330
19.7	Probabilistic Inference in Polytrees	332
19.7.1	Evidence Above	332
19.7.2	Evidence Below	334

19.7.3 Evidence Above and Below	336
19.7.4 A Numerical Example	336
19.8 Additional Readings and Discussion	338
Exercises	339
 20 Learning and Acting with Bayes Nets	 343
20.1 Learning Bayes Nets	343
20.1.1 Known Network Structure	343
20.1.2 Learning Network Structure	346
20.2 Probabilistic Inference and Action	351
20.2.1 The General Setting	351
20.2.2 An Extended Example	352
20.2.3 Generalizing the Example	356
20.3 Additional Readings and Discussion	358
Exercises	358
 IV Planning Methods Based on Logic	 361
 21 The Situation Calculus	 363
21.1 Reasoning about States and Actions	363
21.2 Some Difficulties	367
21.2.1 Frame Axioms	367
21.2.2 Qualifications	369
21.2.3 Ramifications	369
21.3 Generating Plans	369
21.4 Additional Readings and Discussion	370
Exercises	371

22	Planning	373
22.1	STRIPS Planning Systems	373
22.1.1	Describing States and Goals	373
22.1.2	Forward Search Methods	374
22.1.3	Recursive STRIPS	376
22.1.4	Plans with Run-Time Conditionals	379
22.1.5	The Sussman Anomaly	380
22.1.6	Backward Search Methods	381
22.2	Plan Spaces and Partial-Order Planning	385
22.3	Hierarchical Planning	393
22.3.1	ABSTRIPS	393
22.3.2	Combining Hierarchical and Partial-Order Planning	395
22.4	Learning Plans	396
22.5	Additional Readings and Discussion	398
	Exercises	400

V Communication and Integration 405

23	Multiple Agents	407
23.1	Interacting Agents	407
23.2	Models of Other Agents	408
23.2.1	Varieties of Models	408
23.2.2	Simulation Strategies	410
23.2.3	Simulated Databases	410
23.2.4	The Intentional Stance	411
23.3	A Modal Logic of Knowledge	412
23.3.1	Modal Operators	412
23.3.2	Knowledge Axioms	413

23.3.3 Reasoning about Other Agents' Knowledge	415
23.3.4 Predicting Actions of Other Agents	417
23.4 Additional Readings and Discussion	417
Exercises	418
 24 Communication among Agents	 421
24.1 Speech Acts	421
24.1.1 Planning Speech Acts	423
24.1.2 Implementing Speech Acts	423
24.2 Understanding Language Strings	425
24.2.1 Phrase-Structure Grammars	425
24.2.2 Semantic Analysis	428
24.2.3 Expanding the Grammar	432
24.3 Efficient Communication	435
24.3.1 Use of Context	435
24.3.2 Use of Knowledge to Resolve Ambiguities	436
24.4 Natural Language Processing	437
24.5 Additional Readings and Discussion	440
Exercises	440
 25 Agent Architectures	 443
25.1 Three-Level Architectures	444
25.2 Goal Arbitration	446
25.3 The Triple-Tower Architecture	448
25.4 Bootstrapping	449
25.5 Additional Readings and Discussion	450
Exercises	450
 Bibliography	 453
Index	493

Preface

This introductory textbook employs a novel perspective from which to view topics in artificial intelligence (AI). I will consider a progression of AI systems or “agents,” each slightly more complex than its predecessor. I begin with elementary agents that react to sensed properties of their environments. Even such simple machines allow me to treat topics in machine vision, machine learning, and machine evolution. Then, by stages, I introduce techniques that allow agents to exploit information about the task environment that cannot be immediately sensed. Such knowledge can take the form of descriptive information about the state of the environment, iconic models of the environment, state-space graphs, and logical representations. Because the progression follows what plausibly might have been milestones in the evolution of animals, I have called this approach *evolutionary artificial intelligence*. I intend the book to be as much a proposal about how to think about AI as it is a description of AI techniques. Examples will be used to provide motivation and grounding.

Although I use agents to motivate and illustrate AI techniques, the techniques themselves have much broader application. Many ideas invented by AI researchers have been assimilated into computer science generally for applications in expert systems, natural language processing, human-machine interaction, information retrieval, graphics and image processing, data mining, and robotics (to name some examples). The agents theme serves to unify what might otherwise seem to be a collection of disparate topics.

Regarding coverage, my intention is to treat the middle ground between theory and applications. This middle ground is rich in important AI *ideas*, and in this book I try to motivate and explain the ideas that I think have lasting value in AI. (Being subject to the usual human frailties, I admit to possible errors of omission

and commission in selecting topics for inclusion.) Also, some subjects are treated in more depth than others—both because I thought some subjects more important and because I wanted to provide at least some examples of greater depth of exposition. Although some pseudocode algorithms are presented, the book is not an AI programming and implementation book. (Some “AI techniques” books are [Shoham 1994, Norvig 1992, Tracy & Bouthoorn 1997].) I do not give proofs of all of the important theoretical results, but I try to give intuitive arguments and citations to formal proofs. My goals are to present a modest-sized textbook for a one-semester introductory college course, to give the student and reader sufficient motivation and preparation to go on to more advanced AI courses, and to make the extensive literature on AI accessible.

A somewhat unconventional feature of the book is that machine learning is not treated as a separate topic; instead, various aspects of learning arise throughout the book. Neural nets and fundamental ideas about supervised learning are presented early; techniques for learning search heuristics and action policies are discussed in the chapters on search; rule learning, inductive logic programming, and explanation-based learning are treated toward the end of the chapters on logic; and learning plans is presented after discussing logic-based planning.

In my previous books, I included a “bibliographic and historical remarks” section at the end of each chapter. (Some readers may find those sections of some interest still.) I have not done so in the present book, both because AI history has now accumulated to such a great extent and because the longer text by [Russell & Norvig 1995] has already done such a thorough job in that regard. Instead, I include remarks and citations as appropriate throughout the text and provide some additional ones in discussion sections at the end of most chapters. The serious student who intends to specialize in AI research will want to consult many of the references. I hope the casual reader is not bothered by the many citations.

Sample exercises are included at the end of each chapter. They vary in difficulty from routine application of ideas presented in the book to mildly challenging. I expect that instructors will want to augment these problems with favorite ones of their own, including computer exercises and projects. (In keeping with my decision to concentrate on ideas instead of programs, I have not included any computer exercises or projects. Several good programming and project ideas can be found in texts devoted to AI programming techniques.)

The following typographical conventions are used in this book. Sans serif font is used for the names of actions and for “proto-English” sentences communicated among agents. SANS SERIF capitals are used for the names of computer languages, algorithms, and AI systems. Boldface capital letters, such as **W** and **X**, are used for vectors, matrices, and modal operators. Typewriter font is used for genetic programs, for expressions and subexpressions in the predicate calculus, and for

STRIPS rules and operators. Lowercase Greek letters are used for metavariables ranging over predicate-calculus expressions, subexpressions, and occasionally for substitutions. Uppercase Greek letters are used to denote sets of predicate-calculus formulas. Lowercase p's are used to denote probabilities.

Students and researchers will find much helpful material about AI on the World Wide Web. I do not provide URLs here; any list written today would be incomplete and inaccurate within months. Use of one of the web search engines will quickly steer the reader to sites with sample applications, frequently asked questions, extensive bibliographies, research papers, programs, interactive demonstrations, announcements of workshops and conferences, homepages of researchers, and much more.

Material specifically in support of this book is provided on a Web page on the publisher's Web site at www.mkp.com/nips. If you discover any errors, please email them to the publisher at aibugs@mkp.com. Errata and clarifications can be found at <http://www.mkp.com/nips/clarified>.

My previous AI textbook, *Principles of Artificial Intelligence* (Morgan Kaufmann, 1980), is by now quite out of date, but some of the material in that book is still useful, and I have borrowed freely from it in preparing the present volume. Cross-checking against other AI textbooks (particularly [Russell & Norvig 1995, Rich & Knight 1991, Stefik 1995]) was also very helpful.

Students and teaching assistants in my Stanford courses on artificial intelligence and machine learning have already made several useful suggestions. I hope the following list includes most of them: Eyal Amir, David Andre, Scott Benson, George John, Steve Ketchpel, Ron Kohavi, Andrew Kosoresow, Ofer Matan, Karl Pfleger, and Charles Richards. Colleagues and reviewers at Stanford and elsewhere helped me learn what they already knew. Thanks to Helder Coelho, Oscar Firschein, Carolyn Hayes, Giorgio Ingargiola, Leslie Kaelbling, Daphne Koller, John Koza, Richard Korf, Pat Langley, John McCarthy, Bart Selman, Yoav Shoham, Devika Subramanian, Gheorghe Tecuci, and Michael Wellman. Special thanks go to Cheri Palmer, my production editor at Morgan Kaufmann, who kept me on schedule, cheerfully accepted my endless changes, and worked extra hard to meet a difficult publication date. Work on this book was carried on in the Robotics Laboratory of Stanford's Department of Computer Science and at the Santa Fe Institute. Continuing research support by the National Science Foundation is gratefully acknowledged.

This page intentionally left blank

1 Introduction

I believe that understanding intelligence involves understanding how knowledge is acquired, represented, and stored; how intelligent behavior is generated and learned; how motives, and emotions, and priorities are developed and used; how sensory signals are transformed into symbols; how symbols are manipulated to perform logic, to reason about the past, and plan for the future; and how the mechanisms of intelligence produce the phenomena of illusion, belief, hope, fear, and dreams—and yes even kindness and love. To understand these functions at a fundamental level, I believe, would be a scientific achievement on the scale of nuclear physics, relativity, and molecular genetics.

—James Albus, Response to Henry Hexmoor, from URL:
<http://tommy.jsc.nasa.gov/er/er6/mrl/papers/symposium/albus.txt>
February 13, 1995

1.1 What Is AI?

Artificial Intelligence (AI), broadly (and somewhat circularly) defined, is concerned with intelligent behavior in artifacts. Intelligent behavior, in turn, involves perception, reasoning, learning, communicating, and acting in complex environments. AI has as one of its long-term goals the development of machines that can do these things as well as humans can, or possibly even better. Another goal

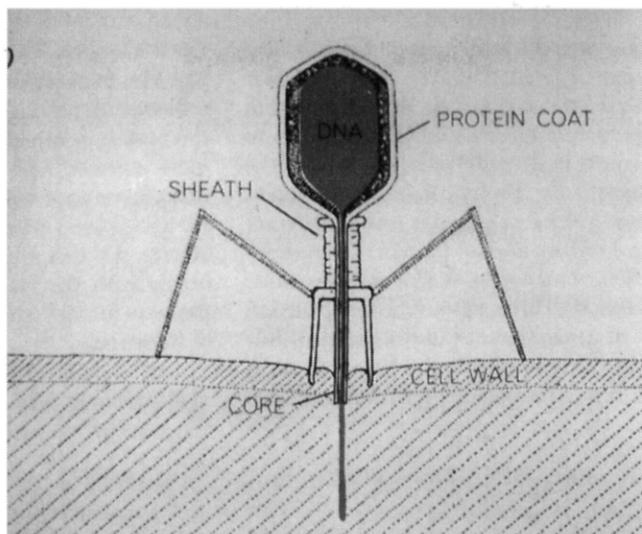
of AI is to understand this kind of behavior whether it occurs in machines or in humans or other animals. Thus, AI has both engineering and scientific goals. In this book, I will largely be concerned with AI as engineering, focussing on the important concepts and ideas underlying the design of intelligent machines.

AI has always been surrounded by controversy. The question *Can machines think?* has interested philosophers as well as scientists and engineers. In a famous article, Alan Turing, one of the founders of computer science, rephrased that question in terms more amenable to an empirical test, which has come to be called the *Turing test* [Turing 1950]. I will describe that test shortly, but Turing also noted that the answer to the question “Can machines think?” depends on how we define the words *machine* and *think*. He might also have added that it depends on how we define *can*.

Let’s consider the word *can* first. Do we mean can machines think someday or can they now? Do we mean that in principle they might be able to think (even if we could never build ones that do), or are we asking for an actual demonstration? These are important questions since no artifact yet possesses broad thinking skills.

Some people believe that thinking machines might have to be so complex and have such complex experiences (interacting with their environment and with other thinking machines, for example) that we could never actually design or build them. The processes that generate global weather provide a good analogy. Even if we knew everything of importance about the weather, that knowledge wouldn’t necessarily allow us to duplicate weather phenomena artificially in all of its richness. No system less complex than the actual earth’s surface, atmosphere, and seas—embedded in space, warmed by the sun, and influenced by the tides—would be able to duplicate weather phenomena in all of their details. Similarly, full-scale, human-level intelligence may be too complex, or at least too dependent on the precise physiology of humans, to exist apart from its *embodiment* in humans situated in their environment. (For a discussion of the importance of this notion of embodiment see, for example, [Lakoff 1987, Winograd & Flores 1986, Harnad 1990, Mataric 1997].) The matter of whether or not we could ever build human-level thinking machines is still undecided, and AI progress toward that goal has been steady, albeit slower than some early pioneers predicted. I am optimistic about our eventual success.

Next, we come to the word *machine*. To many people, a machine is a rather stolid thing. The word evokes images of gears grinding, steam hissing, and steel parts clanking. How could such a thing think? Nowadays, however, the computer has greatly expanded our notion of what a machine can be. Our growing understanding of biological mechanisms is expanding it even further. Consider, for example, the simple virus called *E6 Bacteriophage* shown schematically in Figure 1.1. Its head contains viral DNA. The virus attaches itself to the cell wall of a bacterium with its tail fibers, punctures the wall, and squirts its DNA into the bacterium. The DNA then directs the bacterium to manufacture thousands of

**Figure 1.1**

Schematic Illustration of E6 Bacteriophage

copies of all of the viral parts. These parts then automatically assemble themselves into new viruses that explode out of the bacterium to repeat the process. The complete assembly looks and operates very much like a machine, and we might as well call it a machine—one made of proteins.

What about other biological processes and organisms? The complete genome of the bacterium *Haemophilus influenzae* Rd has recently been sequenced [Fleischmann, et al. 1995]. This genome has 1,830,137 base pairs (consisting of the “letters” A, G, C, and T). That’s roughly 3.6×10^6 bits or one-half a megabyte. Although the function of all of its 1743 genes is not yet known, scientists are beginning to explain the development and functioning of such organisms in the same way that they would explain machines—very complex machines, of course. In fact, techniques quite familiar to computer scientists, namely, the use of timing diagrams for logic circuits, are proving useful for understanding the regulation by genes of the complex biochemistry of a bacteria-infecting virus [McAdams & Shapiro 1995]. Sequencing the complete genomes of other organisms, including humans, is proceeding. Once we know these “blueprints,” will we think of these organisms—bacteria, worms, fruit flies, mice, dolphins, humans—as machines? If humans are machines, then machines *can* think! We have an existence proof. We “simply” don’t know yet how the human machine works.

Yet, even should we agree about what a machine is, there are some additional twists to this argument. Even if machines made of proteins can think, perhaps ones made of silicon wouldn’t be able to. A well-known philosopher, John Searle,

believes that what we are made of is fundamental to our intelligence [Searle 1980, Searle 1992]. For him, thinking can occur only in very special machines—living ones made of proteins.

Directly opposed to Searle's belief (and to the notion of embodiment mentioned earlier) is the *physical symbol system hypothesis* of Newell and Simon [Newell & Simon 1976]. That hypothesis states that a physical symbol system has the necessary and sufficient means for general intelligent action. According to Newell and Simon, a physical symbol system is a machine, like a digital computer, that is capable of manipulating symbolic data—adding numbers, rearranging lists of symbols (such as alphabetizing a list of names), replacing some symbols by others, and so on. An important aspect of this hypothesis is that it doesn't matter what the physical symbol system is made of! Newell and Simon's hypothesis is "substrate neutral." An intelligent entity could be made of protein, mechanical relays, transistors, or anything else, so long as it can process symbols.¹

Still others believe that it isn't whether or not machines are made of silicon or protein that is important; these people think that much intelligent behavior is the result of what they call *subsymbolic* processing—processing of *signals*, not symbols. Take the recognition of familiar faces, for example. Humans do that effortlessly, and although we don't know how they do it, it is suspected that the best explanation for the process would involve treating images or parts of them as multidimensional signals, not as symbols.

I could list many other points of view about what sorts of machines might be capable of humanlike thought. Some of the claims one often hears are

- The brain processes information in parallel, whereas conventional computers do it serially. We'll have to build new varieties of parallel computers to make progress in AI.
- Conventional computing machinery is based on true-or-false (binary) logic. Truly intelligent systems will have to use some sort of "fuzzy" logic.
- Animal neurons are much more complex than switches—the basic building blocks of computers. We'll need to use quite realistic artificial neurons in intelligent machines.

Perhaps it is still too early for the field of AI to reach a consensus about what sort of machinery is required, although many AI researchers accept the physical symbol system hypothesis.

Finally, we come to the most difficult word, *think*. Rather than attempt to define this word, Turing proposed a test, the Turing test, by which it could be

1. Of course, some building materials will be better than others if we take into account such practical matters as speed, permanence, reliability, suitability for parallel processing, temperature sensitivity, and so on.