Mind

Introduction to Cognitive Science

second edition

Paul Thagard

A Bradford Book
The MIT Press
Cambridge, Massachusetts
London, England

MIT Press books may be purchased at special quantity discounts for business or sales promotional use. For information, please email special_sales@mitpress.mit.edu or write to Special Sales Department, The MIT Press, 5 Cambridge Center, Cambridge, MA 02142.

This book was set in Stone sans and Stone serif by SNP Best-set Typesetter Ltd., Hong Kong. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Thagard, Paul.
Mind: introduction to cognitive science/Paul Thagard.—2nd ed.
p. cm.
"A Bradford book."
Includes bibliographical references and index.
ISBN-10: 0-262-20154-2 (hc. : alk. paper)—0-262-70109-X (pbk. : alk. paper)
ISBN-13: 978-0-262-20154-4 (hc. : alk. paper)—978-0-262-70109-9 (pbk. : alk. paper)
1. Cognitive science—Textbooks. I. Title.

BF311.T42 2005
153—dc22

2004053091

10 9 8 7 6 5 4 3

bortekre, 1.10.09

For Adam, megamind

Contents

Preface

Cognitive science is the interdisciplinary study of mind and intelligence, embracing philosophy, psychology, artificial intelligence, neuroscience, linguistics, and anthropology. Its intellectual origins are in the mid-1950s when researchers in several fields began to develop theories of mind based on complex representations and computational procedures. Its organizational origins are in the mid-1970s when the Cognitive Science Society was formed and the journal *Cognitive Science* began. Since then, more than sixty universities in North America and Europe have established cognitive science programs and many others have instituted courses in cognitive science.

Teaching an interdisciplinary course in cognitive science is difficult because students come to it with very different backgrounds. Since 1993, I have been teaching a popular course at the University of Waterloo called Introduction to Cognitive Science. On the one hand, the course attracts computationally sophisticated students from computer science and engineering who know little psychology or philosophy; on the other, it attracts students with good backgrounds in psychology or philosophy but who know little about computation. This text is part of an attempt to construct a course that presupposes no special preparation in any of the fields of cognitive science. It is intended to enable students with an interest in mind and intelligence to see that there are many complementary approaches to the investigation of mind.

There are at least three different ways to introduce cognitive science to a multidisciplinary audience. The first is to concentrate on the different fields of psychology, artificial intelligence, and so on. The second is to organize the discussion by different functions of mind, such as problem solving, memory, learning, and language. I have chosen a third approach, systematically describing and evaluating the main theories of mental representation that have been advocated by cognitive scientists, including logic, rules, concepts, analogies, images, and connections (artificial neural networks). Discussing these fundamental theoretical approaches provides a unified way of presenting the accomplishments of the different fields of cognitive science to understanding various important mental functions.

My goal in writing this book is to make it accessible to all students likely to enroll in an introduction to cognitive science. Accomplishing this goal requires, for example, explaining logic in a way accessible to psychology students, computer algorithms in a way accessible to English students, and philosophical controversies in a way accessible to computer science students.

Although this book is intended for undergraduates, it should also be useful for graduate students and faculty who want to see how their own fields fit into the general enterprise of cognitive science. I have not written an encyclopedia. Since the whole point of this exercise is to provide an integrated introduction, I have kept the book relatively short and to the point, highlighting the forest rather than the trees. Viewing cognitive science as the intersection rather than as the union of all the relevant fields, I have omitted many topics that are standard in introductions to artificial intelligence, cognitive psychology, philosophy of mind, and so on. Each chapter concludes with a summary and suggestions for further reading.

The book is written with great enthusiasm for what theories of mental representation and computation have contributed to the understanding of mind, but also with awareness that cognitive science has a long way to go. The second part of the book discusses extensions to the basic assumptions of cognitive science and suggests directions for future interdisciplinary work.

I have been grateful for the reception of the first edition of this book, especially its translation into Italian, German, Czech, Portuguese, Japanese, Korean, and two variants of Chinese. For this second edition, I have brought part I up to date and substantially revised part II, adding new chapters on brains, emotions, and consciousness. Other additions include a list of relevant Web sites at the end of each chapter, and a glossary at the end of the book. My anthology, *Mind Readings: Introductory Selections on Cognitive Science* (MIT Press, 1998) remains a useful accompaniment.

Acknowledgments

I am grateful to the students at the University of Waterloo who participated in several courses in which this material was first developed. For suggestions to the first edition, I owe thanks to Andrew Brook, Chris Eliasmith, Kim Honeyford, Janet Kolodner, Amy Pierce, Michael Ranney, and Eric Steinhart. For suggestions that contributed to the second edition, I am grateful to Chris Eliasmith, Ashok Goel, Michael Ranney, and Ethan Toombs. Thanks to Tom Stone at MIT Press for encouragement and suggestions. While I was writing and revising this text, my research was supported by the Natural Sciences and Engineering Research Council of Canada. Thanks to Abninder Litt for help with proofreading.

I Approaches to Cognitive Science

1 Representation and Computation

Studying the Mind

Have you ever wondered how your mind works? Every day, people accomplish a wide range of mental tasks: solving problems at their work or school, making decisions about their personal life, explaining the actions of people they know, and acquiring new concepts like *cell phone* and *Internet.* The main aim of cognitive science is to explain how people accomplish these various kinds of thinking. We want not only to describe different kinds of problem solving and learning, but also to explain how the mind carries out these operations. Moreover, cognitive science aims to explain cases where thinking works poorly—for example, when people make bad decisions.

Understanding how the mind works is important for many practical activities. Educators need to know the nature of students' thinking in order to devise better ways of teaching them. Engineers and other designers need to know what potential users of their products are likely to be thinking when they use their products effectively or ineffectively. Computers can be made more intelligent by reflecting on what makes people intelligent. Politicians and other decision makers can become more successful if they understand the mental processes of people with whom they interact.

But studying the mind is not easy, since we cannot just pop one open to see how it works. Over the centuries, philosophers and psychologists have used a variety of metaphors for the mind, comparing it, for example, to a blank sheet on which impressions are made, to a hydraulic device with various forces operating in it, and to a telephone switchboard. In the last fifty years, suggestive new metaphors for thinking have become available through the development of new kinds of computers. Many but not all cognitive scientists view thinking as a kind of computation and use computational metaphors to describe and explain how people solve problems and learn.

What Do You Know?

When students begin studying at a college or university, they have much more to learn than course material. Undergraduates in different programs will have to deal with very different subject matters, but they all need to acquire some basic knowledge about how the university works. How do you register for courses? What time do the classes begin? What courses are good and which are to be avoided? What are the requirements for a degree? What is the best route from one building to another? What are the other students on campus like? Where is the best place to have fun on Friday night?

Answers to these questions become part of the minds of most students, but what sort of part? Most cognitive scientists agree that knowledge in the mind consists of *mental representations*. Everyone is familiar with nonmental representations, such as the words on this page. I have just used the words "this page" to represent the page that you are now seeing. Students often also use pictorial representations such as maps of their campuses and buildings. To account for many kinds of knowledge, such as what students know about the university, cognitive scientists have proposed various kinds of mental representation including rules, concepts, images, and analogies. Students acquire rules such as *If I want to graduate, then I need to take ten courses in my major*. They also acquire concepts involving new terms such as "bird" or "Mickey Mouse" or "gut," all used to describe a particularly easy course. For getting from building to building, a mental image or picture of the layout of the campus might be very useful. After taking a course that they particularly like, students may try to find another similar course to take. Having interacted with numerous students from different programs on campus, students may form stereotypes of the different kinds of undergraduates, although it may be difficult for them to say exactly what constitutes those stereotypes.

The knowledge that students acquire about college life is not acquired just for the sake of accumulating information. Students face numerous problems, such as how to do well in their courses, how to have a decent social life, and how to get a job after graduation. Solving such problems requires doing things with mental representations, such as reasoning that you still need five more courses to graduate, or deciding never to take another course from Professor Tedium. Cognitive science proposes that people have mental *procedures* that operate on mental representations to produce thought and action. Different kinds of mental representations such as rules and concepts foster different kinds of mental procedures. Consider different ways of representing numbers. Most people are familiar with the Arabic numeral representation of numbers (1, 2, 3, 10, 100, etc.) and with the standard procedures for doing addition, multiplication, and so on. Roman numerals can also represent numbers (I, II, III, X, C), but they require different procedures for carrying out arithmetic operations. Try dividing CIV (104) by XXVI (26).

Part I of this book surveys the different approaches to mental representations and procedures that have developed in the last four decades of cognitive science research. There has been much controversy about the merits of different approaches, and many of the leading cognitive science theorists have argued vehemently for the primacy of the approach they prefer. My approach is more eclectic, since I believe that the different theories of mental representation now available are more complementary than competitive. The human mind is astonishingly complex, and our understanding of it can gain from considering its use of rules such as those described above as well as many other kinds of representations including some not at all familiar. The latter include "connectionist" or "neural network" representations that are discussed in chapter 7.

Beginnings

Attempts to understand the mind and its operation go back at least to the ancient Greeks, when philosophers such as Plato and Aristotle tried to explain the nature of human knowledge. Plato thought that the most important knowledge comes from concepts such as *virtue* that people know innately, independently of sense experience. Other philosophers such as Descartes and Leibniz also believed that knowledge can be gained just by thinking and reasoning, a position known as *rationalism*. In contrast, Aristotle discussed knowledge in terms of rules such as *All humans are mortal* that are learned from experience. This philosophical position, defended by Locke, Hume, and others, is known as *empiricism*. In the eighteenth century, Kant attempted to combine rationalism and empiricism by arguing that human knowledge depends on both sense experience and the innate capacities of the mind.

The study of mind remained the province of philosophy until the nineteenth century, when experimental psychology developed. Wilhelm Wundt and his students initiated laboratory methods for studying mental operations more systematically. Within a few decades, however, experimental psychology became dominated by *behaviorism*, a view that virtually denied the existence of mind. According to behaviorists such as J. B. Watson (1913), psychology should restrict itself to examining the relation between observable stimuli and observable behavioral responses. Talk of consciousness and mental representations was banished from respectable scientific discussion. Especially in North America, behaviorism dominated the psychological scene through the 1950s.

Around 1956, the intellectual landscape began to change dramatically. George Miller (1956) summarized numerous studies that showed that the capacity of human thinking is limited, with short-term memory, for example, limited to around seven items. (This is why it is hard to remember long phone or social security numbers.) He proposed that memory limitations can be overcome by recoding information into chunks, mental representations that require mental procedures for encoding and decoding the information. At this time, primitive computers had been around for only a few years, but pioneers such as John McCarthy, Marvin Minsky, Allen Newell, and Herbert Simon were founding the field of artificial intelligence. In addition, Noam Chomsky (1957, 1959) rejected behaviorist assumptions about language as a learned habit and proposed instead to explain people's ability to understand language in terms of mental grammars consisting of rules. The six thinkers mentioned in this paragraph can justly be viewed as the founders of cognitive science.

The subsequent history of cognitive science is sketched in later chapters in connection with different theories of mental representation. McCarthy became one of the leaders of the approach to artificial intelligence based on formal logic, which we will discuss in chapter 2. During the 1960s, Newell and Simon showed the power of rules for accounting for aspects of human intelligence, and chapter 3 describes considerable subsequent work in this tradition. During the 1970s, Minsky proposed that conceptlike frames are the central form of knowledge representations, and other researchers in artificial intelligence and psychology discussed similar structures called schemas and scripts (chapter 4). Also at this time, psychologists began to show increased interest in mental imagery (chapter 6). Much experimental and computational research since the 1980s has concerned analogical thinking, also known as case-based reasoning (chapter 5). The most exciting development of the 1980s was the rise of connectionist theories of mental representation and processing modeled loosely on neural networks in the brain (chapter 7). Each of these approaches has contributed to the understanding of mind, and chapter 8 provides a summary and evaluation of their advantages and disadvantages.

Many challenges and extensions have been made to the central view that the mind should be understood in terms of mental representations and procedures, and these are addressed in part II of the book (chapters 9–14). The 1990s saw a rapid increase in the use of brain scanning technologies to study how specific areas of the brain contribute to thinking, and currently there is much work on neurologically realistic computational models of mind (chapter 9). These models are suggesting new ways to understand emotions and consciousness (chapters 10 and 11). Chapters 12 and 13 address challenges to the computational-representational approach based on the role that bodies, physical environments, and social environments play in human thinking. Finally, chapter 14 discusses the future of cognitive science, including suggestions for how students can pursue further interdisciplinary work.

Methods in Cognitive Science

Cognitive science should be more than just people from different fields having lunch together to chat about the mind. But before we can begin to see the unifying ideas of cognitive science, we have to appreciate the diversity of outlooks and methods that researchers in different fields bring to the study of mind and intelligence.

Although cognitive psychologists today often engage in theorizing and computational modeling, their primary method is experimentation with human participants. People, usually undergraduates satisfying course requirements, are brought into the laboratory so that different kinds of thinking can be studied under controlled conditions. To take some examples from later chapters, psychologists have experimentally examined the kinds of mistakes people make in deductive reasoning, the ways that people form and apply concepts, the speed of people thinking with mental images, and the performance of people solving problems using analogies. Our conclusions about how the mind works must be based on more than "common sense" and introspection, since these can give a misleading picture of mental operations, many of which are not consciously accessible. Psychological experiments that carefully approach mental operations from diverse directions are therefore crucial for cognitive science to be scientific.

Although theory without experiment is empty, experiment without theory is blind. To address the crucial questions about the nature of mind, the psychological experiments need to be interpretable within a theoretical framework that postulates mental representations and procedures. One of the best ways of developing theoretical frameworks is by forming and testing computational models intended to be analogous to mental operations. To complement psychological experiments on deductive reasoning, concept formation, mental imagery, and analogical problem solving, researchers have developed computational models that simulate aspects of human performance. Designing, building, and experimenting with computational models is the central method of artificial intelligence (AI), the branch of computer science concerned with intelligent systems. Ideally in cognitive science, computational models and psychological experimentation go hand in hand, but much important work in AI has examined the power of different approaches to knowledge representation in relative isolation from experimental psychology.

Although some linguists do psychological experiments or develop computational models, most currently use different methods. For linguists in the Chomskyan tradition, the main theoretical task is to identify grammatical principles that provide the basic structure of human languages. Identification takes place by noticing subtle differences between grammatical and ungrammatical utterances. In English, for example, the sentences "She hit the ball" and "What do you like?" are grammatical, but "She the hit ball" and "What does you like?" are not. A grammar of English will explain why the former are acceptable but not the latter. Later chapters give additional examples of the theoretical and empirical work performed by linguists in both the Chomskyan tradition and others.

Like cognitive psychologists, neuroscientists often perform controlled experiments, but their observations are very different, since neuroscientists are concerned directly with the nature of the brain. With nonhuman subjects, researchers can insert electrodes and record the firing of individual neurons. With humans for whom this technique would be too invasive, it has become possible in recent years to use magnetic and positronic scanning devices to observe what is happening in different parts of the brain while people are doing various mental tasks. For example, brain scans have identified the regions of the brain involved in mental imagery and word interpretation. Additional evidence about brain functioning is gathered by observing the performance of people whose brains have been damaged in identifiable ways. A stroke, for example, in a part of the brain dedicated to language can produce deficits such as the inability to utter sentences. Like cognitive psychology, neuroscience is often theoretical as well as experimental, and theory development is frequently aided by developing computational models of the behavior of sets of neurons.

Cognitive anthropology expands the examination of human thinking to consider how thought works in different cultural settings. The study of mind should obviously not be restricted to how English speakers think but should consider possible differences in modes of thinking across cultures. Chapters 12 and 13 describe how cognitive science is becoming increasingly aware of the need to view the operations of mind in particular physical and social environments. For cultural anthropologists, the main method is ethnography, which requires living and interacting with members of a culture to a sufficient extent that their social and cognitive systems become apparent. Cognitive anthropologists have investigated, for example, the similarities and differences across cultures in words for colors.

With a few exceptions, philosophers generally do not perform systematic empirical observations or construct computational models. But philosophy remains important to cognitive science because it deals with fundamental issues that underlie the experimental and computational approaches to mind. Abstract issues such as the nature of representation and computation need not be addressed in the everyday practice of psychology or artificial intelligence, but they inevitably arise when researchers think deeply about what they are doing. Philosophy also deals with general questions such as the relation of mind and body and with methodological questions such as the nature of explanations found in cognitive science.

In addition to descriptive questions about how people think, philosophy concerns itself with normative questions about how they *should* think. Along with the theoretical goal of understanding human thinking, cognitive science can have the practical goal of improving it, which requires normative reflection on what we want thinking to be. Philosophy of mind does not have a distinct method, but should share with the best theoretical work in other fields a concern with empirical results.

In its weakest form, cognitive science is merely the sum of the fields just mentioned: psychology, artificial intelligence, linguistics, neuroscience, anthropology, and philosophy. Interdisciplinary work becomes much more interesting when there is theoretical and experimental convergence on conclusions about the nature of mind. Later chapters provide examples of such convergences that show cognitive science working at the intersection of various fields. For example, psychology and artificial intelligence can be combined through computational models of how people behave in experiments. The best way to grasp the complexity of human thinking is to use multiple methods, especially combining psychological and neurological experiments with computational models. Theoretically, the most fertile approach has been to understand the mind in terms of representation and computation.

The Computational-Representational Understanding of Mind

Here is the central hypothesis of cognitive science: Thinking can best be understood in terms of representational structures in the mind and computational procedures that operate on those structures. Although there is much disagreement about the nature of the representations and computations that constitute thinking, the central hypothesis is general enough to encompass the current range of thinking in cognitive science, including connectionist theories. For short, I call the approach to understanding the mind based on this central hypothesis *CRUM*, for *Computational-Representational Understanding of Mind.*

CRUM might be wrong. Part II of this book presents some fundamental challenges to this approach that suggest that ideas about representation and computation might be inadequate to explain fundamental facts about the mind. But in evaluating the successes of different theories of knowledge representation, we will be able to see the considerable progress in

understanding the mind that CRUM has made possible. Without a doubt, CRUM has been the most theoretically and experimentally successful approach to mind ever developed. Not everyone in the cognitive science disciplines agrees with CRUM, but inspection of the leading journals in psychology and other fields reveals that CRUM is currently the dominant approach to cognitive science.

Much of CRUM's success has been due to the fact that it employs a fertile analogy derived from the development of computers. As chapter 5 describes, analogies often contribute to new scientific ideas, and comparing the mind with computers has provided a much more powerful way of approaching the mind than previous metaphors such as the telephone switchboard. Readers with a background in computer science will be familiar with the characterization of a computer program as consisting of data structures and algorithms. Modern programming languages include a variety of data structures including strings of letters such as "abc," numbers such as 3, and more complex structures such as lists (A B C) and trees. Algorithms—mechanical procedures—can be defined to operate on various kinds of structures. For example, children in elementary school learn an algorithm for operating on numbers to perform long division. Another simple algorithm can be defined to reverse a list, turning (A B C) into (C B A). This procedure is built up out of smaller procedures for taking an element from one list and adding it to the beginning of another, enabling a computer to build a reversed list by forming (A), then (B A), then (C B A). Similarly, CRUM assumes that the mind has mental representations analogous to data structures, and computational procedures similar to algorithms. Schematically:

Program	*Mind*
data structures + algorithms = running programs	mental representations + computational procedures = thinking

This has been the dominant analogy in cognitive science, although it has taken on a novel twist from the use of another analog, the brain. Connectionists have proposed novel ideas about representation and computation that use neurons and their connections as inspirations for data structures, and neuron firing and spreading activation as inspirations for algorithms. CRUM then works with a complex three-way analogy among the mind, the brain, and computers, as depicted in figure 1.1. Mind, brain,

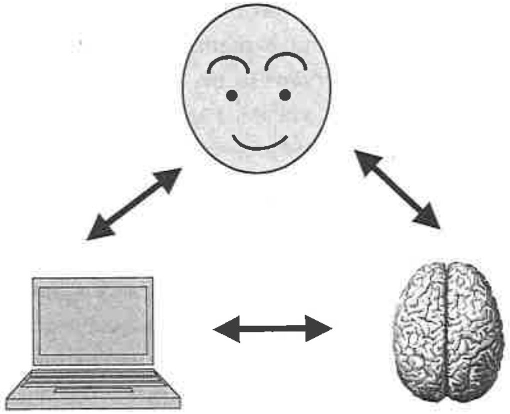


Figure 1.1
Three-way analogy between minds, computers, and brains.

and computation can each be used to suggest new ideas about the others. There is no single computational model of mind, since different kinds of computers and programming approaches suggest different ways in which the mind might work. The computers that most of us work with today are serial processors, performing one instruction at a time, but the brain and some recently developed computers are parallel processors, capable of doing many operations at once.

If you already know a lot about computers, thinking about the mind computationally should come fairly naturally, even if you do not agree that the mind is fundamentally like a computer. Readers who have never written a computer program but have used cookbooks can consider another analogy. A recipe usually has two parts: a list of ingredients and a set of instructions for what to do with them. A dish results from applying cooking instructions to the ingredients, just as a running program results from applying algorithms to data structures such as numbers and lists, and just as thinking (according to CRUM) results from applying computational procedures to mental representations. The recipe analogy for thinking is weak, since ingredients are not representations and cooking instructions require someone to interpret them. Chapters 2–7 provide simple examples of computational procedures that map much more directly onto the operations of mind.

Theories, Models, and Programs

Computer models are often very useful for theoretical investigation of mental processes. Comprehension of cognitive science models requires noting the distinctions and the connections among four crucial elements: theory, model, program, and platform. A cognitive *theory* postulates a set of representational structures and a set of processes that operate on these structures. A computational *model* makes these structures and processes more precise by interpreting them by analogy with computer programs that consist of data structures and algorithms. Vague ideas about representations can be supplemented by precise computational ideas about data structures, and mental processes can be defined algorithmically. To test the model, it must be implemented in a software *program* in a programming language such as LISP or Java. This program may run on a variety of hardware *platforms* such as Macintoshes, Sun Workstations, or IBM PCs, or it may be specially designed for a specific kind of hardware that has many processors working in parallel. Many kinds of structures and processes can be investigated in this way, from the rules and search strategies of some traditional sorts of artificial intelligence, to the distributed representations and spreading activation processes of newer connectionist views.

Suppose, for example, that you want to understand how children learn to add numbers together in problems such as 13 + 28 = ? A cognitive theory would postulate how children represent these numbers and how they process the representations to accomplish addition. The theory would propose whether 13 is to be represented by a single structure, a combined structure such as *10 plus 3*, or by a complex of neuronlike structures. The theory would also propose processes that operate on the structures to produce a result such as 41, including the carrying operation that somehow turns 30-plus-11 into 41. A computational model would specify the nature of the representations and processes more precisely by characterizing programmable structures and algorithms that are intended to be analogous to the mental representations and processes for addition. To evaluate the theory and model, we can write a computer program in a computer language such as LISP, running the program to compare its performance with human adders and checking that the program not only gets the same right answers as the humans but also makes the same kind of mistakes. Our program might run on any number of different platforms such as PCs, or it might be specially tailored to a particular kind of computer such as one that mimics the neuronal structure of the brain.

The analogy between mind and computer is useful at all three stages of the development of cognitive theories: discovery, modification, and evaluation. Computational ideas about different kinds of programs often suggest new kinds of mental structures and processes. Theory development, model development, and program development often go hand in hand, since writing the program may lead to the invention of new kinds of data structures and algorithms that become part of the model and have analogs in the theory. For example, in writing a computer program to simulate human addition, a programmer might think of a kind of data structure that suggests new ideas about how children represent numbers. Similarly, evaluation of theory, model, and program often involves all three, since our confidence in the theory depends on the model's validity as shown by the program's performance. If the computer program for doing addition cannot add, or if it adds more perfectly than humans, we have reason to believe that the corresponding cognitive theory of addition is inadequate.

The running program can contribute to evaluation of the model and theory in three ways. First, it helps to show that the postulated representations and processes are computationally realizable. This is important, since many algorithms that seem reasonable at first glance do not scale up to large problems on real computers. Second, in order to show not only the computational realizability of a theory but also its psychological plausibility, the program can be applied qualitatively to various examples of thinking. Our addition program, for example, should be able to get the same kinds of right and wrong answers as children. Third, to show a much more detailed fit between the theory and human thinking, the program can be used quantitatively to generate detailed predictions about human thinking that can be compared with the results of psychological experiments. If there are psychological experiments that show that children get a certain percentage of a class of addition problems right, then the computer program should get roughly the same percentage right. Cognitive theories by themselves are normally not precise enough to generate such quantitative predictions, but a model and program may fill the gap between theory and observation.

Box 1.1
Criteria for evaluating theories of mental representation.

(1) Representational power
(2) Computational power
  (a) Problem solving
    (i) Planning
    (ii) Decision
    (iii) Explanation
  (b) Learning
  (c) Language
(3) Psychological plausibility
(4) Neurological plausibility
(5) Practical applicability
  (a) Education
  (b) Design
  (c) Intelligent systems
  (d) Mental illness

Evaluating Approaches to Mental Representations

We can now be more specific about what to expect of a theory of mental representation. Box 1.1 lists five complex criteria for evaluating a particular account of the representations and computations that can be claimed to explain thought. Chapters 2–7 use these criteria to evaluate six different approaches to mental representation: logic, rules, concepts, images, cases, and connections (artificial neural networks).

Each of the approaches described in chapters 2–7 proposes a particular kind of representation and a corresponding set of computational procedures. The first criterion, representational power, concerns how much information a particular kind of representation can express. For example, a university calendar urges: "Once admitted to the University, students are advised to preregister for their courses well in advance of the beginning of lectures." Students who take such advice seriously will need to represent it internally in a form that leads to further inferences, such as the conclusion that they should get over to the registrar's office to sign up for next term's courses. We will see that different proposed kinds of mental representation vary greatly in representational power.

Mental representations are important not only for what they express, but especially for what you can do with them. We can evaluate the computational power of an approach to mental representation in terms of how it accounts for three important kinds of high-level thinking. The first is problem solving: a theory of mental representation should be able to explain how people can reason to accomplish their goals. There are at least three kinds of problem solving to be explained: planning, decision making, and explanation. Planning requires a reasoner to figure out how to get from an initial state to a goal state by traversing various intermediate states. Planning problems include mundane issues such as how to get to the airport before your flight leaves, to the sort of exercise students are commonly posed in their textbooks and their exams. In these questions, students are given some information and need to figure out how to calculate the answer. The starting state involves what the student knows and the information in the problem description, and the goal state includes having an answer. The student has to find a solution by constructing a successful sequence of calculations.

In decision making, people are faced with a number of different means for accomplishing their goals and need to select the best one. For example, a student about to graduate may need to choose among looking for a job, going to graduate school, or attending a professional school such as law or business. Such decisions are very difficult, since they require students to identify their goals and figure out which course of action will best accomplish those goals. In planning problems, the task is to find a successful sequence of actions, whereas in decision problems the task is to choose the best plan from among a number of possible actions.

Explanation problems are ones that require people to figure out *why* something happened. They range from mundane questions such as why a friend is late for dinner, to deep scientific questions such as why human language has evolved. Every minimally intelligent human being is capable of planning, decision making, and generating explanations. A cognitive theory must have sufficient computational power to offer possible explanations for how people solve these kinds of problems.

The computational power of a system of representations and procedures is not just a matter of how much the system can compute, but must also take into account how efficient the computation is. Imagine a procedure that takes only a second to be applied once, but twice as long the second time, and twice as long as that the third time, and so on. Then twenty applications would take 2^{20} seconds, which are more seconds than there have been in the approximately 15 billion years since the universe was formed. Both naturally and artificially intelligent systems need to have sufficient speed to work effectively in their environments.

When people solve a problem, they are usually able to learn from the experience and thereby solve it much more easily the next time. For example, the first time that students register for classes is usually very confusing since they do not know what procedures to follow or how to go about choosing good classes. Subsequently, however, registering typically gets a lot easier. Part of being intelligent involves being able to learn from experience, so a theory of mental representation must have sufficient computational power to explain how people learn. In discussing different approaches to mental representation, we will encounter diverse kinds of human learning, ranging from the acquisition of new concepts such as *registration* and rules such as *Never sign up for an 8:30 class* to more subtle kinds of adjustment in performance.

In addition to problem solving and learning, a general cognitive theory must account for human language use. Ours is the only species on Earth capable of complex use of language. General principles of problem solving and learning might account for language use, but it is also possible that language is a unique cognitive capacity that must be dealt with specially. At least three aspects of language use need to be explained: people's ability to comprehend language, their ability to produce utterances, and children's universal ability to learn language. Different approaches to knowledge representation provide very different answers to how these work.

If artificial intelligence is viewed as a branch of engineering, it can develop computational models of problem solving, learning, and language that ignore how people accomplish these tasks; the question is just how to get computers to do them. But cognitive science has the goal of understanding *human* cognition, so it is crucial that a theory of mental representation not only have a lot of representational and computational power, but also be concerned with how people think. Accordingly, the third criterion for evaluating a theory of mental representation is psychological plausibility, which requires accounting not just for the qualitative capacities of humans but also for the quantitative results of psychological experiments concerning these capacities. Relevant experiments include ones dealing with the same high-level tasks that were discussed under the heading of computational power: problem solving, learning, and language. The difference between this criterion and the last is that a cognitive theory of mental representation must not only show how a task is possible computationally, but also try to explain the particular ways that humans do it.

Similarly, since human thought is accomplished by the human brain, a theory of mental representation must at least be consistent with the results of neuroscientific experiments. Until recently, neurological techniques such as recording EEGs of brain waves seemed too crude to tell us much about high-level cognition, but the past two decades have brought new scanning techniques that can identify where and when in the brain certain cognitive tasks are performed. Cognitive neuroscience has thereby become an important part of reflection on the operations of mind, so we should try to assess each approach to knowledge representation in terms of neurological plausibility, even though information about how the brain produces cognition is still limited (see chapter 9).

The fifth and final criterion for evaluating theories of mental representation is practical applicability. Although the main goal of cognitive science is to understand the mind, there are many desirable practical results to which such understanding can lead. This book considers what each of the approaches to knowledge representation has to tell us about four important kinds of application: education, design, intelligent systems, and mental illness. For educational purposes, cognitive science should be able to increase understanding of how students learn, and also to suggest how to teach them better. Design problems, such as how to make computer interfaces that people like to use, should benefit from an understanding of how people are thinking when they perform such tasks. Developing intelligent systems to act either as stand-alone experts or as tools to support human decisions can directly benefit from computational ideas about how humans think. Different theories of mental representation have given rise to very different sorts of expert computer systems, including rule-based, case-based, and connectionist tools. Other potential practical applications of cognitive science include understanding and treatment of mental illness.

As we will see, no single approach to mental representation fully satisfies all these criteria. Moreover, there are aspects of human thinking such as perception (sight, hearing, touch, smell, taste), emotion, and motor control that are not included in these criteria (see chapters 10–12). Nevertheless, the criteria provide a framework for comparing and evaluating current theories of mental representation with respect to their accomplishments as well as their shortcomings.

Summary

Researchers in psychology, artificial intelligence, neuroscience, linguistics, anthropology, and philosophy have adopted very different methods for studying the mind, but ideally these methods can converge on a common interpretation of how the mind works. A unified view of cognitive science comes from seeing various theoretical approaches as all concerned with mental representations and procedures that are analogous to the representations and procedures familiar in computer programs. The Computational-Representational Understanding of Mind operates with the following kind of explanation schema:

Explanation target

Why do people have a particular kind of **intelligent behavior**?

Explanatory pattern

People have mental **representations**.

People have algorithmic **processes** that operate on those **representations**.

The **processes**, applied to the **representations**, produce the **behavior**.

The words in boldface are placeholders, indicating that to explain various kinds of intelligent behavior, various kinds of representations and processes can be considered. Currently, there are six main approaches to modeling the mind, involving logic, rules, concepts, analogies, images, and neural connections. These can be evaluated according to five criteria: representational power, computational power, psychological plausibility, neurological plausibility, and practical applicability.

The fundamental presuppositions that have guided the writing of this book are:

1. The study of mind is exciting and important. It is exciting for theoretical reasons, since the attempt to investigate the nature of mind is as challenging as anything attempted by science. It is also exciting for practical reasons, since knowing how the mind works is important for such diverse endeavors as improving education, improving design of computers and other artifacts, and developing intelligent computational systems that can aid or replace human experts.

2. The study of mind is interdisciplinary. It requires the insights that have been gained by philosophers, psychologists, computer scientists, linguists, neuroscientists, anthropologists, and other thinkers. Moreover, it requires the diversity of methodologies that these fields have developed.

3. The interdisciplinary study of mind (cognitive science) has a core: the Computational-Representational Understanding of Mind (CRUM). Thinking is the result of mental representations and computational processes that operate on those representations.

4. CRUM is multifarious. Many kinds of representations and computations are important to understanding human thought, and no single computational-representational account now available does justice to the full range of human thinking. This book reviews (in chapters 2–8) the six major current approaches to understanding the mind in terms of representations and computation.

5. CRUM is successful. The computational-representational approach has exceeded all previous theories of mind in its theoretical ability to account for psychological performance and its practical ability to improve that performance.

6. CRUM is incomplete. Not all aspects of human thought and intelligence can be accounted for in purely computational-representational terms. Substantial challenges have been made to CRUM that show the necessity of integrating it with biological research (neuroscience) and with research on social aspects of thought and knowledge.

Discussion Questions

1. What are additional examples of things that students learn when they go to college or university?

2. Why have researchers in different fields adopted different methods for studying the mind?

3. Can you think of any alternatives to the computational-representational understanding of mind?

4. What aspects of human thinking are most difficult for computers to perform or model? What would it take to convince you that a computer is intelligent?

5. Are theories and models in cognitive science like theories and models in physics and other fields?

6. Are there additional criteria that you would want a theory of mental representation to meet?

Further Reading

Three recent reference works contain valuable articles on many aspects of cognitive science: *The MIT Encyclopedia of the Cognitive Sciences* (Wilson and Keil 1999), *A Companion to Cognitive Science* (Bechtel and Graham 1998), and *Encyclopedia of Cognitive Science* (Nadel 2003).

On the history of cognitive science, see Gardner 1985 and Thagard 1992, chap. 9. Other introductions to cognitive science include Johnson-Laird 1988, Stillings et al. 1995, Dawson 1998, and Sobel 2001. General collections of articles include Polk and Seifert 2002 and Thagard 1998.

Textbooks on cognitive psychology include Anderson 2000, Medin, Ross, and Markman 2001, and Sternberg 2003. For introductions to artificial intelligence, see Russell and Norvig 2003 and Winston 1993. Graham 1998 and Clark 2001 provide introductions to the philosophy of mind and cognitive science. An introductory linguistics text is Akmajian et al. 2001. For accessible introductions to cognitive neuroscience, see LeDoux 2002 and Kosslyn and Koenig 1992; Churchland and Sejnowski 1992 present a more computational approach. D'Andrade 1995 provides an introduction to cognitive anthropology.

Web Sites

Note: Live links to all the sites mentioned in this book can be found at my own Web site, http://cogsci.uwaterloo.ca/courses/resources.html.

Artificial Intelligence in the news (American Association for Artificial Intelligence): http://www.aaai.org/AITopics/html/current.html

Artificial intelligence on the Web: http://aima.cs.berkeley.edu/ai.html

Biographies of major contributors to cognitive science: http://mechanism.ucsd.edu/~bill/research/ANAUT.html

Cognitive Science dictionary, University of Alberta: http://web.psych.ualberta.ca/~mike/Pearl_Street/Dictionary/dictionary.html

Cognitive Science Society: http://www.cognitivesciencesociety.org/

Cogprints (archive of papers on cognitive science): http://cogprints.ecs.soton.ac.uk/

Dictionary of Philosophy of Mind: http://www.artsci.wustl.edu/~philos/MindDict/

Science Daily (mind and brain news): http://www.sciencedaily.com/news/mind_brain.htm

Yahoo! Cognitive Science page: http://dir.yahoo.com/Science/cognitive_science/

Notes

Discussions of thinking as computation often begin with an abstract model of computation such as the Turing machine, a simple device that consists of a tape and a mechanical head that can write symbols on spaces on the tape. Although it can be proven mathematically that such a machine can in principle do anything that any other computer can, the Turing machine is an excessively abstract analog of thinking, which is much better discussed in terms of higher-level computational ideas such as data structures and algorithms.

For more on explanation schemas and patterns, see Kitcher 1993, Leake 1992, Schank 1986, and Thagard 1999.

2 Logic

Although formal logic has not been the most influential psychological approach to mental representation, there are several reasons for beginning our survey with it. First, many basic ideas about representation and computation have grown out of the logical tradition. Second, many philosophers and artificial intelligence researchers today take logic as central to work on reasoning. Third, logic has substantial representational power that must be matched by other approaches to mental representation that may have more computational efficiency and psychological plausibility.

Formal logic began with the Greek philosopher Aristotle more than two thousand years ago. He systematically studied such inferences as

All students are overworked.

Mary is a student.

So, Mary is overworked.

Such patterns of inference, with two premises and a conclusion, are called *syllogisms*. In addition to cataloging many different kinds of syllogism, Aristotle showed how they can be analyzed purely in terms of their form. For the conclusion in the example to follow from the two premises, it does not matter that the syllogism is about overworked students. We can substitute "sausage" for "student," "orange" for "overworked," and "Marvin" for "Mary," and the conclusion that Marvin is orange follows from the revised premises even if it makes little sense. Aristotle initiated the use of symbols to show the form of the inference:

All *S* are *O*.

M is *S*.

So, *M* is *O*.

Aristotle's discovery of how to analyze syllogisms purely in terms of their form, ignoring their content, has had a major influence on logic. The discovery's usefulness, however, has been challenged from a psychological perspective, as we will see below in the section on psychological validity.

The syllogism is a form of *deductive* inference, in which the conclusion follows necessarily from the premises: if the premises are true, the conclusion is true also. *Inductive* inference is more dangerous since it introduces uncertainty. If all the students you know are overworked, you might inductively infer that all students are overworked. But your conclusion might well be erroneous—for example, if there are basket-weaving majors you do not know who take it easy.

Although the syllogism dominated discussions of formal logic for two thousand years, it is not sufficient to represent all inferences. Syllogisms are fine for simple predicates like "is a student" but they can not handle relations such as *take* in sentences like "Students who take courses get credit for them." Here *take* is a relation between a student and a course. Modern logic began in 1879 with the work of the German mathematician Gottlob Frege (1960), who devised a formal system of logic much more general than Aristotle's. Subsequently, Bertrand Russell and many other logicians have found ways of increasing the representational and deductive power of formal logic.

The early theory of computation was developed by logicians such as Alonzo Church and Alan Turing. In the 1930s, Church, Turing, and others developed mathematical schemes for specifying what could be effectively computed. These schemes turned out to be mathematically equivalent to each other, providing support for the thesis that the intuitive concept of effective computability can be identified with well-defined mathematical concepts such as Turing-machine computability. When digital computers became available in the late 1940s and 1950s, the mathematical theory of computability provided a powerful tool for understanding their operations. It is not surprising that, when artificial intelligence began in the mid-1950s, mathematically trained researchers such as John McCarthy took logic to be the most appropriate tool. We shall see, however, that other pioneers such as Allen Newell, Herbert Simon, and Marvin Minsky preferred different approaches.

Representational Power

Modern formal logic has the resources to represent many kinds of deductive inferences. The simplest system of formal logic is propositional logic, in which formulas like "*p*" and "*q*" are used to stand for sentences such as "Paula is in the library" and "Quincy is in the library." Simple formulas can be combined into more complex ones using symbols such as "&" for "and," "v" for "or," and "→" for "if-then." For example, the sentence

If Paula is in the library, then Quincy is in the library.

becomes

$p \rightarrow q.$

Such if-then sentences are called conditionals, consisting of antecedents (the "if" part) and consequents (the "then" part). To express negation, "*not-p*" can be written $\sim p$. From these building blocks we can construct formalizations for complex statements such as "If Paula or Quincy is in the library, then Debra is not," which can be formalized as

$(p \vee q) \rightarrow \sim d.$

Here, "*p*" stands for "Paula is in the library," "*q*" stands for "Quincy is in the library," and "*d*" stands for "Debra is in the library."

More complicated logics have been developed that allow different kinds of propositional operators. Modal logic adds operators for necessity and possibility, so that we can represent statements such as "It is possible that Paula is in the library." Epistemic logic adds operators for knowledge and belief, so that *Kp* represents "It is known that *p*." Deontic logic represents moral ideas such as that *p* is permissible or forbidden.

Propositional logic requires treating statements such as "Paula is a student" as an indivisible whole, but predicate logic allows us to break them down. Predicate calculus distinguishes between predicates such as "is a student" and constants referring to such individuals as Paula or Quincy. In the version of predicate calculus usually taught in philosophy courses, "Paula is a student" is formalized as "*S(p)*," where "*p*" now stands for Paula rather than a whole proposition. Computer scientists tend to express this more mnemonically as "is-student (paula)." In addition to simple properties, predicates can be used to express relations between two or more things. For example, "Paula takes Philosophy 256" becomes: "takes (Paula, Phil256)."

Predicate calculus can formalize sentences with quantifiers such as "all" and "some" by using variables such as "*x*" and "*y*." For example, "All students are overworked" becomes

(for-all *x*) (student(*x*) → overworked (*x*)).

Literally, this says "For any *x*, if *x* is a student, then *x* is overworked," which is equivalent to saying that all students are overworked. The sentence "Students who take courses get credit for them" could be formalized as

(for-all *x*) (for-all *y*) [(student (*x*) & course (*y*) & take (*x*, *y*)) → get-credit-for (*x*, *y*)]

This looks complicated, but what it is saying in English is "For any *x* and *y*, if *x* is a student, *y* is a course, and *x* takes *y*, then *x* gets credit for *y*."

Readers whose interest lies predominantly in human psychology might now be asking, why are you throwing these mathematical symbols at me? The answer is that some rudiments of formal logic are required for understanding much current work in cognitive science, including some proposals about how humans do deduction. At a minimum, we have to notice that people can comprehend such statements as "Students who pass courses get credit for them" and use them to make inferences. Predicate logic, unlike some other approaches to representation we will discuss, has sufficient representational power to handle this example.

Although predicate logic is useful for many purposes, it has limitations that become obvious as soon as we try to translate a natural language text. For example, try to put the last paragraph into logical form. Its first sentence includes the word "now," and extending predicate logic to deal with time is not an easy matter. It also contains the word "you," which the reader can figure out refers to Paul Thagard, the author of this book, but it is not obvious how to express this in logic. Moreover, the structure of this sentence includes the relation "asks," which involves both an asker and the proposition that is asked, so that we need to be able to embed a proposition within a proposition, which is not naturally done in the usual formalism for predicate logic. If translation from language to logical formalism were easier, we could have greater confidence that formal logic captures everything that is necessary for mental representation.

Propositional and predicate logic work well for making assertions that take statements to be true or false, but they provide no means to deal with uncertainty, as in "Paula is probably in the library." For such assertions, formal logic can be supplemented with probability theory, which assigns numbers between 0 and 1 to propositions. We can then write "$P(p) = 0.7$" to symbolize that the probability that Paula is in the library is 0.7.

Computational Power

Representations by themselves do nothing. To support thinking, there must be operations on the representations. To derive a conclusion in logic, we apply *rules of inference* to a set of premises. Two of the most common rules of inference make it possible to draw conclusions using conditionals (if-then sentences):

Modus ponens

$p \rightarrow q$

p

Therefore, q.

Modus tollens

$p \rightarrow q$

not-q

Therefore, *not-p*.

From the conditional "If Paula is in the library, then Quincy is in the library" and the information that Paula is in the library, modus ponens enables us to infer that Quincy is in the library. From the information that Quincy is not in the library, it follows by modus tollens that Paula is not in the library.

In predicate logic, there are rules of inference for dealing with the quantifiers "all" and "some." For example, the rule of universal instantiation allows the derivation of an instance from a general statement, licensing the inference from (for-all *x*)(cool (*x*)) to cool(Paula), that is, from "Everything is cool" to "Paula is cool." A more complicated application applies the generalization that all students are overworked: (for-all *x*) (student(*x*) → overworked (*x*)). Applying this to Mary, we get the conclusion that if Mary is a student, she is overworked: student (Mary) → overworked (Mary).

Abstract rules of inference such as modus ponens are not in themselves processing operations. To produce computations, they need to be part of a human or machine system that can apply them to sentences with the appropriate logical form. From a logical perspective, deductive reasoning consists of applying formal inference rules that consider only the logical form of the premises.

Problem Solving

Planning Many planning problems are open to solutions that employ logical deduction. Suppose Tiffany is a student who wants to get a degree in psychology. Her college or university catalog tells her that she needs to take ten psychology courses, including two statistics courses, Statistics 1 and Statistics 2. The first of these is a prerequisite for the other, and the second is a prerequisite for Research Methods, which is also required for the degree. From the general description in the catalog, Tiffany can infer by the inference rule universal instantiation the conditionals that apply to her, including

take (Tiffany, Stat1) → can-take (Tiffany, Stat2)

can-take (Tiffany, Stat2) & open (Stat2) → take (Tiffany, Stat2)

take (Tiffany, Stat2) → can-take (Tiffany, RM)

can-take (Tiffany, RM) & open (RM) → take (Tiffany, RM)

take (Tiffany, RM) & take (Tiffany, Stat1) & take (Tiffany, Stat2) & take (Tiffany, seven-other-courses) → graduate-with (Tiffany, psychology-degree).

The last conditional is a somewhat awkward formalization of the statement that if Tiffany takes Research Methods, the two statistics courses, and seven other courses, then she can graduate with a psychology degree. Tiffany can use these conditionals and the inference rule modus ponens to derive a plan, which in logical terms is a deduction from her initial state, where she has taken no psychology courses, to the goal state, where she graduates. Tiffany can construct the deductive plan that she can take Statistics 1, and then Statistics 2, and then Research Methods, and then seven other courses, and finally graduate with a psychology degree.

For planning to be computationally realizable, deduction must be more constrained than the general set of inference rules found in formal logic. For example, propositional logic contains the following conjunction rule:

Conjunction

p

q

Therefore, *p* & *q*.

This rule is fine logically, but computationally it is potentially disastrous. If Tiffany has taken both statistics courses, she could usefully infer

take (Tiffany, Stat1) & take (Tiffany, Stat2).

But it would gain her nothing to make the additional valid inference to

take (Tiffany, Stat1) & take (Tiffany, Stat2) & take (Tiffany, Stat1) & take (Tiffany, Stat2).

Uncontrolled inference of this sort would quickly exhaust the memory of any human or machine system.

The deductive method of planning is intuitively appealing, but it encounters a number of computational problems. First, it tends to be slow, with an enormous amount of inference required to accomplish even simple plans, although various computational strategies have been developed to make deduction more effective. Second, purely deductive planning is *monotonic*: it can only draw new conclusions and not reject previous ones. (A monotonic mathematical function is one whose values continuously increase or continuously decrease without oscillation; reasoning is not monotonic because we do not continuously add new beliefs, since sometimes old beliefs must be abandoned.) AI researchers have developed several techniques to make logic nonmonotonic, but they are computationally expensive. Third, a purely deductive planner is not capable of learning from experience. Having solved a problem once, it will go through the same laborious deductive process when faced with it again, unless some method of learning from its experience has been added.

I have barely scratched the surface in describing artificial intelligence work on deductive planning (see, for example, Dean and Wellman 1991; Russell and Norvig 2003). The reader should see that logical deduction can be a useful way to describe how planning problems are solved, but that this view of planning has some difficulties. Later chapters will describe numerous other approaches to planning.

Decision Deductive planning finds a logical path from an initial state to a goal state. But what happens if there is more than one reasonable path?

In the example in the last section, Tiffany's deduced plan was to take Statistics 1, then Statistics 2, then Research Methods. But often she will face decisions that require her to choose between actions. For example, she may be required to take a humanities course, and therefore will have to choose among philosophy, English, and Spanish. Deductive planning will not tell her which choice to make, since each path will take her to the desired goal state of satisfying the humanities requirement. Tiffany needs to decide which of the courses will satisfy her other goals, such as learning something interesting, not working too hard, and taking a course that fits reasonably with the rest of her schedule. Deduction may be relevant to working out the consequences of some possible choices. If Spanish is only offered at 8:30 in the morning, Tiffany might deduce that she would have to get up early in the morning if she took it. But other consequences might not be so clear, since students are often not sure about what a course will be like.

Hence, decision making often requires considering of probabilities. Tiffany might believe that philosophy will probably be more interesting than English, or vice versa, or that Spanish will probably be more useful than English. Hence, she needs to base her decision both on what her goals are and on her estimated probabilities that the actions will accomplish those goals. There is thus room for judgments that apply the formal theory of probability. We can write $P(p/q)$ to represent the probability of p given q, so that "P(interesting course/English course)" could express the probability that Tiffany gets an interesting course given that she takes an English course. To estimate this probability, she could use her background knowledge of what proportion of English courses on her campus are interesting. In deciding whether to take philosophy, English, or Spanish, Tiffany will have to calculate the *expected value* of each choice, taking into account both the probability of various outcomes and the extent to which her goals are satisfied by the outcomes.

Computational systems for decision making based on probabilities have been developed. Holtzman (1989) used probability theory and other formal ideas to develop an intelligent decision system for helping infertile couples decide what kind of treatment to use. Developing probabilistic computer systems is tricky, because using probability can be computationally explosive: the number of probabilities needed can increase exponentially as the number of propositions or variables in the model increase. Clever techniques have been developed for keeping probabilistic reasoning computationally tractable (Neapolitan 1990; Pearl 1988, 2000). A different issue treated below is whether people's normal decision making uses probabilities.

Explanation Whereas in a planning problem you are trying to figure out how to accomplish a goal, in an explanation you are trying to understand why something happened. Suppose that Sarah was expecting to meet Frank at the student bar, but he did not show up. She would naturally try to generate an explanation for his absence. Like plans, explanations can sometimes be viewed as logical deductions: you can try to deduce what you want to explain for what you know. Someone might tell Sarah that Frank is studying for an exam, and that whenever he studies he forgets about social engagements. From this information Sarah can deductively explain why Frank did not show up.

The view that explanations are logical deductions was developed and defended by the philosopher of science Carl Hempel (1965). Especially in mathematical areas of science such as physics, explanations can be described as logical deductions. We shall see in later chapters, however, that not all explanations are deductive. Moreover, not all deductions are explanations. For example, we can deduce the height of a flagpole from information about its shadow along with trigonometry and laws of optics, but it seems odd to say that the length of a flagpole's shadow explains the flagpole's height.

In rare cases, the reason Frank did not show up could be deduced—for example if he is a rigid person who misses appointments if and only if he is sick. Sarah could then apply modus ponens: if Frank misses an appointment, he is sick; Frank missed an appointment; therefore Frank is sick. But normally there will more than one explanation available. Just like a planner constructing multiple paths to a goal, Sarah might be able to construct several deductive explanations based on conditionals such as

If Frank is sick, then he will not arrive.

If Frank has had a car accident, then he will not arrive.

If Frank has fallen in love with someone else, then he will not arrive.

If Sarah did not actually know that Frank is sick, or that he has had a car accident, or that he has fallen in love, then she would not immediately be able to deduce that he will not arrive. But the three conditionals just given can be used to form hypotheses about what happened: maybe he's sick, or maybe he had a car accident, or maybe he has fallen in love. This kind of inference, where you form a hypothesis in order to generate an explanation, was called *abduction* by the nineteenth-century American philosopher Charles Peirce (1992). Sarah may abduce that Frank is sick because this hypothesis, in conjunction with the rule that if Frank is sick he will not arrive, allows her to deductively explain why Frank did not arrive. Abductive inference is a risky but powerful kind of learning.

Learning

Intelligent systems should be able not only to solve various kinds of problems but also to use experience to improve their performance. How can we improve planning, decision making, and explanation? Little work has been done within the logical approach on direct improvements to problem solving, but logical representations are useful for describing some kinds of learning programs.

Consider the learning problem faced by students first arriving on campus. They usually start with little knowledge about the kinds of course offerings available or the kinds of people they will meet. But they quickly accumulate information about particular examples of courses or types of people and naturally proceed to make inductive *generalizations* about them. Crude generalizations might include such statements as that philosophy classes are fun (or boring, as the case may be) and that statistics classes are demanding. These generalizations are inductive in that they involve uncertainty, a leap from what is definitely known to what is at best probable. Students who have taken two philosophy classes might be prepared to generalize from information that could be expressed in logical form as follows:

fun (Phil100)

fun (Phil200)

Therefore, (for-all x) (philosophy-course (x) → fun (x)).

The conclusion is that all philosophy courses are fun. But it is obviously possible that these two courses might be fun whereas other philosophy courses (e.g., Philosophy of Basket Weaving) are boring.

Computer programs for inductive generalization do not always use logical representations for input. One of the most widely used learning

programs is Quinlan's (1983) ID3 program. It can be classified as within the logical approach because it uses probabilities to form generalizations from sets of instances. For example, it could be given a sample of students from different sections of a university along with a description of their traits. It could then start to form generalizations concerning how students from such areas as arts, sciences, and engineering differ with respect to personal, social, and intellectual characteristics.

Like inductive generalization, but unlike deduction, abduction is obviously a very risky sort of inference. There may be all sorts of reasons unknown to Sarah that explain why Frank did not show up for an appointment with her. But abduction is indispensable in science and everyday life, whether paleontologists are trying to generate explanations of why the dinosaurs became extinct or students are trying to understand their friends' behavior. Since abduction's purpose is to generate explanations, and explanations can sometimes be understood in terms of logical deduction, it is natural to treat abduction within a logical framework (e.g., Konolige 1992). Later chapters describe alternative ways of thinking about abduction.

Sarah does not want to find just *some* explanation of why Frank did not arrive, she wants to find the *best* explanation. From a logical perspective, assessing the best explanation involves probabilities. Sarah will want to be able to assess the conditional probability of Frank being sick, given that he did not arrive, as well as the conditional probabilities of all the other hypotheses. A theorem of the probability calculus, Bayes's theorem, is potentially very useful. In words, it says that the probability of a hypothesis given the evidence is equal to the result of multiplying the prior probability of the hypothesis, $P(h)$, by the probability of the evidence given the hypothesis, all divided by the probability of the evidence. For Sarah, the prior probability that Frank is sick is her estimate of how likely he is to be sick in general, without considering his failure to arrive. To apply Bayes's theorem, she also needs to consider the probability of his failure to arrive, assuming he is sick. Probabilistic approaches to the problem of how to choose explanatory hypotheses have been popular in both artificial intelligence (Pearl 1988, 2000) and philosophy (Howson and Urbach 1989; Glymour 2001). But alternative approaches are available, as we will see in chapter 7.

The term "induction" can be very confusing, since it has both a broad and a narrow sense. The broad sense covers any inference that, unlike deduction, introduces uncertainty. The narrow sense covers only inductive generalization, in which general conclusions are reached from particular examples. Abduction (forming explanatory hypotheses) is induction in the broad sense but not in the narrow one. My practice in this book is to use "learning" for the broad sense of induction and "inductive generalization" for the narrow sense. Additional computational accounts of learning will be encountered in later chapters.

Language

Linguists have sometimes taken formal logic to be a natural tool for understanding the structure of language. There are even two editions of a book called *Everything That Linguists Have Always Wanted to Know about Logic—But Were Ashamed to Ask* (McCawley 1993). The philosopher Richard Montague (1974) contended that there are no important theoretical differences between natural languages and the artificial languages of logicians. Most linguists and psychologists would disagree with this claim, however, and formal logic has played a minor role in the understanding of human language. Stabler (1992) has used logic to formalize some of Chomsky's recent ideas about language, which include the postulation of a level of "logical form" at which meaning is most explicitly represented (Chomsky 1980). Later chapters discuss how other kinds of representation, particularly rules and concepts, have been used to describe and explain human use of language.

Psychological Plausibility

Historically, logicians have disagreed about the mutual relevance of logic and psychology. Some early writers on logic, such as John Stuart Mill, saw an intimate connection between human psychology and logic, which was construed as the art and science of reasoning. In contrast, the founders of modern formal logic, Gottlob Frege and Charles Peirce, emphatically distanced their work from psychology. Today, we can distinguish at least three positions concerning the relations and relative merits of formal logic and psychology:

1. Formal logic is an important part of human reasoning.

2. Formal logic is only distantly related to human reasoning, but the distance does not matter, since the role of logic in philosophy and artificial intelligence is to provide a mathematical analysis of what constitutes optimal reasoning.

3. Formal logic is only distantly related to human reasoning, so cognitive science should pursue other approaches.

The first position is advocated by a few psychologists who have provided experimental evidence that people use rules like modus ponens. The second position is popular among philosophers and artificial intelligence researchers who prefer formal approaches. The third position is probably now the dominant view in psychology, but is less popular in philosophy and artificial intelligence.

The psychologists who have most aggressively defended the first position are Martin Braine (1978; Braine and O'Brien 1998) and Lance Rips (1983, 1986, 1994). Rips (1986, 279) lists several kinds of psychological evidence for mental logic. Theories of mental logic successfully predict the validity judgments that subjects give for a fairly wide range of propositional arguments. For example, people recognize as valid arguments that have the same form as modus ponens, but reject arguments of the form "If A, then C; C, therefore, A." Theories of mental logic also account for reaction times and help make sense of what subjects say when they think aloud about validity decisions.

Nevertheless, other kinds of experiments have made many psychologists skeptical about mental logic. The best-known experimental technique uses Wason's (1966) selection task, in which subjects are informed that they will be shown cards that have numbers on one side and letters on the other. They are then given a rule such as *If a card has an A on one side, then it has a 4 on the other*. The subjects are then shown four cards and asked to indicate exactly which cards must be turned over to determine whether the rule holds. They can be given, for example, the four cards shown in figure 2.1. Then they must decide which of these cards should be turned over. Most people realize that it is necessary to turn the A over to check whether it has a 4 on the other side. This can be interpreted as an application of modus ponens, since the rule *If A then 4* combined with the premise A suggests checking to see if there is a 4. On the other hand, a great many people neglect to check the 7, failing to realize that if this card has an A on the other side, it refutes the rule in question. Recognition that the card with a 7 needs to be turned over requires an appreciation of modus tollens:" If A then 4; 7 means not-4; so not-A is required for the rule to hold." Some

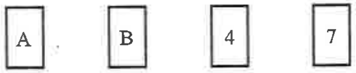


Figure 2.1
Cards in Wason's selection task.

people are confused enough about the task to turn over the cards with B and 4 on them, even though these are irrelevant to determining the truth of the rule.

The point of this kind of experiment is not to show that people are stupid in violating the rules of formal logic. Rather, the experiment becomes interesting if it suggests that people approach this kind of reasoning task with representations and computations quite different from those used in formal logic. Subsequent experiments have shown that people have little difficulty with tasks like Wason's card problem if they are given familiar concrete examples. Suppose that people are told that the cards have on one side information about whether individuals are in a bar and on the other side numbers representing their ages. They can then be given a rule such as *If a person is in the bar, then he or she is over 21*. They can then be asked what cards need to be turned over to determine whether this rule holds, choosing, for example, from IN-BAR, NOT-IN-BAR, 23, 18. In contrast to the way they perform on abstract problems with letters and numbers, most people can recognize that it is necessary not only to turn over the IN-BAR card to check the age of the person, but also to turn over the 18 card to make sure that the person is not in the bar.

Cheng and Holyoak (1985) have argued that people approach these tasks, not with mental logic, but with *pragmatic reasoning schemas*. For example, a permission schema has the form *If one is to do X, then one must satisfy precondition Y*. Then the reason that people do so much better with the concrete bar-and-age example than with the abstract letter-and-number example is that the permission schema is naturally applied to the former. The psychological application of rules and schemas is discussed further in chapters 3 and 4.

The most persistent critic of the mental logic view has been the psychologist Philip Johnson-Laird (1983). Johnson-Laird and Byrne (1991) argue that deductive reasoning is carried out neither by formal logical rules nor by content-specific rules or schemas, but by *mental models*, which are mental representations that correspond in structure to the situations that they represent. Johnson-Laird and Byrne claim that when people interpret a conditional such as "If a card has an A on one side, then it has a 4 on the other," they construct a mental representation something like this:

[A] 4

Here "[A]" indicates a model in which a card has an A on it, and "4" adds that in this model it also has a 4 on the other side. Johnson-Laird and Byrne explain many people's performance in the selection task by supposing that they consider only those cards that are explicitly represented in their models of the rule. Hence, people turn over the A card because it is represented in the model they have constructed, but fail to turn over the 7 card because it is not represented.

The theory of mental models has also been applied to many kinds of reasoning with the quantifiers "all" and "some." From a formal logic perspective, reasoning with quantifiers proceeds by first using inference rules such as universal instantiation (presented above) to remove quantifiers, then using propositional rules of inference such as modus ponens to make inferences, and finally reapplying quantifiers using additional rules of inference. Consider the simple reasoning:

All football players are strong.
Anyone strong can lift heavy objects.
Therefore, all football players can lift heavy objects.

In logic, it can be confirmed that this is a valid form of inference by instantiating it into nonquantified statements such as "If *x* is a football player, then *x* is strong" and "If *x* is strong then *x* can lift heavy objects." Propositional logic then yields "If *x* is a football player, then *x* can lift heavy objects," which can be generalized by a deductive inference rule to hold for any x. In contrast, Johnson-Laird maintains that people work with models rather than abstract forms, constructing the following sort of model:

football-player strong lifts-heavy-objects

In the model constructed, there are no football players who cannot lift heavy objects, so the conclusion that all football players can lift heavy objects goes through. More complicated kinds of inference with mixtures of "all" and "some" and "not" require more complex kinds of models.

Johnson-Laird argues that the comparative difficulties that people have with different kinds of inferences of this sort correspond exactly to the complexity of different kinds of models that have to be constructed. Rips (1994) and O'Brien, Braine, and Yang (1994) have responded with arguments that mental logic accounts for the psychological evidence about deductive inference better than mental models do. But mental model theory has been applied to many kinds of human thinking, including causal reasoning (Goldvarg and Johnson-Laird 2001).

Just as Johnson-Laird has challenged the relevance of formal logic to human deductive reasoning, psychologists have done experiments that suggest that human inductive reasoning may not have much to do with probability theory. Tversky and Kahneman (1983), for example, have shown that people sometimes violate the rule that the probability of a conjunction will also be less than or equal to the probability of one its conjuncts, $P(p \& q) \leq P(p)$. Suppose you are told that Frank likes to read a lot of serious literature, attend foreign movies, and discuss world politics. You are then asked to estimate the probability that Frank is college educated, that Frank is a carpenter, and that Frank is a college-educated carpenter. Not surprisingly, people in experiments like this one tend to judge it to be more probable that Frank is college educated than that he is a carpenter, but they often violate probability theory by judging it to be more likely that Frank is a college-educated carpenter than that he is a carpenter. When people approach such examples, they seem to employ a kind of matching process that judges the degree of fit between the description of the individual and their stereotypes such as college-educated and carpenter (see chapter 4). Numerous other instances have been found where people's inductive reasoning appears to be based on something other than formal rules of probability theory (Kahneman, Slovic, and Tversky 1982; Gilovich, Griffin, and Kahneman 2002). However, just as Rips and others have defended mental deductive logic, some psychologists have offered different interpretations of Tversky and Kahneman's results that are consistent with the view that people employ probabilistic reasoning (Gigerenzer, Hoffrage, and Kleinbölting 1991; Gigerenzer 2000).

One open possibility is that mental logic may give an appropriate account of some narrow kinds of human reasoning such as applying modus ponens, whereas more vivid representations such as mental models are needed to account for more complex kinds of human reasoning such as that involving "all" and "some." It is at least obvious that the logical approach is not the only possible way of understanding human thinking, and various alternatives are discussed in the chapters to come. Of course, philosophers and artificial intelligence researchers not interested in psychology can maintain that whether or not people use logic in their thinking is less important than developing formal logical models of how people and other intelligent systems *should* think. What they risk missing is the appreciation that human intelligence and the kind of machine intelligence we want to build may rest on representational structures and computational processes that differ markedly from those that logic affords.

Neurological Plausibility

Until recently, little was known about the neurological plausibility of formal logic. Metaphorically, every synaptic connection between neurons looks like a miniature inference using modus ponens: if neuron 1 fires, then neuron 2 fires. Neuron 1 fires, so neuron 2 fires. However, it is obvious that single neurons do not represent whole propositions, and how groups of neurons perform inferences is unknown. However, it is now possible to investigate at a larger scale how the brain performs deductive reasoning. Brain scanning experiments are being used to determine whether people perform deductions using just the left half of the their brains, as suggested by the mental logic view that deduction is formal and independent of content. The alternative hypothesis is that people perform deductions using the right half of their brains, as suggested by the mental models view that deduction requires regions in the right hemisphere of the brain that involve spatial reasoning (Wharton and Grafman 1998). (See chapter 8 for an introduction to how brain scanning is used to identify neural correlates of different kinds of thinking.)

Goel et al. (1998) used brain scans to identify regions involved in reasoning tasks such as syllogisms. They found no significant right-hemisphere activation, suggesting that deductive reasoning is purely linguistic as implied by the mental logic theory. However, Kroger, Cohen, and Johnson-Laird (forthcoming) compared brain regions involved in logical reasoning and mathematical calculation and found that parts of the right half of the brain were more active in reasoning than in calculation. They judged that their results are incompatible with a purely linguistic theory of logical reasoning based on formal rules of inference. Goel (2003) reviewed several neuroimaging studies of syllogistic reasoning and argued that it involves two neural pathways, including both linguistic and visual-spatial systems. The debate between mental logic and mental model accounts of deductive reasoning now involves three of the methodologies of cognitive science: psychological experiments, computational models, and neurological experiments.

Practical Applicability

The logical approach to cognitive science has not been of great educational use from the perspective of providing deeper understanding of human learning. Piaget and Inhelder (1969) tried to base some of the principles of human cognitive development on logical categories, but claims about the role of propositional logic in developmental stages are not part of modern educational theory. Logic is, however, useful from another educational perspective, in that it can suggest ways that people should reason better. Courses on informal logic and critical thinking have proliferated because of the perceived need to improve people's reasoning. Formal deductive logic and probability theory certainly do provide useful tools for prescribing how some kinds of thinking should be done.

According to Dym and Levitt (1991), engineering design often involves satisfying requirements that may be expressed as logical statements. For example, a structural code may state "If a beam is simply supported, its depth shall be greater than one-thirtieth of its clear span." PROLOG, a programming language that uses logic representations and deductive techniques, has been applied to problems such as designing buildings that satisfy physical and legal constraints. Levesque et al. (1997) have developed a logic-based programming language intended for applications in high-level control of robots and industrial processes. Although logic has been a favored tool of artificial intelligence theorists, practical intelligent systems have tended to use techniques such as rules, cases, and neural networks discussed in later chapters. However, there is a growing use of probabilistic reasoning in intelligent systems, for example, in a tutoring computer program that deals with uncertainty about the knowledge and goals of the students it teaches (Conati, Gertner, and Vanlehn 2002).

Summary

Formal logic provides some powerful tools for looking at the nature of representation and computation. Propositional and predicate calculus serve to express many complex kinds of knowledge, and many inferences can be understood in terms of logical deduction with inference rules such as modus ponens. The explanation schema for the logical approach is as follows:

Explanation target

Why do people make the inferences they do?

Explanatory pattern

People have mental representations similar to sentences in predicate logic.

People have deductive and inductive procedures that operate on those sentences.

The deductive and inductive procedures, applied to the sentences, produce the inferences.

It is not certain, however, that logic provides the core ideas about representation and computation needed for cognitive science, since more efficient and psychologically natural methods of computation may be needed to explain human thinking.

Discussion Questions

1. What do you know that is hard to express in formal logic?
2. Are people logical? Should they be?
3. Is deduction a central kind of human thinking? How do people make deductions?
4. Is nondeductive reasoning done in accord with the laws of probability?
5. Is natural language based on logic?

Further Reading

On the history of logic, see Prior 1967. There are many good introductory logic textbooks; for example, Bergmann, Moor, and Nelson 2003. Pollock 1995 approaches philosophical problems from a computational perspective based on formal logic. The logical approach to artificial intelligence is expounded in Genesereth and Nilsson 1987 and Russell and Norvig 2003. Rips 1994 develops and defends a logical approach to human deductive reasoning.

Web Sites

Introduction to logic: http://people.hofstra.edu/faculty/Stefan_Waner/RealWorld/logic/logicintro.html

Logic programming: http://www.afm.sbu.ac.uk/logic-prog/

Mental models: http://www.tcd.ie/Psychology/Ruth_Byrne/mental_models/index.html

Notes

Formal logic is concerned not only with syntax, the structure of sentences that I have described in this chapter, but also with semantics, the truth conditions of sentences. For example, the conjunction p & q is true just in case p is true and q is true, and false otherwise.

Most logic-based planners in artificial intelligence do not use a full set of logical inference rules, but instead use an inference procedure based on a simple rule of inference known as the *resolution principle* (Genesereth and Nilsson 1987). Resolution is too complicated to explain in detail here, but what it does is take expressions that have been translated into a simplified version of predicate calculus and apply a powerful kind of operator to them to see what can be deduced. Fikes and Nilsson (1971) used logical deduction in a planning system called STRIPS that has been applied to robotics and other applications.

In symbols, Bayes's theorem can be written

$$P(h/e) = \{P(h) * P(e/h)\}/ P(e).$$

In artificial intelligence research, logic and probability are often considered to be alternative approaches to knowledge representation, but I have combined them because they both base inference on highly general and abstract principles.

This chapter has focused on one approach to mental models, but other kinds of model-based reasoning are also important (Magnani, Nersessian, and Thagard 1999).

3 Rules

Rules are if-then structures such as: *IF you pass forty Arts courses, THEN you graduate with a B.A.* These structures are very similar to the conditionals discussed in chapter 2, but they have different representational and computational properties. Whereas most logic-based computational models have not been intended as models of human cognition, rule-based models have had psychological aims from the start. The first artificial intelligence program was the Logic Theorist of Allen Newell, Cliff Shaw, and Herbert Simon (1958). Written in 1956 on a primitive computer, this program did proofs in formal logic. Its proving behavior was intended not just as a mathematically sophisticated intelligent system, but also as a model of how humans do proofs in logic. In addition to logical rules of inference, the Logic Theorist included strategic rules for finding proofs efficiently. The Logic Theorist was soon generalized into the first broad framework for understanding human thinking, GPS (the General Problem Solver; Newell and Simon 1972). GPS used rules to simulate human solutions to various kinds of problems, such as cryptarithmetic problems described later in this chapter. In artificial intelligence, rules are often called productions.

Since GPS, two different rule-based cognitive systems have had a substantial impact on cognitive science because of their broad applicability to human cognition. The ACT system of John Anderson (1983, 1993) has had a wealth of psychological applications. More recently, Allan Newell, in collaboration with John Laird and Paul Rosenbloom, developed SOAR, a powerful rule-based program that has had many technological and psychological applications (Newell 1990; Rosenbloom, Laird, and Newell 1993).

The thrust of this chapter is not to describe any of these systems in detail, but rather to convey what makes rules so computationally and

psychologically powerful. Later chapters, however, will provide alternative views of cognition that suggest that rules do not tell the whole story about human thinking.

Representational Power

Although rules have a very simple structure, with just an IF part (sometimes called the *condition*) and a THEN part (called the *action*), they can be used to represent many different kinds of knowledge. First, they can represent general information about the world, such as that students are overworked: *IF x is a student, THEN x is overworked.* Second, they can represent information about how to do things in the world: *IF you register early, THEN you will get the courses you want.* Third, rules can represent linguistic regularities such as *IF an English sentence has a plural subject, THEN it has a plural verb.* Fourth, rules of inference such as modus ponens can be recast in rule form: *IF you have an if-then rule and the if part is true, THEN the then part will be true too.* As this example shows, rules can have multiple conditions (multiple clauses in the IF part). They can also have multiple actions: *IF you register early, THEN you get the classes you want, and you have a short line to stand in.*

It may seem surprising at first that rule-based systems have been so important in cognitive science, since rules are not as representationally elegant as formal logic. Logic provides a standardized way of representing relations and basic operations such as "and," "or," and "not," whereas these can be implemented in various nonstandard ways in rule-based systems. But the developers of rule-based systems have been happy to lose some of the representational rigor of logic-based systems for the sake of increased computational power. One advantage comes from the fact that rules do not have to be interpreted as universally true. The logical generalization (for all *x*) (student (*x*) → overworked (*x*)) must be interpreted as saying that every student is overworked. But the rule that *IF x is a student THEN x is overworked* can be interpreted as a *default*, that is, as a rough generalization that can admit exceptions. We might have another rule that says that *IF x is a student and x is taking only easy courses, THEN x is not overworked.* These two rules might coexist in the same system, but the result need not be the contradictory conclusion that a particular student is both overworked and not overworked, since the computational operations of

the rule-based system can ensure that only the more appropriate rule is applied.

Unlike logic, rule-based systems can also easily represent strategic information about what to do. Rules often contain actions that represent goals, such as *IF you want to go home for the weekend, and you have bus fare, THEN you can catch a bus.* Such information about goals serves to focus the rule-based problem solver on the task at hand. Hence, although the rules in a rule-based system may not have the full representational power of formal logic, they can be expressed in ways that enhance computational power and psychological plausibility.

Computational Power

Problem Solving

In logic-based systems the fundamental operation of thinking is logical deduction, but from the perspective of rule-based systems the fundamental operation of thinking is *search.* When you have a problem to solve—for example, how to write an essay for a course—you have a *space* of possibilities that you must navigate. This space includes the possible topics you might write on, the range of available library resources you might consult, and the means you might employ to actually write the essay. Accomplishing your assigned task requires you to search through the space of possible actions to find a path that will get you from your current state (essay to be done) to the desired state (finished essay that will earn a good grade). Rule-based systems can efficiently perform this kind of search for a solution.

In complex problems, it is impossible to search the space exhaustively for the very best solution. Suppose, for example, you wear four different articles of clothing (shirts, socks, etc.) and you have ten pieces of each article (ten shirts, etc.). Then there are ten thousand (10^4) different combinations of clothes that you might wear each day, but no one has the time or interest to consider all these possibilities. Instead, people rely on *heuristics*, which are rules of thumb that contribute to satisfactory solutions without considering all possibilities. A heuristic such as "Wear brown shoes with brown pants but not with black pants" helps to provide an efficient solution to the problem of planning what to wear. Problem solving, learning, and language use can all be described in terms of rule-based heuristic search through a complex space of possibilities.

Psychologists make an important distinction between long-term memory, the mind's permanent store of information, and short-term memory, a much smaller selection of information immediately available for processing. From the rule-based perspective, you have many rules in long-term memory, but only a small selection of rules and facts are active in your short-term memory and ready for current use. You probably have your mother's birthday in long-term memory, but reading this sentence may make you conscious of it as it becomes active in short-term memory.

Computer scientists and psychologists make an important distinction between serial processing, in which thinking proceeds one step at a time, and parallel processing, in which many steps occur at once. Rule-based processing can be either serial, with one rule being applied at a time, or parallel, with many rules being applied simultaneously. Conscious thought tends to be serial, as we notice ourselves making one inference at a time, but these inferences may depend on numerous rules of which we are not conscious being applied simultaneously. Chapter 11 discusses the role of consciousness in thinking.

Planning Many students go to college or university in a town or city away from their home town or city, so they frequently face the problem of how to get home for the weekend or at the end of term. The available means for getting from university to home can be expressed in collections of rules, such as

IF you drive on highway 1, THEN you can get from university city to home city.

IF you take the parkway, THEN you can get from university city to the highway.

IF you take Main Street from the university, THEN you can get from the university to the parkway.

IF you take a bus from the bus depot, THEN you can get from university city to home city.

IF you take a bus from the university to the bus depot, THEN you can get to the bus depot.

Other possibilities may also exist, such as taking the train or hitchhiking. Students who must solve the problem of how to get home for the weekend can search the space of possible actions (go to the bus depot, head for the highway) and put together a plan that gets them where they want to be.

Rules can be used to reason either *forward* or *backward*. Reasoning backward, a student might think that "To get home, I can take the highway, which requires taking the parkway, which requires taking Main Street, which requires getting a car." The goal is to get home, but the plan is constructed by considering a series of subgoals such as getting to the highway. Reasoning forward, the student might use inference akin to modus ponens to see that "Main Street gets me to the parkway, which gets me to the highway." Forward and backward reasoning both try to find a series of rules that can be used to get from the starting point to the goal, but they differ in the search strategy employed.

Another possible reasoning strategy is *bidirectional* search, which combines working forward from the starting place with working backward from the goal. Although many planning problems can be understood in terms of rule-based reasoning, planning in this way is difficult when there are many potentially relevant rules and the reasoner has to select which ones to use at the key points in problem solving. Rule-based problem solving sounds a lot like logical deduction, but it differs in that much more attention is paid to strategies for applying the right rules at the right time.

The same is true for the sort of planning problems that students encounter in courses. A mathematical word problem gives you some information and requires you to calculate an answer. For example, you may be told that it takes 75 minutes to get from the university city to the home city, a distance of 65 miles (100 kilometers), and be asked to calculate the average speed of the trip. Rules, embodied in mathematical operations, show you how to move forward from the information given to an answer that can be derived from it. Often, however, it will be more effective instead or in addition to work backward from the goal—the desired answer—toward the initial information given. In either case, you are trying to find a sequence of rules that provides a path between the start and the goal. Not all planning, however, is rule based. We will see in later chapters how schemas and analogies can help to solve planning problems.

Decision Although rules are very useful for finding plans, they are not always very helpful for deciding between competing plans. A student may be able to use rules to construct two different ways of getting home for the weekend, but the result does not provide guidance about which plan to adopt. Driving, taking the bus, and taking the train will all get you home, but which way you go will require a more complex balancing of goals such as wanting to minimize cost, time, and hassle. For decision making, therefore, rule-based reasoning needs to be supplemented by other processes, such as the expected-value calculation mentioned in chapter 2, or the deliberative coherence determination described in chapter 7.

Explanation As we saw in chapter 2, explanation can often be viewed as a kind of deductive process, which rules can perform as well as logical deduction. Some kinds of hypothesis formation can be described as a search for explanations performed by rules. Suppose you try to register for a course and it turns out to be full. Various rules might apply:

IF a course is required for many programs, THEN it fills up quickly.

IF a course has a popular instructor, THEN it fills up quickly.

Knowing that the course's instructor is popular, in conjunction with the second rule just stated, allows you to explain why it was full when you tried to sign up for it. Even if you do not know for sure that the course has a popular instructor or is required for many programs, you can conjecture that these might be true (see the discussion of abductive learning below). Thus, solving explanation problems can be understood in terms of rule-based reasoning, if there is a sequence of rules that allows you to generate what needs to be explained from what you already know.

Learning

Numerous important kinds of learning are naturally understood in terms of the acquisition, modification, and application of rules. Some rules may be innate, comprising part of the biological equipment with which we are born. A physical rule such as *IF something is coming toward your eyes, THEN blink* is not one that people or other organisms have to learn. More controversially, some cognitive scientists discussed in the next section believe that many rules of language are innate. But no one would claim that rules for how to register for university courses are innate, so how are they acquired?

Like the logical statements described in chapter 2, rules can be learned by *inductive generalization,* in which examples are summarized by means of a rule. Sometimes rules require many examples to support them: you should not conclude from just one engineering class that all engineering classes are hard or from just one philosophy course that all philosophy courses are interesting. But students do gradually acquire from experience such rules as *IF x is a programming class, THEN x will be time consuming* and *IF you want to get into popular courses, THEN you should register early.*

In inductive generalization, rules are formed from examples; but rules can also be formed from other rules by a process that in the SOAR model of cognition is called *chunking* and in the ACT model is called *composition.* Suppose that you have used lots of rules to plan how to get from university to home and found a good series of rules about how to get from university to the parkway to the highway home. The next time you want to go home, you need not go through the whole search. Instead, you can chunk the rules into a general rule like *IF you want to get from university to home, THEN drive.* Similarly, the first time you put together a class schedule, you may have to do lots of complex searching for a plan, but with experience you can use a higher-level rule such as *IF you want a good schedule, THEN arrange your classes at close times rather than spread over five days.* We will see in the section on psychological plausibility that the computational process of chunking rules has been used to model many kinds of human learning.

Another way that rules can be formed from rules is by *specialization,* in which an existing rule is modified to deal with a specific situation. If driving home on Friday afternoon may be slow because of heavy traffic, experience might lead you to produce the specialized rule *IF you want to get from university to home and it is Friday afternoon and you are in a hurry, THEN don't drive.*

As we saw in chapter 2, rules can also be used in abductive learning. Suppose a friend of yours is angry and depressed. Naturally, you try to construct explanations of what is bothering him or her. Suppose you have acquired by inductive generalization the rule *IF a student gets a bad grade, THEN the student is angry and depressed.* You could then generate a possible explanation of why your friend is angry and depressed by conjecturing that he or she might have received a bad grade. This is abductive reasoning, in which a rule is run backward to provide a possible explanation of what happened. Obviously, this kind of inference is highly risky, since there might be a much better explanation of your friend's state of mind—

for example, one based on the rule *IF someone is rejected by a partner THEN he or she becomes angry and depressed.* Picking the best explanation requires a more complex kind of inference discussed in chapter 7. But rules can be very useful for generating hypotheses such as that your friend got a bad grade. Hence abductive inference fits naturally with rule-based reasoning (Thagard 1988), although we will see other kinds of representation that also support it.

Rules can also be used to describe slow incremental learning, if each rule has a numerical value representing the usefulness or plausibility associated with it. The more a rule gets used successfully, the more it is judged to be plausible and useful. For example, each time a student successfully applies the rule, *IF you want to get from university to home, THEN drive,* the stronger the rule gets and the more likely it is to be used in the future. In sum, rules can be created from examples, created from other rules, applied abductively, and quantitatively evaluated based on their performance.

Language

Before the cognitive revolution of the 1950s, language was widely thought to consist of behavior learned by association. Through repeated experience with pairs of words, people come to expect to hear them used together. The linguist Noam Chomsky developed a very different view of language beginning with his 1957 book *Syntactic Structures*. Chomsky argued that the behaviorist learning models could not account for the generativity of language, the fact that there are an indefinite number of sentences that people can produce and understand. You have probably never encountered the sentence "She rode to the university on a purple camel," but you have no trouble understanding it.

According to Chomsky, our ability to speak and understand language depends on our possessing a complex grammar that consists of rules that we do not consciously know we have. Children who learn English, for example, start forming the past tenses of verbs by adding "ed," without being aware that they are applying a rule like *IF you want to use a verb to describe the past, THEN add "ed" to the verb.* Notoriously, children under five overgeneralize the rule, saying "goed" and "bringed" rather than using the irregular forms that are exceptions to the rule. Pinker (1999) argues at length that rules such as adding "ed" to make past tenses are an essential ingredient of our ability to produce and comprehend language. Taatgen and Anderson (2002) use the ACT system to model how children acquire past tenses in English. Akmajian et al. (2001) describe rules that apply to several different aspects of language. For example, as English speakers we know how to form nouns from verbs by adding "er," as in turning "write" into "writer." We also know phonetic rules such as how to pronounce plurals: compare the predictably different pronunciations of the "s" in "cats"/"huts" and "dogs"/"hugs." Syntactic rules enable us systematically to turn statements into questions, as when we turn "I am happy" into "Am I happy?" by moving the auxiliary verb "am" to the beginning of the sentence.

Chomsky's influential views have been controversial on a number of issues. In chapter 7, we will consider connectionist views that language consists not of rules but of looser associations represented by weights between simple units. Independent of the issue of whether our knowledge of language is best represented by rules, there is the issue of whether it is learned or innate. Chomsky continues to maintain that every human is born with an innate universal grammar. In a departure from his early views, in which children acquire the ability to use language abductively by forming hypotheses about what rules apply to their own individual languages (Chomsky 1972), he currently holds that children learn a language automatically by merely recognizing which of a finite set of possibilities that language employs (Chomsky 1988). All human languages have nouns, verbs, adjectives, and prepositions or postpositions. But languages such as Japanese do not have articles like "the" and "a" in English, so a child learning Japanese has to instantiate universal grammar in a different way than child learning English. Most recently, Chomsky (2002) has raised doubts about whether grammar is a system of rules.

Psychological Plausibility

Of all the computational-representational approaches described in this book, which has had the most psychological applications? The answer is clear: rule-based systems. I cannot attempt to give a comprehensive review of all of these applications, but I will provide a sample of some typical ways in which rule-based systems have been used to account for human thought.

Newell (1990) has shown how SOAR, a sophisticated current rule-based model, can be applied to a wide range of interesting psychological phenomena. For example, he describes how SOAR solves cryptarithmetic problems, which are puzzles in which letters are substituted for numbers (see also Newell and Simon 1972). One puzzle is *DONALD* + *GERALD* = *ROBERT*, where each letter must be replaced by a distinct number between 0 and 9 in a way that makes the equation true. Rules can be very useful for solving this problem, which for our usual addition algorithm is more perspicuously represented as

DONALD
GERALD

———

ROBERT

How does one begin to solve this puzzle? You might notice that in the second column from the left *O* added to *E* produces *O*. This might bring to mind the rule: *IF 0 (zero) is added to a number THEN the number is unchanged*. This rule suggests that *E* is 0. Then, looking at the fourth column, you could see that since *A* + *A* = 0, *A* should be 5. But following this line of reasoning is likely to get you into trouble, because this assumes that no carry was involved in adding *L* and *L* to get *R* or in adding *N* and *R* to get *B*. So somehow *R* needs to be twice as big as *L*, yet small enough that adding *N* to it in the third column does not produce a carry. Carrying in our familiar addition algorithm involves the rule *IF the digits added exceed 10, THEN write down the second digit of the sum and carry 1 over to the next column to the left*. This rule shows that there is another possible value for *E*: if *E* = 9, then *O* + *E* = *O*, provided that there is a carry from the column *N* + *R* = *B*. Working out the consequences of this starting point using rules about addition and carrying, along with additional knowledge such as that *IF a digit is at the beginning of a number, THEN the digit is not 0*, allows you (with considerable effort) to come up with a solution. SOAR is able to model aspects of this effort by having various operators that suggest numerical values for the digits and checking to see whether the results are consistent with each other.

SOAR has also been used to model other kinds of high-level reasoning tasks such as determining what follows from "Some archers are not bowlers" and "All canoeists are bowlers." SOAR uses neither mental logic nor mental models, the two approaches to deduction described in chapter 2, but instead does a search through a space of possible inferences, eventually forming the conclusion that "Some archers are canoeists." This is logically incorrect, but the point of a cognitive account of deduction is to model how people reason, including how they sometimes make errors.

Newell also uses SOAR to account for many aspects of human learning, particularly the power law of practice, according to which the rate of learning slows down as more is learned. This law applies to many tasks such as typing and learning to write reversed letters the way they appear in a mirror. Chunking in SOAR provides an explanation of why learning slows down as people become more experienced with a task. At the beginning of practice on a task, people can build more chunks rapidly, and as chunks build up, the speed of performance increases. For example, someone learning to type may make rapid early progress in speed and accuracy. But as higher-level chunks build up, they become less and less useful, because the situations they apply to are rare. So the learning rate of a rule-based system slows down with practice, just like that of people.

Holland et al. (1986) used rule-based systems to account for many different kinds of learning. Conditioning in rats—for example, when they learn to avoid shocks—can be explained by supposing that part of learning with rules is adjusting the strengths of different rules that are used. Every time a rat presses a lever to get food, the rule *IF lever, THEN food* gets strengthened. On the other hand, a shock can produce the conflicting rule *IF lever, THEN shock* and lead to the rat's ceasing to press the lever. Rules can also be used to describe the dynamic mental models that people have of changes in the physical world, such as *IF a car hits a pole, THEN the pole is damaged*. People's abilities and limitations in dealing with the physical and social worlds can also be understood in terms of rules. For example, once people learn a social stereotype, they tend to apply it too generally.

Crowley and Siegler (1993) have shown how variations in children's ability to play tic-tac-toe, a simple game in which the goal is to get 3 Xs or 3 Os in a row, can be understood in terms of their acquisition of rules. Children need to acquire rules about what moves to make, as well as strategic knowledge about what rules to apply when. Here are some of the rules:

Win IF there is a row, column, or diagonal with two of my pieces and a blank space, THEN play the blank space to win.

Block IF there is a row, column, or diagonal with two of my opponent's pieces and a blank space, THEN play the blank space to block the opponent.

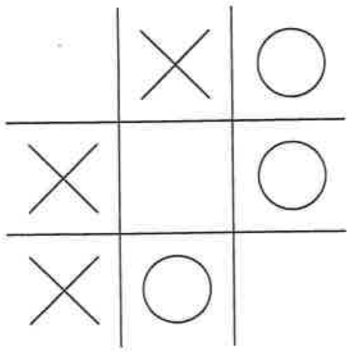


Figure 3.1
Rule application in tac-tac-toe. X's turn matches the IF part of four different rules: Win (a move to the top left); Block (a move to the bottom right); Play Center (a move to the middle); and Play empty corner (a move to either empty corner. Adapted with permission from Crowley and Siegler 1993, p. 537.

Play center IF the center is blank, THEN play the center.

Play empty corner IF there is an empty corner, THEN move to an empty corner.

Figure 3.1 shows a partially played game in which all four of these rules are applicable. Children's ability to play tic-tac-toe improves as they acquire rules such as these as well as recognition of the priority of rules. For example, many preschoolers do not have the blocking rule, and many who do have it will apply it even when they have a chance to win.

Rule-based systems have also been used to account for the acquisition and use of language. Anderson (1983) describes human knowledge of English in terms of such rules as this one (in simplified form):

IF the goal is to communicate a meaning structure of the form (relation, agent, object), THEN set as subgoals

1. to describe agent
2. to describe relation
3. to describe object.

Additional rules show how the mentioned subgoals can be accomplished so that eventually a full sentence, such as "The girl threw the ball," is produced to describe what the agent did to the object. Anderson (1993) describes numerous applications of his ACT rule-based system to acquiring skills such as geometry problem solving and computer programming. There are many examples of a fit between the performance of rule-based systems and the behavior of human thinkers.

Neurological Plausibility

There is a crude analogy between rules and neurons connected by synapses, in that IF one neuron fires, it can THEN cause the firing of the neuron connected to it. But this similarity is superficial and in fact little is known about how rules might be implemented in the brain. Anderson (1993) sketches a possible neural implementation of ACT, and simple rule-based systems have been implemented in artificial neural nets (see chapter 7). More recently, Anderson et al. (forthcoming) have related the newest version of the ACT system, ACT-R, to specific brain regions. Based on brain scans of people solving problems, Anderson et al. infer that production rules are implemented by the brain's basal ganglia, which are a collection of nuclei deep in the white matter of the cerebral cortex. They also estimate that the facts that the rules matched are stored in a set of buffers in the prefrontal cortex. Thus the ACT system, which originated as a purely cognitive model, is becoming a neurological model as well. For more on brains, brain scans, and neurological models, see chapter 9.

Practical Applicability

If what we learn consists of rules, then education must be concerned with helping children and other students better acquire those rules. Anderson (1993) discusses numerous educational applications of ACT rule-based systems, including understanding how people learn computer programming, text editing, and doing proofs in geometry. Rule-based systems have been used not only to model learners' performance, but also to build computer tutors that can help them learn.

Design in engineering and other fields can also be understood in terms of rules. Newell (1990) describes a version of SOAR that designs computer algorithms by using operators that generate and test program specifications to search a space of possible algorithms. He and his collaborators have also discussed the implications of viewing human computer users as rule-based systems for designing computers that people can easily use (Card, Moran, and Newell 1983). SOAR is now being incorporated into computer games such as Quake, in order to produce opponents that behave like humans (Laird 2001). Characters in computer games usually have very limited flexibility, but SOAR can give them some of the complex decision making that make human opponents challenging.

Most expert systems used in industry and government are rule-based systems, which were the first kind of applied intelligent system to be developed (Buchanan and Shortliffe 1984; Feigenbaum, McCorduck, and Nii 1988). Expertise in many domains, from configuring computers to prospecting for oil, can be captured in terms of rules. Recent examples of rule-based expert systems (and other kinds as well) can be found in the *Proceedings of the Nineteenth* (and previous) *Innovative Applications of Artificial Intelligence Conference*, published by AAAI Press. Langley and Simon (1995) provide numerous examples of industrial application of computer programs that learn rules from examples, including systems for chemical process control, making credit decisions, and diagnosis of mechanical devices.

Summary

Much of human knowledge is naturally described in terms of rules, and many kinds of thinking such as planning can be modeled by rule-based systems. The explanation schema used is as follows:

Explanation target

Why do people have a particular kind of intelligent behavior?

Explanatory pattern

People have mental rules.

People have procedures for using these rules to search a space of possible solutions, and procedures for generating new rules.

Procedures for using and forming rules produce the behavior.

Computational models based on rules have provided detailed simulations of a wide range of psychological experiments, from cryptarithmetic problem solving to skill acquisition to language use. Rule-based systems have also been of practical importance in suggesting how to improve learning and how to develop intelligent machine systems.

Discussion Questions

1. What areas of knowledge do you have that are easily described in terms of rules?

2. What areas of knowledge do you have that are difficult to describe in terms of rules?

3. How does the rule-based approach differ from the logic approach described in chapter 2?

4. How might the brain implement rules?

5. Is knowledge of language innate or learned?

Further Reading

Classic sources on the mind as a rule-based system include Newell and Simon 1972, Newell 1990, and Anderson 1983, 1993. Holland et al. 1986 discusses many kinds of learning in terms of rule-based systems. Smith, Langston, and Nisbett 1992 makes the case for rules in reasoning; see also Nisbett 1993. Pinker 1994 is an entertaining defense of the Chomskyan approach to language, including the importance of rules to language. Pinker 2002 emphasizes the role of innateness in human behavior in general, but see Elman et al. 1996 and Quartz and Sejnowski 2002 for skeptical discussions of innateness claims.

Web Sites

ACT home page at Carnegie Mellon University: http://act-r.psy.cmu.edu/

John Anderson's home page: http://act-r.psy.cmu.edu/people/ja/

Stephen Pinker's home page: http://pinker.wjh.harvard.edu/

SOAR home page at the University of Michigan: http://sitemaker.umich.edu/soar

Notes

Non-rule-based approaches can also be described in terms of search, but historically that description has been most closely associated with rule-based systems. The search metaphor works well with well-defined problems where the states and operators can be specified, but is much less clear for problems where the task involves learning new representations and operators.

More technically, the power law of practice can be expressed as "If the logarithm of the reaction time in a task is plotted against the logarithm of the number of practice trials, the result is a straight line that slopes downward."

4 Concepts

When students learn their way around their colleges or universities, they acquire new rules about them, but they also acquire new concepts. Many new administrative concepts must be acquired, such as *major*, *register*, and *transcript*. Students also quickly learn new concepts for describing courses, such as *bird* or *gut* or *cake* for an unusually easy course. Social knowledge increases dramatically too, as students learn concepts for describing their fellow students, such as *computer geek*, *jock*, *artsie*, (Arts student), and *keener* (eager student). Students who encounter different kinds of classes like seminars and huge lectures must modify the concept of *class* they acquired in high school.

Concern with the role of concepts in knowledge goes back more than two thousand years to the Greek philosopher Plato. He asked questions such as "What is justice?" and "What is knowledge?" and showed that concepts such as *justice* and *knowledge* are very hard to define. Plato believed that knowledge of such concepts is innate and that education can serve to remind us of the essence of these concepts. Just as Chomsky argues that linguistic rules are innate, Plato and later philosophers such as Leibniz and Descartes contended that the most important concepts are purely in the mind.

Other philosophers such as Locke and Hume contended that concepts are learned through sensory experience. The way you acquire a concept such as *dog*, for instance, is not just by thinking about what dogs are, but by encountering a variety of examples of dogs. Although Jerry Fodor (1975), a contemporary philosopher heavily influenced by Chomsky, maintains that concepts are largely innate, most cognitive scientists today are interested in processes by which concepts are learned from experience and from other concepts.

Psychological and computational interest in the nature of the concepts boomed in the mid-1970s when researchers introduced terms such as "frame," "schema," and "script" to describe new views of the nature of concepts. (Somewhat similar ideas had been advanced by Bartlett (1932) and Kant (1965).) In the most influential artificial intelligence paper of the decade, Minsky (1975) argued that thinking should be understood as frame application rather than logical deduction. In a computational-psychological collaboration, Schank and Abelson (1977) showed how a great deal of our social knowledge consists of scripts, which describe typical sequential occurrences such as going to a restaurant. Around the same time, psychologists such as David Rumelhart (1980) were describing knowledge in terms of conceptlike structures called schemas that represent, not the essence of a concept such as *dog*, but what is typical of dogs. Similarly, the philosopher Hilary Putnam (1975) argued that the meaning of concepts should be thought of in terms of stereotypes, not in terms of defining conditions. During the 1980s, discussion of concepts from a computational perspective took on a different complexion through the development of connectionist models that learn concepts; we will take up this topic in chapter 7.

Representational Power

How often have you heard someone insist, "Define your terms!"? People frequently say things like, "We can't talk about intelligence until we can define what the word 'intelligence' means." The demand requires a definition that would provide the exact rules *IF x is intelligent, THEN x has the properties y* and *IF x has the properties y, THEN x is intelligent*. But as Plato discovered with concepts such as *justice*, such definitions are very hard to come by. As an exercise, try coming up with rules that will exactly capture what is meant by *college*, *university*, *course*, or *geek*. At best, a concept such as *intelligence* will be defined at the end of an inquiry, not at the beginning, and outside of mathematics we should not expect definitions to be available at all.

Construed as frames, schemas, or scripts, concepts are understood as representations of typical entities or situations, not as strict definitions. For example, students acquire a concept of *course* in which instruction takes place by a professor and students get a grade at the end. What is expected of a course can be summarized as a set of slots, each of which can be filled in with expected information such as the instructor's name:

Course

A kind of: process (systematic series of actions)

Kinds of courses: lecture course, seminar, etc.

Instructor:

Room:

Meeting time:

Requirements: exams, essays, etc.

Instances: Philosophy 100, Mathematics 242, etc.

One of the first things a student does in signing up for a course is to find out who the instructor is, thus filling in an important slot. The *course* concept could perhaps be represented differently by a set of rules such as *IF x is a course, THEN x has an instructor*, but we will see in the next section that there are computational reasons why it is useful to think of a concept in terms of a set of slots. Although typically courses have an instructor, this should not be taken as part of the definition of a course, since there are team-taught courses with more than one instructor and correspondence courses that have no instructor.

Some concepts involve a temporal sequence, as in taking an exam:

Exam

A kind of: course requirement

Kinds of exam: written, oral, take-home, short-answer, etc.

Room:

Sequence:

Get the exam questions

Put your name on the exam paper

Answer the questions

Check your answers

Hand in your exam

Again, this concept can be thought of in terms of rules such as *IF you take an exam, THEN first get the exam questions*. But it is cognitively useful to have acquired a package of information that can be applied as a whole.

Although the slots in concepts can usually be translated into rules, it is important to realize that slots do not express universal truths, only what is expected to hold typically. The values of slots are sometimes called *defaults*, as when the default value for number of instructors in a class is one. Rules can also be understood as expressing default expectations rather than universal truths, as in the statement *IF x is a course, THEN x typically has an instructor*. Exams typically take place in a room on campus, but take-home exams are an exception in that they do not have any special room in which everyone takes the exam.

Concepts organize knowledge in important ways that are not usually found in rule-based systems. Notice that the concept *course* includes slots that state what kind of thing a course is and what kinds of things are courses. These kind relations establish a hierarchical network of concepts, as in figure 4.1. A seminar is a kind of course which is a kind of educational process, and so on. Conceptual organization of this sort has computational consequences that give concepts properties that unorganized sets of rules might lack. Another sort of slot that is important for many physical concepts is *part*, which establishes another sort of hierarchy. For example, a toe is a part of a foot, which is part of a leg, which is part of a body. Slots involving parts can also be translated into rules, such as *IF x is a toe, THEN x is part of a foot*. But organizing concepts into slots and hierarchies has computational advantages discussed in the next section.

Concepts are clearly not intended to be a complete theory of mental representation. The information that if a course is full you can still get into it by getting a signature from the professor is not part of the concept of *course*; rather, it is a rule you learn about courses. But concepts have computational properties that make them useful additions to rules for modeling human thinking.

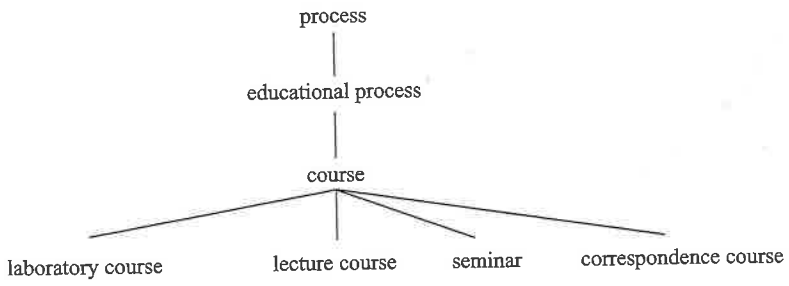


Figure 4.1
Hierarchical organization of the concept of a course.

Computational Power

Packaging information into concepts that are hierarchically organized makes possible powerful kinds of computations. Large rule-based systems face the problem of selecting rules to apply. It does a system no good to have a rule that is relevant unless that rule can be retrieved from memory in order to be applied. One process that can be applied very efficiently in a concept-based system is *inheritance*, in which inferences about concepts can be quickly made using the hierarchy established by the *kind* slot. Does a seminar have an instructor? The answer to this question might not be directly represented as part of the concept of *seminar*, but it can quickly be gained by noting that a seminar is a kind of course, and courses typically have instructors. This is not a logical deduction, since not all courses have instructors. But it is reasonable to expect that a seminar has an instructor, where this expectation is inherited by virtue of a seminar's being a kind of course.

What do you think of when you hear the word "desk"? Perhaps you begin to think about chairs, studying, or lamps. Not all thinking is a matter of making inferences in the way that logic-based and rule-based systems do. One might associate desks with chairs by means of some rule such as that every desk has a chair, but this association could also come about more casually by virtue of the fact that a desk is a kind of furniture, and another kind of furniture is a chair. This kind of loose association is described computationally as a process of *spreading activation*. One concept in a system is active, and activation spreads in a network to other concepts that are linked to it by kind and other relations. Spreading activation is like a type of electronic contagion, in which one electrified object electrifies other objects connected to it. For example, if something activates your concept of *desk*, activation may spread to your concepts of *furniture* (desks are a kind of furniture) and *drawer* (a drawer is a part of a desk). Activation of these concepts may then lead to activation of related concepts such as *table* (a table is a kind of furniture) and *wood* (drawers are typically made of wood). Some rule-based systems include spreading activation as a mechanism along with rule firing for modeling how people retrieve rules from memory (Anderson 1983; Thagard 1988).

Packaging information in a concept is most useful when it can be used to deal with new situations. When a course begins, students quickly fit it into their conceptual system, categorizing it as a lecture course, a bird (easy) course, or whatever. There are two crucial steps in the process, *matching* and *inference*. Finding the most appropriate concepts to apply to a course requires matching the slots of various relevant courses against the particular information known about the course. For example, if there are only ten students enrolled in the course, this information will fit with the slot in the concept of *seminar* that suggests that class size is typically small. If instead there are a hundred students in the class, it will not fit in the size slot of the *seminar* concept, but will better match the *lecture* concept. Once a concept is matched to a situation, students can make inferences about the situation by carrying over the full set of expectations produced by the concept. Once a course is classified as a seminar, the student will probably expect it to have lots of class discussion. Thus, to understand the computational role of concepts, we need to think of these steps in a processing system:

1. The system has active concepts that represent a situation.
2. These concepts spread activation to other potentially relevant concepts.
3. Some concepts that match the current situation well are selected.
4. The system makes inferences about the situation by inheritance from the selected concepts.

Problem Solving

Planning The first time you encounter a planning situation such as registering for courses, you may need to use general rules to search for a solution. But successful registrations will make it easier for you next time around, since you can then simply follow the same sequence of operations. You have acquired a script or concept for registration. Planning then is not search or logical deduction, but concept application. Given a representation of your current situation and the goals you want to accomplish such as getting into the courses you want, you retrieve from memory a concept of registration that matches the situation and goals. This script can then be applied to tell you what to do in the appropriate order: sign up for courses, pay tuition, and so on.

Concept application, however, works only if you have an organized package of information that closely matches your current situation. After a year or more of college or university, students have a set of schemas that are useful for many educational situations, but first-year students may err by trying to apply schemas acquired in high school. Students taking the course Introduction to Cognitive Science are sometimes confused because they expect its content to be like that of courses they are already familiar with in philosophy, psychology, or computer science. They have difficulty fitting an interdisciplinary course into their previous concepts. Scripts can be very useful when they can be applied to situations that occur frequently, but they can hinder planning in novel situations where existing situations do not fit. Recall the saying: To someone whose only tool is a hammer, everything looks like a nail. The lesson is that you should not try to apply the concept of hammering to situations in which it is not relevant.

Decision The same lesson applies to decision making using concepts. In some cases, making decisions based on a familiar script will not cause you problems, as when you always order the same flavor in an ice cream store. But people often apply schemas unreflectively. Hiring decisions, for example, are sometimes made not on the basis of a reasoned judgment of which candidate will best meet the needs of the organization hiring, but rather because a particular candidate fits the boss's concept of the right kind of employee. That concept might have appropriate slots that describe the intelligence and industriousness of the ideal candidate, but it also may include extraneous requirements such as race and gender. Thus, although some decisions are undoubtedly made by concept application, it is a good thing that not all are. Concept application is a quick and easy way to make a decision, but it does not always take into account the complex of concerns about actions and goals that are part of more reflective decision making.

Explanation Like plans, explanations sometimes come in schematic packages. Social concepts are often used in explanations, probably more than they should be. Why did Fred stay up all night programming? Because he's a computer geek. Why does Sarah always wear black? Because she's an artsie. Why did Alice get an A in that course even though she did not study? Because it's a bird (gut, cake). In all these cases, explanation comes almost automatically by matching a concept to a situation that it seems to fit.

But concepts also have more reflective explanatory uses. Scientific explanation sometimes has the deductive flavor that logic-based and rule-based systems give to explanation. In physics, for example, there are often general laws such as *force = mass times acceleration* that can be applied mathematically to produce a deductive explanation of planetary motion. But in many fields, such as evolutionary biology and the social sciences, laws are hard to come by. Then explanation is better characterized as application of a schema that includes a target—what is to be explained—and a kind of pattern that furnishes the explanation. Here is a simplified explanatory schema for using Darwin's theory of evolution by natural selection to explain why a species has a particular trait (from Thagard 1999; see also Kitcher 1993 and Schank 1986):

Explanation target

Why does a given **species** have a particular **trait?**

Explanatory pattern

The **species** has a set of variable **traits.**

The **species** experiences environmental **pressures.**

The **pressures** favor members of the **species** that have a particular **trait.**

So members of the **species** with that **trait** will survive and reproduce better than members of the **species** that lack the **trait.**

So eventually most members of the **species** will have the **trait.**

The terms presented in boldface are variables that can be filled in by many different examples. If you want to explain, for example, why some bacteria are resistant to antibiotics, this pattern can be applied by noticing that the trait of resistance to antibiotics is variable in a species of bacteria, that antibiotics introduce environmental pressures, so that bacteria with resistance to antibiotics will survive and reproduce better until the species is resistant.

We have already seen various instances of the explanatory schema that is fundamental to cognitive science. The summary for chapter 1 provided a general explanatory schema based on representations and computational procedures, and the summaries for chapters 2–7 include explanatory schemas for particular representational approaches. Part II discusses aspects of mind and intelligence to which such schemas are harder to apply.

Learning

We saw three kinds of answers to the question of where rules come from: they can be innate, formed from experience, or formed from other rules. The same three kinds of answer apply to concepts, which can be innate, formed from examples, or formed from other concepts. Different answers are appropriate for different concepts. Young children acquire new words and the corresponding new concepts at the rate of around ten per day.

Consider, for example, the concept of a *human face* consisting of two eyes, a nose, and a mouth. Perhaps babies learn this concept from experience as they repeatedly encounter examples of faces. But there is experimental evidence that babies do not have to learn the typical structure of faces, but rather are born expecting faces to look a certain way. Similarly, there is growing evidence that basic physical concepts such as *object* are innate, since very young infants show strong expectations about how objects should behave—for example, when they disappear behind another object and then reappear. Thus, whereas it is implausible to suppose that *all* our concepts right up to *DVD player* and *cell phone* are innate, some basic concepts as well as the mechanisms for forming new ones seem to be part of our inborn mental equipment.

Some concepts are learned from examples in much the way that some rules are formed by inductive generalization. Some concepts must be gained laboriously from many examples, as when a child learns to discriminate dogs from other animals. When you know a lot, however, you can acquire concepts quickly from a small number of examples. If you walk into a course and are surprised to discover that it has just ten students and that instead of lectures there is much discussion, you can acquire the concept of *seminar* from that example alone. Of course, the concept may be revised on the basis of subsequent examples. Just as rules are fine tuned for content and plausibility by repeated use, so concepts can be modified as additional examples are encountered. Many sophisticated computational models of concept formation from examples have been developed (Langley 1996). Connectionist methods of concept formation are discussed in chapter 7.

Not all concepts need to be formed from examples, since we can produce new concepts by combining ones we already have. Examples may play a role in filling in details of concepts such as *music television* and *electronic mail*, but much of the content is furnished by the concepts that are combined to produce the new one. Some conceptual combination is straightforward, as when we can figure out that a *pet fish* is just a fish that is kept as a pet. But other conceptual combinations are more complex; for example, a *computer geek* is not something that is both a computer and a geek, but rather a strange person obsessed with computers. Some surprising conceptual combinations can even involve an abductive component when hypotheses are required to explain how the combination might be possible. For example, the concept of *blind lawyer* is formed not simply by combining the attributes of *blind* and *lawyer*, but by adding emergent attributes such as *courageous* that are needed to explain how a blind person can become a lawyer (Kunda, Miller, and Claire 1990). There are computer models of simple kinds of conceptual combination (Thagard 1988), but not yet of the more complicated abductive kind. Costello and Keane (2000) present a computational theory that explains both the creativity and the efficiency of people's conceptual combinations.

Schemas that include causal information can be used to perform a kind of abductive inference. Here is a script for acquiring a contagious disease such as a cold:

Contagious disease

Contact: You come into contact with some germs (viruses or bacteria).

Incubation: The germs multiply.

Symptoms: The germs cause you to develop symptoms such as a runny nose.

Cure: Eventually, your body's immune system kills the germs.

If you have symptoms such as a runny nose, you can fill in the symptom slot in this schema, and then fill in the contact slot to abductively infer that you must have come in contact with some germs.

Language

In spoken and written language, concepts are represented by words. Not all concepts need have words that describe them, but there is a close correspondence between our words and many of our concepts. In the last section, we discussed grammar in terms of linguistic rules. Knowledge of language, however, obviously cannot consist of such rules alone: we need to know words to plug into grammatical structures. A set of words in a dictionary is called a *lexicon*, so that the set of words or concepts represented in a mind is called the *mental lexicon*.

George Miller and others have argued that the mental lexicon is organized hierarchically (Fellbaum 1998). He and coworkers have produced a huge electronic lexicon called WordNet, with more than 60,000 English words. Nouns such as "dog" are organized hierarchically in terms of kinds and parts as was described in the above section on representational power (figure 4.1). Verbs that express actions such as "register" and "run" have a different kind of organization in terms of ways of doing things. For example, in one sense running is a way of traveling, and sprinting is a way of running. Adjectives such as "easy" are organized in other ways. Our use of language depends on our ability to store and use concepts for such nouns, verbs, and adjectives. Miller (1991) discusses the structure of the mental lexicon, how words are formed, and how children's vocabularies grow.

Learning a language is not just a matter of acquiring grammatical rules; it also involves developing a whole conceptual system. Linguists in the Chomskyan tradition have assumed a sharp distinction between grammar and lexicon, but the distinction is challenged by advocates of a different approach, called *cognitive grammar* (Taylor 2003). Langacker (1987) and Lakoff (1987) argue that syntactic structure is very closely tied in with the nature and meaning of concepts.

What is the meaning of a concept and how does it contribute to the meaning of a sentence? Philosophers have been particularly vexed by this question and have developed a range of possible answers to it. On the one hand, the meaning of a concept seems to derive from the meaning of other concepts, as when a child is told the meaning of *sprint* by saying it is a kind of fast running. On the other hand, the meaning of a concept is connected to observations of things in the world, as when the child actually sees someone sprinting. A concept's meaning is normally not given by definition in terms of other concepts, since successful exact definitions are rare. Nor is meaning exhausted by a set of examples, as if one identified the concept of *dog* with the set of dogs. A theory of meaning of the concepts must therefore include an account of how concepts are related both to each other and to the world (see chapter 12). Both aspects are necessary in order for us to understand how concepts underlie our ability to use language.

Psychological Plausibility

How does one show the psychological plausibility of a particular kind of mental representation? The direct method is to perform psychological experiments producing results that follow immediately from the assumption that people have the proposed kind of mental representation. The indirect method is to use computer simulations employing the proposed kind of mental representation to explain the results of experiments concerning some general sort of performance. Much of the evidence for the psychological plausibility of rules described in chapter 3 is of the second, indirect sort—for example, the rule-based simulation of cryptarithmetic problem solving. In contrast, much of the evidence for the psychological reality of concepts comes from experiments about concepts rather than from computer simulations. Psychologists have performed a vast number of experiments designed to determine the nature of concepts and their role in categorization. Here I will mention just a small selection of important experiments.

While behaviorism dominated psychology, there was little talk of concepts or any other mental representations. When research on concept learning began in the 1950s, it presupposed the classical view of concepts as sharply defined (Bruner, Goodnow, and Austin 1956). During the 1970s, however, evidence mounted that concepts should be understood in terms of typical conditions rather than defining conditions. Defining conditions are ones that provide strict rules, as when we say that a figure is a triangle if and only if it has exactly three sides. Typical conditions allow for exceptions, as when we say that dogs typically have four legs, even though some have only three. A prototype is a set of typical conditions, so that the prototype for *dog* is something like "has four legs, is furry, barks," and so on. On the classical view, applying the concept *dog* to a particular example such as Benji is a matter of checking whether the defining conditions of *dog* apply to Benji. But on the prototype view, applying *dog* to Benji is a looser process of seeing whether the typical conditions of *dog* match Benji's characteristics.

Psychological experiments suggest that concept application fits the prototype view rather than the classical view. Posner and Keele (1970) used patterns of dots as perceptual categories. Experimental subjects were required to learn sets of four distortions of each of four prototypical patterns of dots and were then given a new set of patterns to classify. Of the new patterns, subjects found ones that matched the prototypes easiest to classify, but took longer and made more errors in classifying patterns more distant from the prototype. Similarly, people can more quickly verify the truth of the sentence "A robin is a bird" than they can verify "A goose is a bird," presumably because robins are closer to the prototype for *bird* than geese are.

Rips, Shoben, and Smith (1973) showed that people reliably rate some category members as more typical than others. For example, in North America a banana is a more typical fruit than a mango. When people are asked to list examples of a concept, they tend to produce items that are considered most typical (Rosch 1973). If you are asked to name a bird, you will be more likely to say "sparrow" than "penguin." Rosch and Mervis (1975) found that judgment of how typical a kind of bird is correlates highly with the extent to which the bird has the properties that are most commonly assigned to birds, such as flying and building nests. A robin and a penguin are both birds, and would both have to fall under the definition of *bird* if one could be produced, but cognitively they differ enormously because a robin is much closer to the prototype for *bird* than is a penguin.

Viewing concepts as prototypes helps to account for various features of how concepts are applied, including mistakes that people make. For example, when Brewer and Treyens (1981) asked subjects to recall what items were in a university office where they had been kept waiting, they often mistakenly reported that there were books in the office. Books are part of the prototype of *academic office*. Psychological experiments have also been performed that tease out some of the aspects of conceptual combination (Smith et al. 1988).

The findings about prototypes fit well with the computational view of concepts as framelike structures that list typical properties. However, there is experimental evidence that the structure of concepts is not fully captured by prototypes. Barsalou (1983) and others have argued that concepts are much more flexible and context dependent than a package of typical properties would be. Some psychologists have argued that our knowledge of a particular concept is closely tied in with the initial examples from which we learned the concept. Applying a concept is then a matter not of matching to a prototype but of comparing new examples to the old ones.

Concept application is then similar to analogical reasoning, discussed in chapter 5. Barsalou et al. (2003) argue that conceptual processing depends on specific modalities such as perception, so that a concept like *car* is tied in with memory of sensory experiences of cars.

Murphy and Medin (1985), Keil (1989), and others have argued that neither sets of typical features nor examples capture all that there is to concepts. Concept application is sometimes as much a matter of causal explanation as it is of matching features—for example, when we classify as drunk someone who jumps into a pool fully clothed. Jumping into a pool fully clothed is not a defining or typical feature of the concept *drunk*, but it fits with a theory of impaired judgment that is part of the concept: being drunk causes people to do silly things. Murphy (2002, chap. 6) reviews psychological evidence that concepts are part of our general knowledge of the world. Perhaps, therefore, we should envision concepts as involving rules such as *IF x is drunk, THEN x has impaired judgment*. Concluding that someone is drunk is not just matching a prototype, but is a kind of abductive inference based on rules. Kunda, Miller, and Claire (1990) found evidence for such inference in conceptual combination. Thus, concepts may be intimately connected with rules and examples, as well as with typical features.

Neurological Plausibility

Spreading activation between concepts in conceptual networks is similar to the way neurons activate each other by electrochemical impulses, but little is known about how concepts are realized in the brain. Brain-scanning techniques are being used to learn more about language organization. Posner and Raichle (1994) describe studies that monitored brain responses to words such as "hammer." These studies identified distinct areas of the brain involved in word perception and speech production. Ashby and Walrdron (2000) review evidence that the prefrontal cortex and basal ganglia contribute to concept learning. Another way of learning about the neural structure of the mental lexicon is to study deficits that occur in people who have had brain damage resulting from strokes. One patient had difficulty naming inanimate objects such as musical instruments, but could comprehend the names of foods, flowers, and animals relatively well; another patient suffered a stroke and lost the ability to name fruits and vegetables (Kosslyn and Koenig 1992). Artificial neural networks (chapters 7 and 9) have provided some ideas about how concepts might be stored and used in the brain.

Practical Applicability

One of the functions of education is to turn novices into experts in a domain such as physics or another branch of science. What is the difference between novices and experts? One answer might be that the latter have more rules, but educational research has suggested that experts have highly organized knowledge that can be described in turns of concepts or schemas (Bruer 1993). For example, students who are only beginning to learn physics have a schema for an inclined plane that includes only superficial features such as its angle and length. In contrast, the expert's schema immediately connects the concept of an inclined plane with the laws of physics that apply to it. Nersessian (1989) and Chi (1992) argue that science education is made difficult by the fact that students need to acquire abstract concepts such as *field* and *heat* that they erroneously treat as substances. Learning a complex discipline often requires active and intentional conceptual change (Sinatra and Pintrich 2003).

Any design problem involves concepts that can be represented by schemas or frames. In the context of building design, the concept of *beam* can be represented by a frame that has slots for span, load, support, and maximum stress (Allen 1992). This frame is part of a conceptual hierarchy that involves various kinds of beams, such as ones made of steel (I-beam or box-beam), concrete (reinforced or prestressed), and wood. Dym and Levitt (1991) describe an expert system called SightPlan that was developed to provide computer support for the task of locating temporary facilities on a construction site. SightPlan uses frames to represent concepts such as *construction site*, *power plant*, and various parts of power plants.

Although not so common as rule-based systems, frame-based systems have found various applications in artificial intelligence. Pure frame-based expert systems are rare, but some rule-based systems also use frames (Buchanan and Shortliffe 1984). The most ambitious current intelligent system is Cyc (originally short for "Encyclopedia"), which uses more than a million rules to encode a huge amount of commonsense knowledge that underlies intelligent performance in many domains (Lenat and Guha

1990). Cyc has a database of thousands of representations for many everyday concepts and objects, organized by means of an "ontology" of fundamental concepts such as *thing, individual,* and *animal.* The most general part of the Cyc ontology is available on the Web site listed below.

Summary

Concepts, which partly correspond to the words in spoken and written language, are an important kind of mental representation. There are computational and psychological reasons for abandoning the classical view that concepts have strict definitions. Instead, concepts can be viewed as sets of typical features. Concept application is then a matter of getting an approximate match between concepts and the world. Schemas and scripts are more complex than concepts that correspond to words, but they are similar in that they consist of bundles of features that can be matched and applied to new situations. The explanatory schema used in concept-based systems is as follows:

Explanatory target

Why do people have a particular kind of intelligent behavior?

Explanation pattern

People have a set of concepts, organized via slots that establish *kind* and *part* hierarchies and other associations.

People have a set of procedures for concept application, including spreading activation, matching, and inheritance.

The procedures applied to the concepts produce the behavior.

Concepts can be translated into rules, but they bundle information differently than sets of rules, making possible different computational procedures.

Discussion Questions

1. What concepts are learned? What concepts are innate?
2. What concepts can be defined? What concepts have typical features you can specify?
3. What concepts do not correspond to English words? What concepts are known only unconsciously?

4. Can concepts be reduced to rules? Can rules be reduced to concepts?
5. How does concept-based explanation differ from rule-based explanation?
6. How would you represent the concept of *mind?*
7. How are concepts related to things in the world?

Further Reading

Murphy 2002 is a comprehensive review of psychological research on concepts. Ward, Smith, and Vaid 1997 contains many articles on conceptual combination and creative use of concepts. Aitchison 1987 and Miller 1991 provide introductions to the mental lexicon. Frame-based AI systems are reviewed in Maida 1990 and Winston 1993. Langley 1996 has several chapters on computational models of concept learning. Margolis and Laurence 1999 is a collection of important articles on concepts.

Web Sites

Concept mapping: http://cmap.coginst.uwf.edu/info/

The Cyc ontology (organized set of thousands of concepts): http://www.cyc.com/

Visual thesaurus: http://www.visualthesaurus.com/online/index.html

WordNet, a lexical database for the English language: http://www.cogsci.princeton.edu/~wn/

Notes

Systems of hierarchically organized concepts are sometimes called semantic networks, although in AI the term "ontology" is used. In philosophy, ontology is the study of what fundamentally exists.

Inference by inheritance is used in object-oriented programming.

Conceptual change in the major revolutions in the history of science is analyzed in Thagard 1992.

5 Analogies

Imagine what life would be like if you always had to figure everything out from scratch, if every class were your first class, if every date were your first date. Fortunately, people are able to remember previous experiences and learn from them. But the learning that takes place does not always establish general knowledge of the sort that is found in rules and concepts. If you are a student in your second or later year of college, you may remember how you previously registered and chose your courses. That experience may have been too limited for you to capture it in a general rule or concept, but you can still use the particular experience to guide your choices for this year. If you ended up in a particularly disastrous course, you can try to avoid courses with similar topics and instructors. On the other hand, if you had a course that was a big success for you, you can try to enroll in similar courses.

Analogical thinking consists of dealing with a new situation by adapting a similar familiar situation. Human use of analogy is documented as far back as there are written records: Homer used analogies in the *Iliad*, and parables in the Bible serve to provide analogies between stories that are told and the readers' own situations. The importance of analogies in reasoning has long been recognized by philosophers (e.g., Mill 1974; Hesse 1966), but intense psychological and computational investigation is relatively recent. Evans (1968) developed the first computational model of analogical reasoning, and numerous models have been developed since then. Today, there are several research teams working to develop sophisticated models of analogy use. Keith Holyoak and I have developed a computational theory of human analogy use (Holyoak and Thagard 1995). In ways elaborated later, our view is similar to but also different from the influential view of Dedre Gentner and her colleagues (Gentner 1983, 1989;

Forbus, Gentner, and Law 1995; Forbus 2001). In artificial intelligence today, analogical reasoning is often called *case-based* reasoning, and numerous interesting applications have been developed (Kolodner 1993; Leake 1996). Douglas Hoftstadter and his associates (Hofstadter 1995; Mitchell 1993) have developed novel models of creative analogy use.

Representational Power

Do analogies say anything more than can be said with logic, rules, or concepts? For analogical reasoning, we need to be able to express two situations, the *target* analog representing the new situation to be reasoned about, and the *source* analog representing the old situation that can be adapted and applied to the target analog. Each analog is a representation of a situation, and the analogy is a systematic relationship between them. Representing analogs requires paying attention not only to predicates like "student" that apply to individuals but also to predicates like "teach" that describe a relation between two or more individuals. Interesting analogies hold between situations that share similar relations as well as similar features. Using the kind of logical notation introduced in chapter 2, we can represent some aspects of a course called Philosophy 999 as follows:

1. instructor (Repulso, Phil999); i.e., Professor Repulso is the instructor of Philosophy 999.
2. dull (Repulso); i.e., Repulso is dull.
3. difficult (Phil999); i.e., the course is difficult.
4. enrolled-in (you, Phil999); i.e., you're stuck in the course.
5. grade (you, Phil999, low); i.e., you're getting low grades in the course.

In addition, it may be crucial to your low grade in the course that the instructor was dull and the course was difficult:

6. cause (2 & 3, 5); i.e., the dull instructor and difficult course are causing your low grade.

Statement 6 exemplifies a kind of representational power involving causal relations between statements that is significant for many important analogies. If you are considering taking Psychology 888, which also has a dull instructor and a reputation for being difficult, you may infer by analogy to Philosophy 999 that you are likely to get a low grade in the course and therefore avoid it. Here, Philosophy 999 is the familiar source analog and Psychology 888 is the target analog that you reason about based on the source.

More positive analogies can be used in course selection rather than course avoidance. If there has been a course that you have liked and if you can identify the features of the course that caused you to like it, then you can look for similar courses that you are also likely to enjoy. A sophisticated analogy user will ignore superficial similarities, such as that two courses both have names with the same number of letters. But how are superficial similarities to be distinguished from important ones? For one student, it may not matter what time of day a course meets; for another student, who is most alert in the morning, the time of day will be a relevant factor in choosing courses similar to ones that have already proved to be enjoyable. The key to noticing relevant differences is to appreciate the causal relations that produced outcomes relevant to your goals in taking the class. Hence, representation of analogies needs to include representation of causal relations like the one in statement 6.

Usually, analogs can be represented as collections of the kinds of representations we have already seen. Analogs are like concepts and unlike statements in logic and rules in the way that they bundle together packages of information, but they are like simple statements and unlike concepts and rules in that the information they contain describes only a particular situation. For example, the representation of Philosophy 999 in statements 1–6 provides a package of information about a course, but the pieces of information in the package apply only to that course, not to courses in general. In contrast, analogical schemas, discussed below in the learning section, include general information, like rules and concepts and unlike representations of source and target analogs.

Analogs are sometimes represented using visual images of the sort discussed in chapter 6. Figure 5.1 presents a visual analogy. People use visual analogies when, for example, they use a mental picture of a familiar building to guess how to get around in a similar unfamiliar one. Emotions can also be involved in the representation of analogs, as chapter 10 describes.

Computational Power

If you are solving problems in a very familiar domain where you have lots of expertise, you can put to work general knowledge captured in rules and

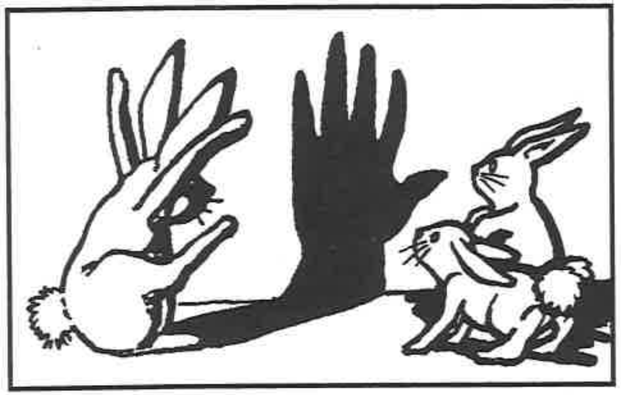


Figure 5.1
A humorous visual analogy. Reprinted by permission from Holyoak and Thagard 1995, p. 14.

concepts. Analogical reasoning, in contrast, becomes useful when you have some previous experience with a domain but little general knowledge of it. Hence analogies can be computationally powerful in situations when conceptual and rule-based knowledge is not available.

Typically, analogical reasoning proceeds in four stages:

1. You face a target problem to be solved.
2. You remember a similar source problem for which a solution is known.
3. You compare the source and target problems, putting their relevant components in correspondence with each other.
4. You adapt the source problem to produce a solution to the target problem.

Understanding analogical reasoning computationally requires specifying procedures for the stages of remembering (retrieving from memory), comparison (mapping the source and target analogs to each other), and adaptation.

Retrieving potentially relevant source analogs from memory is computationally very difficult. How many experiences have you had in your life? If you have accomplished just 10 tasks per day for the past 15 years, you have potentially stored in memory 54,750 task solutions. Faced with a current task for which you hope to find a new solution, you would have

somehow to compare the new problem against the very large number of stored solutions. How does the mind select usable experiences from its vast store? Suppose your current task is to register for the upcoming term or semester. Will you recall every time you had to register for something? Every time you had to stand in line? Every time you were frustrated? Every time you did something in the rain?

Current computational models of analog retrieval disagree about the factors that make for effective retrieval and that account for both the successes and failures of human use of analogies. Combining the ideas of many researchers, Keith Holyoak and I argue that retrieval is governed by three constraints: similarity, structure, and purpose (Holyoak and Thagard 1995). Two analogs are similar to each other at a superficial level if they involve similar concepts. Thinking about registering now will make you think about previous cases of registering and other bureaucratic operations that are conceptually related to registering. The similarity of visual analogs is not just conceptual, but also involves their visual appearance. One car may remind you of another car because they have similar shapes or colors.

However, powerful analogies involve not just superficial similarities, but also deeper structural relations. If registering this year is causing you to miss your favorite afternoon TV show, you may remember a previous time when paying your tuition caused you to miss a show. The correspondence between the two situations is then not just that they both involve bureaucratic tasks and missing a TV show, but the higher relation that the bureaucracy caused you to miss the show. To fully satisfy the structure constraint, two analogs must align exactly:

Target	*Source*
cause: register (you)	cause: pay-tuition (you)
miss (you, TV show)	miss (you, TV show)

The target analog on the left says that your registering caused you to miss your TV show. The source analog on the right says that your paying tuition caused you to miss your TV show. Even though the two situations are different in that one involved registration and the other paying tuition, they have exactly the same structure, since the relations "miss" and "cause" align perfectly.

The third constraint on retrieval is purpose: you want to remember cases that will help you to solve your current problem. In human memory (and in computer databases) there are vast amounts of information, so that retrieving all and only potentially useful information is a difficult psychological and computational problem. The problem of finding and applying source analogs to target problems can be eased by making the purpose of the analogy one of the constraints on its development. For example, if your purpose in using the analogy between registering and paying tuition is to show bureaucratic inefficiencies at your college or university, then the purpose should encourage remembering other inefficiencies.

Holyoak and I contend that it is these three constraints operating in parallel that make possible retrieval of relevant analogs from the vast number of potentially relevant ones. Other researchers on analogy disagree with us. Forbus, Gentner, and Law (1995) emphasize the role of similarity in retrieval, giving structure and purpose less of an impact. On the other hand, many researchers on case-based reasoning have stressed the constraint of purpose, urging that computer memories be indexed in ways that encourage retrieval of analogs relevant to current goals (Schank 1982; Kolodner 1993). For building expert systems, they propose developing a "generally applicable indexing vocabulary" that will apply to all domains. Whether human memory is indexed in this way can be determined only by psychological experiments (see below).

Once a potential source analog has been retrieved from memory, it must be mapped with the target problem to find the correspondences that can suggest a solution. If the two analogs are very similar, mapping is quite trivial, as when you map the current registration to the previous one. But creative analogies often involve a leap, as in the following example (Dennett 1991, 177):

> The juvenile sea squirt wanders through the sea searching for a suitable rock or hunk of coral to cling to and make its home for life. For this task, it has a rudimentary nervous system. When it finds its spot and takes root, it doesn't need its brain anymore, so it eats it! (It's rather like getting tenure.)

How is a professor getting tenure like a sea squirt eating its brain? Grasping the comparison requires noticing a set of mappings: between sea squirt and professor, between finding a rock and getting tenure, and so on. Cognitive science researchers differ on which constraints play a role in such mappings. Gentner (1983, 1989) maintains that mapping is a matter of noticing structural correspondences, but Holyoak and I argue that superficial similarities and purpose also contribute to analogical mapping. Both sides of this dispute have developed computer models that aid in testing the competing theoretical claims.

If a source analog maps neatly onto a target problem, copying over the relevant part of the source to the target can generate a solution. If you solved your registration problem last time by taking an evening psychology course, you might solve it this time by taking another evening psychology course. If the exact same solution is not possible, you might adapt the previous solution somewhat—for example, by taking an evening philosophy course. The most sophisticated accounts of adaptation have been offered by researchers in case-based reasoning. Kolodner (1993) lists ten methods for adapting previous solutions, from simple substitutions like replacing a philosophy course by a psychology course, to more complex derivations like writing a computer program in one programming language by systematically adapting a program written in another computer language.

Problem Solving

Planning It should be obvious from the above discussion how analogies can contribute to solving planning problems such as registering for good courses. Under the same heading we can also put solving the kinds of problems that students are assigned in science and mathematics courses. A textbook chapter in a technical field often includes examples that show how to go about solving problems where the student is given some information and has to find an answer. For example, given some information about a chemical substance, you might have to calculate additional features of it such as density. Analogy is not the only way to go about solving such problems, but it is often useful to try to solve the exercises at the end of the chapter by flipping back and relating them to solved problems provided in the main text.

Analogies can be very useful in problem solving, but they do not always provide the best way to approach a new problem. There is always the danger that a selected analog will not have the deep relevant similarity that is needed to provide a solution to a target problem. If the target problem is genuinely novel, then no previous solution will apply and analogies will only mislead. In military planning, generals often fight the last war, using outmoded analogs. Similarly, although students can greatly simplify new assignments by perceiving them as analogous to previous ones, this strategy can backfire if the new problems require novel approaches. Techniques learned in mathematics courses are of limited use in courses that require writing essays.

Decision Decisions about what actions to choose are also often made analogically. Legal reasoning frequently makes reference to previous cases that serve as precedents: these are source analogs that get mapped to the current target case. Historians have documented numerous cases of political decisions heavily guided by analogies. For example, when the United States debated in 1991 whether to attack Iraq in retaliation for Iraq's invasion of Kuwait, arguments pro and con often concerned historical analogs. President George Bush compared the Iraqi leader, Saddam Hussein, to Adolf Hitler, suggesting that the invasion of Iraq was as legitimate as the World War II invasion of Germany. Critics of the plan to invade Iraq preferred a different comparison, to the United State's disastrous involvement in Vietnam. These analogies resurfaced in 2003 when the United States again invaded Iraq. Analogies can improve decision making by both suggesting previously successful solutions and reminding leaders of previous disasters. All too often, however, decision makers become fixated on a single previous analog and do not consider how a variety of source analogs might suggest different actions to choose from.

Explanation Analogies are also an important source of solutions to explanation problems, including both educational situations where teachers must convey what they understand to students and research situations where brand-new explanations are being generated. Listen for your instructors' analogies in your next few lectures. Teachers often try to help students understand unfamiliar things by comparison with what the students already know. For example, I might explain the British sport of cricket to an American by comparing it to baseball, since both involve bats, balls, and running between positions. Analogical explanations are often limited by the fact that the things being compared have many differences as well as similarities, but they can be a crucial part of getting someone new to a domain up and running. Later in this chapter, the section on educational applications discusses how to use analogies effectively in teaching.

Analogical explanation abounds in cognitive science. As we have already seen, the fundamental analogy in cognitive science is between the mind and the computer: we attempt to explain how the mind works by modeling it as a computer. The analogy is complex, however, since sometimes we get new ideas about what computing can be like by studying the mind and brain. Early ideas about computing drew heavily on psychological views, and recent connectionist computational models discussed in chapter 7 have been influenced by new views about the brain.

Learning

Analogical thinking involves three kinds of learning. The most mundane is simply the storage of cases based on previous experience. When you figure out how to solve a problem, you can store your solution in memory. This storage does not involve analogy as such, nor does it require the kinds of generalization that underlie forming concepts and rules. But it is a necessary prelude to analogical thinking and constitutes learning at a low level. The second kind of learning is directly the result of analogizing, when you adapt a previous case to solve a new problem. This is again a more particular kind of learning than we saw with rules and concepts, since all you have learned is how to solve the particular new problem. This kind of inference can be abductive if you adapt a previous explanation problem to suggest a new explanatory hypothesis. For example, if a friend is late for a party, you may remember a previous case where somebody was late for a party because of a flat tire, and conjecture analogically that your friend in the current case might have had car trouble.

The third kind of learning introduces a general element. If you use a source analog to solve a target problem, you can abstract from the source and target and form an analogical *schema* that captures what is common to both of them. For example, figuring out how to register for courses this year based on how you registered last year can lead to an abstracted schema for registration. Analogical schemas are very much like the schemas (concepts) discussed in chapter 4, except they should not be expected to have the same degree of generality, since they are generalizations from only two instances. Having registered for courses twice, you may be able to abstract a description of registration from the two situations, an abstraction that includes rough rules concerning how to get the courses you desire. An abstracted analogical schema may be very useful for future problem solving, since it should include those aspects of the source and target analogs that are shared and relevant to problem solution. We will see in the section on psychological plausibility that forming analogical schemas improves problem solving.

Language

Analogy plays an important role in the production and comprehension of language, since it underlies the use of metaphor. When people say that Britney Spears is the new Madonna, they do not literally mean that Britney Spears *is* Madonna. Rather, they are pointing out some systematic similarities between the two: both are female rock stars who perform provocatively. Similarly, the statement that life is a battlefield evokes a systematic comparison between a target (life) and a source (war). Other metaphors, such as that life is a party, evoke very different comparisons. The information superhighway is not a highway, but it is analogous to one in that it provides a fast and effective way of moving electronic data.

Some language theorists see metaphor as a rather deviant use of language since it does not seem to use language literally: why not just say what you mean? In contrast, various linguists, philosophers, and psychologists have viewed metaphor as a pervasive and valuable feature of language, not as an exceptional or deviant use (Glucksberg and Keysar 1990; Lakoff and Johnson 1980). All metaphors have as their underlying cognitive mechanism the sort of systematic comparison that analogical mapping performs, although metaphor may go beyond analogy by using other figurative devices to produce a broader aura of associations. Both the generation of a metaphor by a speaker and its comprehension by the hearer require the perception of an underlying analogy. If I tell you that Professor Repulso is a sea squirt, you should be able to understand that I am not saying that he is a marine animal with a saclike body, but rather that there is some relevant similarity between his mental history and that of the sea squirt.

Psychological Plausibility

Many psychological experiments have examined how people use analogies. I will mention only a few examples that show analogy at work in problem solving, learning, and language use.

How would you go about solving the problem in box 5.1? Most people find it hard to think how the doctor can use the rays to kill the tumor without destroying the healthy tissue: Gick and Holyoak (1980) found that only about 10 percent of college students could produce a good solution.

Box 5.1
The tumor problem (from Gick and Holyoak 1980).

> Suppose you are a doctor faced with a patient who has a malignant tumor in his stomach. It is impossible to operate on the patient, but unless the tumor is destroyed the patient will die. There is a kind of ray that can be used to destroy the tumor. If the rays reach the tumor all at once at a sufficiently high intensity, the tumor will be destroyed. Unfortunately, at this intensity the healthy tissue that the rays pass through on the way to the tumor will also be destroyed. At lower intensities the rays are harmless to healthy tissue, but they will not affect the tumor either. What type of procedure might be used to destroy the tumor with the rays, and at the same time avoid destroying the healthy tissue?

In contrast, 75 percent of college students could produce a good solution to the tumor problem if they were told the fortress story in box 5.2. At first glance, the fortress story has nothing to do with the tumor problem. But many people are able to use the solution in the fortress story that involves the army dividing up and then converging on the fortress to generate a solution to the tumor problem: instead of using a single high-intensity ray, the doctor could administer several low-intensity rays from different directions.

This example illustrates the simple kind of analogical learning where a new problem is solved by adapting an old one. Using the same problem, Gick and Holyoak (1983) investigated how students learn analogical schemas from more than one example. In addition to the fortress story, some students were given a story about a firefighter who extinguished an oil-well fire by using multiple small hoses. The fire was put out by converging water, just as the fortress was conquered by converging armies. Students who had two such examples and were instructed to reflect on the similarities between them were more likely to be able to remember to apply a convergence solution to the tumor problem than students who had only received a single analog. Learning analogical schemas thus contributes to more effective problem solving.

Psychological experiments concerning language have been done to address the question of metaphor use. Glucksberg and Keysar (1990) have

Box 5.2
The fortress story (from Gick and Holyoak 1980).

> A small country fell under the iron rule of a dictator. The dictator ruled the country from a strong fortress. The fortress was situated in the middle of the country, surrounded by farms and villages. Many roads radiated outward from the fortress like spokes on a wheel. A great general arose who raised a large army at the border and vowed to capture the fortress and free the country of the dictator. The general knew that if his entire army could attack the fortress at once it could be captured. His troops were poised at the head of one of the roads leading to the fortress, ready to attack. However, a spy brought the general a disturbing report. The ruthless dictator had planted mines on each of the roads. The mines were set so that small bodies of men could pass over them safely, since the dictator needed to be able to move troops and workers to and from the fortress. However, any large force would detonate the mines. Not only would this blow up the road and render it impassable, but the dictator would destroy many villages in retaliation. A full-scale direct attack on the fortress therefore appeared impossible.
>
> The general, however, was undaunted. He divided his army up into small groups and dispatched each group to the head of a different road. When all was ready he gave the signal, and each group charged down a different road. All of the small groups passed safely over the mines, and the army then attacked the fortress in full strength. In this way, the general was able to capture the fortress and overthrow the dictator.

shown that people find metaphorical meanings even when instructed to find literal meanings. In one study, college students were asked to decide whether or not sentences such as "Some desks are junkyards" were literally true. The students were slower to correctly respond "no" to a sentence that was literally false when it also had a metaphorical interpretation, as in the above example, than to respond to literally false sentences such as "Some desks are roads" that lack a metaphorical interpretation. Similar findings have been obtained for sentences that can be interpreted *both* literally and metaphorically. Keysar (1990) presented students with sentences such as "My son is a baby" in contexts that manipulated whether the sentence was true or false, literally or metaphorically. The students were instructed to press a key as quickly as possible to indicate the *literal* truth value of the sentence. If the sentence was literally false in the context, the students decided more quickly if it was also metaphorically false; if the

sentence was literally true, they decided more quickly if it was also metaphorically true. Such findings imply that literal and metaphorical processing interact with each other. Metaphorical interpretation appears to be an obligatory process that accompanies literal processing, rather than an optional process that occurs after literal processing.

Neurological Plausibility

Neurological research on analogical reasoning is just beginning. Boroojerdi et al. (2001) found that the left prefrontal cortex is involved in analogical reasoning by determining that magnetic stimulation of that part of the brain speeds up solution times for solving analogical problems. This is consistent with recent findings that reasoning involving complex relations, which is crucial for analogical thinking, also involves the left prefrontal cortex (Christoff et al. 2001; Kroger et al. 2002).

Recent computational models of analogy are moving in the direction of using artificial neural networks that approximate to real neuronal behavior. Hummel and Holyoak (1997, 2003) have developed a neural network model of analogy that uses synchrony of neuronal firing to represent relational information. Eliasmith and Thagard (2001) use a different technique to produce representation of complex relations that are distributed over multiple neurons. Neural network models of cognition are described in chapters 7 and 9.

Practical Applicability

As we saw in the section on computational power, analogies can make substantial contributions to explanation. Hence, they potentially have great value in education. Effective teachers often try to help students understand the unfamiliar by systematically comparing it to the familiar. There are, however, many potential pitfalls in educational use of analogies. Avoiding pitfalls requires careful attention to what students know and to how analogies are used and misused.

Here are some brief recommendations for how educators can more successfully use analogies (see Holyoak and Thagard 1995, for more discussion and justification):

1. Use familiar sources. There is no point in explaining science or some other complex, unfamiliar target in terms of something that is equally unfamiliar. You cannot explain the structure of atoms to young children by analogy to the solar system if they do not know the structure of the solar system.
2. Make the mapping clear. With a good analogy, students should be able to figure out for themselves the basic correspondence between the source and target, but some guidance may facilitate finding a mapping. For example, in cognitive science it is important to indicate which aspects of mind correspond to which aspects of computers.
3. Use deep, systematic analogies. Instead of superficial feature comparisons, the most powerful analogies use systematic causal relations that provide clear relevance to the students' goals.
4. Describe the mismatches. Any analogy or metaphor is incomplete or misleading in some respects. Some educators have concluded that analogies are therefore too misleading to be effective in teaching, but the solution to the problem is not to abandon use of analogies but rather to indicate where they break down. No one should expect the information superhighway to have a white stripe painted down the middle.
5. Use multiple analogies. When one analogy breaks down, another can be added to provide understanding of what has been incompletely presented.
6. Perform analogy therapy. Find out what analogies students are already using and correct them as necessary.

These maxims are good advice not only for educational use of analogies but also for all the other uses of analogy, including problem solving and decision making.

Analogies are often a fertile source of creative designs. Georges de Mestral invented Velcro after he observed how burrs stuck to his dog. Alexander Graham Bell modeled the telephone partly on the human ear. New adhesives have been invented based on how the feet of gecko lizards enable them to walk up walls. Industrial designers often use the technique of reverse engineering, where they take a competitor's product apart and figure out how to produce an analogous product.

Analogies and metaphors have also contributed to computer design and discussion of computer-human interaction. The Macintosh interface, which was copied (analogically) by the Windows program on PCs, uses a desktop analogy: the screen is like a desk on which the user lays out various documents and folders. Spreadsheets make numerical calculations using a format that is analogous to a paper ledger. Word processors are in some respects like the typewriters they have replaced.

Effectiveness of design can be hampered by unsuspected analogies that users employ. One computer company reported that a woman phoned to complain that she had her foot on the mouse on the floor but the computer would not start; she seems to have thought it was like a sewing machine. A man complained that his computer would not fax a piece of paper he was holding up to its screen, which he apparently thought was like a copying machine. Designers need to consider positive analogies for consumers to use, but they also need to watch out for misleading analogies that users may come up with themselves. Customers may need analogy therapy.

Kolodner (1993) describes dozens of case-based reasoning systems. Although they differ in particular retrieval and mapping mechanisms, all fundamentally employ analogical reasoning to solve new problems on the basis of old. Lockheed, for example, uses a case-based reasoning system called Clavier to recommend how to arrange airplane parts in a large pressurized convection oven called an autoclave (Hinkle and Toomey 1994). The cases (source analogs) are records of previous loads placed in the autoclave. Although experts on autoclave use were not able to express their technique in rules, a system that stores, retrieves, and adapts cases has proven very effective. Hastings, Branting, and Lockwood (2002) developed a system that uses case-based reasoning along with rules to provide advice about how to deal with grasshopper infestations in Wyoming.

Summary

Analogies play an important role in human thinking, in areas as diverse as problem solving, decision making, explanation, and linguistic communication. Computational models simulate how people retrieve and map source analogs in order to apply them to target situations. The explanation schema for analogies is as follows:

Explanation target

Why do people have a particular kind of intelligent behavior?

Explanatory pattern

People have verbal and visual representations of situations that can be used as cases or analogs.

People have processes of retrieval, mapping, and adaptation that operate on those analogs.

The analogical processes, applied to the representations of analogs, produce the behavior.

The constraints of similarity, structure, and purpose overcome the difficult problem of how previous experiences can be found and used to help with new problems. Not all thinking is analogical, and using inappropriate analogies can hinder thinking, but analogies can be very effective in applications such as education and design.

Discussion Questions

1. How do analogs (cases) differ from rules and concepts?
2. When is analogical problem solving likely to be useful?
3. What are the main stages in analogical thinking? What constraints figure most prominently at each of those stages?
4. What are the main potential drawbacks of thinking by analogy?
5. How do analogies contribute to creativity? What other sources of creativity are there?

Further Reading

Gentner, Holyoak, and Kokinov 2001 contains articles describing many current approaches to analogy. Hall 1989 reviews artificial intelligence work on analogy to that date. French 2002 surveys more recent computational models. Holyoak and Thagard 1995 gives a psychologically oriented survey. For a review of Gentner's work on analogy, see Gentner 1989. Kolodner 1993 is an excellent survey of case-based reasoning work in artificial intelligence; see also Leake 1996. For an entertaining review of the work of Hofstadter's group on creative analogies, see Hofstadter 1995.

Web Sites

Artificial intelligence and case-based reasoning: http://www.ai-cbr.org/theindex.html

Case-based reasoning: http://www.cbr-web.org/

Conceptual metaphor: http://cogsci.berkeley.edu/lakoff

Dedre Gentner's home page: http://www.psych.northwestern.edu/psych/people/faculty/gentner/

Keith Holyoak's home page: http://www.psych.ucla.edu/Faculty/Holyoak/

6 Images

How many windows are there on the front of your house or apartment building? How did you answer that question? If you have never counted the windows before, you must have found a way to count them now. Perhaps you compiled a list of all the rooms that are on the front of your building and did a verbal count of their windows, but many people answer this kind of question by making a mental picture and doing a visual count. Similarly, try to remember how you get from your home to your college or university. Although you may have a purely verbal memory of how to do this ("Go to the traffic light at Main St. and turn right"), many people remember such routes by constructing a series of mental images of the roads, buildings, and other landmarks along the way.

Many philosophers, from Aristotle through Descartes and Locke, assumed that picturelike images are an essential part of human thought. In the early days of modern psychology in the late nineteenth century, researchers such as Wilhelm Wundt studied how people think with imagery and some even claimed that there was no thought without imagery. The rise of behaviorism in the twentieth century made talk of mental images and other internal representations scientifically unrespectable. But the return of cognitive psychology in the 1960s made imagery once again a suitable object of investigation, and researchers such as Paivio (1971) and Shepard and Metzler (1971) began doing experiments with visual images. Many experiments ensued, and computational models of visual imagery began to appear (Kosslyn and Shwartz 1977; Funt 1980). Some cognitive scientists remain skeptical that human thinking involves pictorial representations that are different from verbal ones (Pylyshyn 1984, 2002). But numerous computational, psychological, and neurological considerations suggest that the mind thinks with pictures as well as words.

Although cognitive scientists interested in imagery have concentrated on visual representations, we should not ignore images connected to nonvisual perception. What does a pepperoni pizza taste like? If you have ever had one, you may be able to form a mental image of the taste and smell, and use it to decide whether something else—say, a submarine sandwich—tastes like a pepperoni pizza. Does a growth of beard feel like sandpaper? To answer this, you may form a tactile image of each touch and compare them. How do you hit a baseball to the opposite field, slam-dunk a basketball, or clean a mirror? If you have regularly experienced these physical activities, you may be able to construct a motor image of the bodily sensations associated with them. Finally, people can have emotional images. How did you feel when you heard that you had been admitted to your college or university? Did your friends feel the same? Chapter 10 discusses emotions and consciousness. The rest of this chapter will concentrate on visual images, the kind most investigated to date.

Vision

For people with normal vision, seeing things seems automatic and easy. You look at a room and immediately pick out the furniture and people in it. The complexity of vision becomes apparent, however, when you try to get a computer to do the same thing. It is easy to point a video camera at a room and store the image as a set of pixels, the dots that make up an image on a TV screen. But extracting information from thousands or millions of pixels is very difficult, since the image captured by the video camera may be highly ambiguous. If a person is sitting in a chair, the pixels will reveal only part of the chair, so the computer must somehow infer that there is a chair even though it cannot see anything that matches a standard chair. Some parts of the room may be in brighter light than other parts. A rectangular object on the wall might be a picture, or it might be a mirror reflecting other parts of the room. In the past few decades, computer vision has made substantial progress, enabling robots to identify and manipulate objects under simplified circumstances. But robotic vision remains crude compared to the power of human vision.

Consider the drawing in figure 6.1. If you shift your concentration between the top and the bottom of the cube, you should be able to make it flip back and forth, seeing first one face as the front and then another face as the front. How does this happen? Light reflects off the picture into your eyes and onto your retina, which consists of millions of light-detecting cells. But an enormous amount of processing is required before your brain can interpret the picture as a cube. Your brain must detect edges, distinguishing the lines from the background. In figure 6.1, edge detection is trivial, but this task becomes much more difficult if there are subtle variations in brightness, grays as well as black and white. Moreover, your brain does not receive a single image like what a video camera would produce, but rather gets information from two eyes with slightly different perspectives on objects. The perspective differences make possible your ability to appreciate distances and see objects in three dimensions.

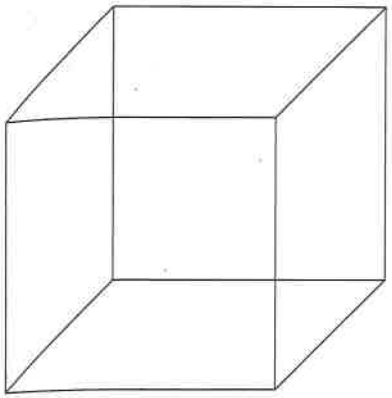
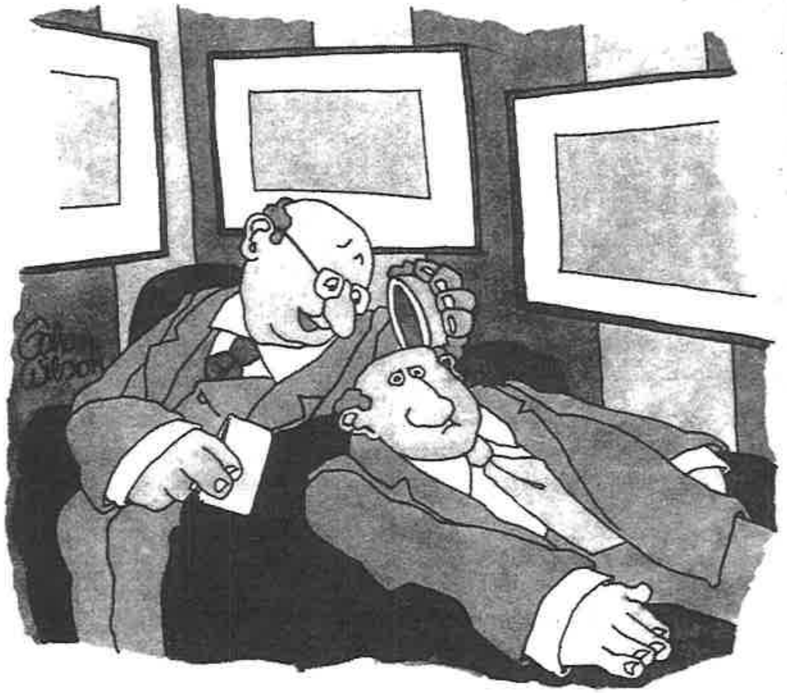


Figure 6.1
The Necker cube. The top edge can be seen either as being at the front or at the back of the cube. Try to make it flip back and forth by concentrating on different edges.

The brain manages to combine inferences about edges, perspective, colors, and other information into a coherent interpretation of objects far more complicated than the cube in figure 6.1. See Marr 1982 and Kosslyn 1994b for much more about visual information processing. The result of all this inference is a visual image. Such images do not depend on an object's being present to the eyes, for we can store the images in memory, retrieve them, and manipulate them in ways that contribute to a variety of mental tasks.



"Looking good!"

Figure 6.2
Drawing by Gahan Wilson. © 1994, The New Yorker Magazine, Inc. Reprinted by permission.

Representational Power

Why do people often say that a picture is worth a thousand words? Pictures can usually be described in words. For example, we can say that figure 6.2 depicts one man sitting behind another and looking into the top of his head. Given enough sentences, we could provide a much fuller description. But the pictorial representation has various advantages. The verbal description might contain the information that the man in the chair is close to the man with his head open, who is on a couch. We could then verbally infer that the chair is close to the couch. Using the pictorial representation, however, no inference is necessary: we can just *see* that the chair is close to the couch. Figure 6.2 is an external representation that we see with our eyes. But if you cover up the picture for a moment, you may still be able to form a mental image of the picture and answer some questions about it. Roughly how old are the men? Is either of them bald? Is either wearing a tie?

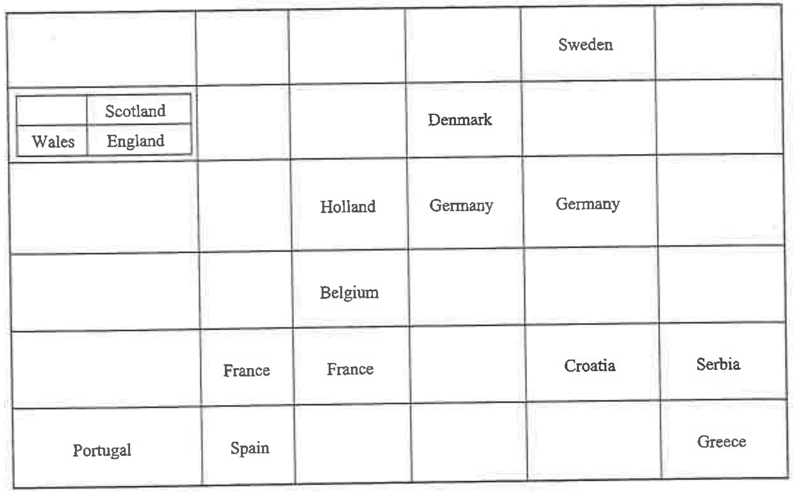
Pictures and visual mental images provide powerful ways of representing how things look and how they are spatially arranged, but not all information is naturally represented in pictures. Abstract sentences like "Justice is fairness" are not visually representable, and general sentences like "All dinosaurs are extinct" are very awkward to represent pictorially. Similarly, causal statements such as "Smoking causes cancer" and "If you get a cold, then you will cough" are not straightforwardly represented by pictures. Hence, visual images complement but do not replace verbal representations of the sort we have seen in the previous chapters.

Earlier chapters assumed that representations are fundamentally verbal: the rules, concepts, and analogs discussed were all presented in words. But these structures may have visual forms as well. A rule might have the structure *IF <picture 1>, THEN <picture 2>*, providing a kind of movie in which picture 1 is followed by picture 2. A concept might be pictorial—for example, if my prototype for a dog is represented not by a set of features but by a picture of a dog that has those features. Similarly, source and target analogs can have visual representations such as the rabbit and shadow in figure 5.1. Hence, in addition to verbal rules, concepts, and analogs, there may be visual rules, concepts, and analogs.

What is the structure of mental images? Kosslyn (1980) and Glasgow and Papadias (1992) proposed that the mind uses arraylike structures to perform visual tasks. For example, we might represent Europe using the array shown in figure 6.3. More recently, Kosslyn (1994b) has argued that the human brain uses various kinds of neural networks to represent spatial information (see the section below on neurological plausibility).

Computational Power

Much thinking that can be done with images can also be done with words, but verbal thinking may be much more awkward for some tasks. Visual thinking is likely to be useful for any problem whose solution depends on visual appearance or spatial relationships. Visual representations, both



				Sweden	
Scotland Wales England			Denmark		
		Holland	Germany	Germany	
		Belgium			
	France	France		Croatia	Serbia
Portugal	Spain				Greece

Figure 6.3
Map of Europe represented as an array. Adapted with permission from Glasgow and Papadias 1992, p. 373.

mental and external, are accessible to different kinds of computational procedures than verbal representations:

1. *Inspect* Imagine a plate that has a knife to the left of it and a fork to the right of it. Is the knife to the left or the right of the fork? The answer could be inferred verbally using the logical properties of the relations "left" and "right," but more immediately the answer could come just by looking at the image formed and seeing that the knife is to the left of the fork. This procedure can also be used to compare two representations by inspecting them both.

2. *Find* Where do you keep your shoes at home? To remember, you might do a mental scan of your room or rooms to find the spot they are likely to be.

3. *Zoom* Does a frog have a tail? Some people answer this question by forming a mental image of a frog and then zooming in to look in more detail at its behind, just as you can look more closely at part of a picture.

4. *Rotate* What does a capital letter "E" look like when it is flat on its back? One way to answer this question is to rotate the letter mentally until it is on its back.

5. *Transform* Follow these instructions from Finke, Pinker, and Farah 1989: Imagine the letter "B." Rotate it 90 degrees to the left. Put a triangle the same width as the rotated "B" directly below it and pointing down. Remove the horizontal line. Many people see the resulting figure as a heart or double ice-cream cone. We seem to be able to alter and combine visual representations in powerful ways, including flipping and juxtaposing them as well as rotating them.

Operations such as these five make possible kinds of problem solving different from the verbal kinds considered in earlier chapters. To answer the question of whether all your shoes have the same number of holes for laces, you might retrieve an image of your closet, scan it to find your shoes, zoom in to inspect your shoes, and transform the shoe images to juxtapose laces to compare the number of holes. On the other hand, if you have only one pair of shoes, and you know the rule that two shoes from the same pair have the same number of holes, it might be easier to deduce the answer without recourse to mental imagery.

Problem Solving

Planning Suppose you have many errands to do: picking up groceries, mailing a parcel, and dropping off dry cleaning. Previous chapters suggested verbal ways in which you might plan how to accomplish these tasks in a reasonably efficient way. A set of IF-THEN rules might have guided you to the grocery store, post office, and dry cleaner's, or perhaps previous experience with these tasks might have guided you with a verbal analog or schema. Alternatively, you might construct a plan visually, imagining yourself driving into the grocery store parking lot, then driving out to the post office, and finally parking at the dry cleaner's. Such visual planning may employ a mental map that you have constructed that encodes the spatial relations of the places you have to go. Not everyone employs such mental maps: some people function better with verbally encoded landmarks. But for many others, getting around in the world is very much helped by being able to use visual images to figure out where they are and how they can get to where they want to be.

Planning with visual representations involves steps similar to those in rule-based problem solving described in chapter 2, except that the steps are executed visually. You must first construct visual representations of the starting and goal states, then construct a visual path from the start to the goal. Visual transformations can be useful in solving construction problems, such as how to build a bridge connecting two banks of a river, and even for more mundane problems in the sciences. Problem solvers often use diagrams as an external aid to supplement the more temporary benefits of mental images. In geometry, for example, it can be very helpful to draw diagrams of figures and angles as an aid to working out how to draw figures. Students solving science problems often make use of diagrams that make complex objects such as springs, molecules, and chromosomes more comprehensible.

Decision Little research has been done on the contribution of imagery to decision making. But suppose you are trying to decide whether to wear your blue or your brown jacket. You might imagine how each would look with the other clothes you are planning to wear, so that the decision about what to wear would be the result of a comparison of visual images. Similarly, if you are trying to decide what to order in a restaurant, your decision might be based in part on imagining what different dishes might taste like. Emotional images can also be important for decision making, as we will see in chapter 10.

Explanation Visual reasoning may be very useful in generating explanations. The great inventor Nikola Tesla could reportedly diagnose the faults in complex machinery just by forming a mental image of the machinery and running it in his head to see where breakdowns might occur. Visual explanation has not been much studied in psychology or artificial intelligence, but there is reason to believe that it is common in scientific and everyday thinking. Look at a map of the world that shows the continents of Africa and South America. Now slide these two continents together until the bulge that constitutes Brazil fits into and under West Africa. Early in this century, the fit between these two continents suggested to Alfred Wegener that they had once been joined, and he formed the hypothesis of continental drift to explain how they had come apart. This hypothesis can be stated in purely verbal terms, but the fit between Africa and South America is best represented visually and can be explained by a visual joining of the two continents. This joining mentally reverses a spatial separation conjectured to have happened long ago. As with planning, visual explanation is not a replacement for verbal reasoning, but provides a valuable complement to it.

Learning

Athletes are often coached to improve their performances by using imagery, and there is experimental evidence that practicing by mental imaging can improve performance if mixed with actual practice (Goss et al. 1986). Someone waiting to perform a dive or to hit a baseball can imagine accomplishing the task perfectly, using both visual and motor images. Running the task through your mind can actually help you to do it better when the time comes.

Images can also be useful for generalization, as when someone uses pictures of members of a category such as *elephant* to form a fairly general mental picture of an elephant. The resulting visual representation of an elephant ignores incidental information about particular elephants (e.g., carrying a rider) in favor of general properties (e.g., being gray, wrinkled). Imagistic learning of generalizations has not received much experimental or computational attention.

Abductive learning can also be visual. If you find a long scratch on the door of your car, you can generate various verbal explanations of it. But you might also construct a kind of mental movie in which someone drives up beside you in the parking lot and opens a door that scrapes along your car just where the scratch appears. Your abductive inference that another car scraped your door is generated visually, by constructing a sequence of pictures that shows how the scratch might have come about. Other pictures are possible too, such as one showing a shopping cart rolling into the car or keys scraping along it. Shelley (1996) describes how archaeologists use visual abduction when they generate explanations of ancient objects.

Language

Language is essentially verbal, so how could imagery be relevant to the use of language? We saw in chapter 5 that language is not just a matter of syntax and simple semantics, but is frequently metaphorical. As Lakoff and Johnson (1980) have pointed out, many metaphors are visual in origin: he's *up* today, she's *on top* of her job. Lakoff (1994) contends that much understanding involves image schemas, which are general concepts that have a visual component. For example, behind understanding of categories is visual understanding of containers: an object can be *in* or *out* of a category such as *dog*, and it can be *put into* or *removed from* such categories. Metaphors can also tie together more than one kind of sensory representation, as in "loud clothes."

Langacker (1987) defends an approach to *cognitive grammar* that takes metaphor and imagery as central to mental life, including language processing. He argues that sensory imagery plays a substantial role in conceptual structure; for example, the meaning of the word "trumpet" may be tied in part to an auditory image of the sound a trumpet makes. This approach to linguistics is controversial, but it suggests how language may depend on visual and other images as well as on words.

Psychological Plausibility

Many psychological experiments have supported the claim that visual imagery is part of human thinking. Cooper and Shepard (1973) measured how long it took students to decide whether a rotated letter was normal or a mirror image. Figure 6.4 shows versions of the letter "R." The first "R" is normal, but the second is a mirror image. The third and fourth "R"s can be discovered to be, respectively, mirror and normal images by mentally rotating them. If letters are relatively close to the normal position, like the "R"s in cases 5 and 6, then less time is needed to determine whether they are normal or mirror images than when they are relatively far from the normal position, like the "R"s in cases 3 and 4.

In addition to rotation experiments, scanning experiments have confirmed the mental imagery hypothesis by finding that people take more time to scan longer distances across images (Kosslyn 1980). Make a mental image of your country, and identify a city on the west coast or border, one in the interior of the country, and one on the east coast or border. For example, Americans should locate San Francisco, Chicago, and New York on their maps. If you are working with a visual image, then it should take longer for you to scan from the western city to the eastern city than it does to scan from the western city to the central city.

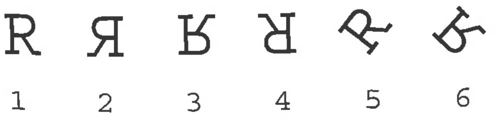


Figure 6.4
Mental rotation: the amount of time it takes to determine whether a letter is normal or a mirror image is directly proportional to how much it needs to be rotated to find an answer.

Finke, Pinker, and Farah (1989) performed experiments that show that people can assign novel interpretations to images that have been constructed out of parts or mentally transformed. In addition to the rotated "B"-into-heart example described above, they gave students instructions such as the following: Imagine the letter "Y." Put a small circle at the bottom of it. Add a horizontal line halfway up. Now rotate the figure 180 degrees. Most people see a stick person as the result of these instructions. The required transformations are shown in figure 6.5. People's frequent success in getting the right answer suggests that they are operating with visual representations. Even financial judgments may be affected by mental imagery (MacGregor et al. 2002).

Although most researchers in psychology are convinced by experiments like those just described that humans use visual imagery, some skeptics maintain that the same kind of verbal representations underlie all thought and that the experiences of imagery are illusory. Rotation, scanning, and other transformations can always be mimicked by nonimagistic computational procedures on lists of words. Within the last decade, however, neurological evidence has accumulated that provides further support for the imagery hypothesis.

Neurological Plausibility

Kosslyn (1994b) extensively reviews two kinds of evidence that parts of the brain used in visual perception are also involved in visual mental imagery. First, patients with brain damage that produces deficits in their perceptual

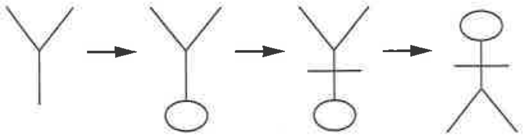


Figure 6.5
Sequence of transformations required to produce the stick person.

abilities sometimes have similar imagery deficits. For example, some patients unable to see one side of space during perception also are unable to see the same side of space during imagery. Damage to the occipital lobe impairs visual imagery. Second, measurements of brain activity have found that when people use visual mental imagery to perform tasks, brain areas used in visual perception become active. Imagery relies on regions of cortex that are spatially organized in ways that correspond to the structure of the retina, the networks of nerve cells that send impulses to the brain. The areas of the brain most immediately connected to the retina have a spatial organization that is structurally similar to that of the retina. Since these areas preserve some of the spatial structure of objects presented to the retina, their activation during imagery suggests that imagery involves picturelike representations, not just verbal descriptions. Kosslyn, Ganis, and Thompson (2001) review neurological studies of visual, auditory, and motor imagery.

Kosslyn describes the brain's processing of mental images in terms of computational mechanisms by which it satisfies multiple constraints in parallel. Chapter 7 describes how similar processes can be performed by artificial neural networks.

Practical Applicability

If mental imagery is useful in problem solving, education may profitably involve teaching people to use images more effectively. Larkin and Simon (1987) describe the conditions under which diagrams contribute to effective problem solving. Most psychological work on imagery, however, has been concerned with how people use images, not with educating them to use images better. Dehaene et al. (1999) report behavioral and brain-imaging experiments that suggest that mathematical intuition sometimes depends on visual and spatial representations; hence, mental images may be relevant to improving the teaching of mathematics.

Many strategies for improving memory rely on visual images. To remember something important, it helps to associate it with a vivid image. For example, to ensure that you will be able to recall the six kinds of mental representation discussed in this book, you might associate each of them with a mental picture of a different zoo animal that you think of as logical, rule-based, and so on.

Many kinds of design by architects, engineers, and product designers use visual representations such as sketches and blueprints. Mental imagery is presumably a part of these designers' creative mental processes, but there is little psychological evidence or computational understanding concerning the role of imagery in design. Kosslyn (1994a) presents a set of principles, based on empirical findings, for making visual displays that people can easily read and understand. Finke, Ward, and Smith (1992) discuss imagery's contribution to creative inventions.

Although artificial intelligence researchers have taken increasing interest in imagery and diagram-based systems, image-based expert systems are rare. Forbus, Nielson, and Faltings (1991) describe a system that does qualitative spatial reasoning about physical devices. Glasgow, Fortier, and Allen (1993) have used an array-based system for determining crystal and molecular structure.

Summary

Visual and other kinds of images play an important role in human thinking. Pictorial representations capture visual and spatial information in a much more usable form than lengthy verbal descriptions. Computational procedures well suited to visual representations include inspecting, finding, zooming, rotating, and transforming. Such operations can be very useful for generating plans and explanations in domains to which pictorial representations apply. The explanatory schema for visual representation is as follows:

Explanation target

Why do people have a particular kind of intelligent behavior?

Explanatory pattern

People have visual images of situations.

People have processes such as scanning and rotation that operate on those images.

The processes for constructing and manipulating images produce the intelligent behavior.

Imagery can aid learning, and some metaphorical aspects of language may have their roots in imagery. Psychological experiments suggest that visual procedures such as scanning and rotating employ imagery, and recent neurophysiological results confirm a close physical link between reasoning with mental imagery and perception.

Discussion Questions

1. Is introspection a reliable guide to our mental representations and procedures? Why is introspection alone not enough to show the importance of mental images?
2. Do you have sensory imagery? When do you most frequently use it?
3. What computations are potentially easier to achieve using imagistic representations?
4. In what kinds of problem solving are visual images useful? When can they become a hindrance?
5. How would a critic of mental imagery explain the psychological and neurological experiments supporting mental imagery?

Further Reading

Kosslyn 1994b and Kosslyn, Ganis, and Thompson 2001 provide a comprehensive review of recent psychological and neurological results. Finke 1989 surveys much experimental work on imagery. Glasgow 1993 reviews the debates about imagery from a computational perspective, with discussion by AI critics. Tye 1991 provides a philosophical examination. Langacker 1987 touches on the relevance of imagery to linguistics. Marr 1982 is a classic source on human and computer vision. For the latest in the imagery debate, see Pylyshyn 2002 and Kosslyn, Ganis, and Thompson 2003.

Web Sites

Diagrammatic reasoning: http://www.hcrc.ed.ac.uk/gal/Diagrams/

Imagination and mental imagery: http://www.calstatela.edu/faculty/nthomas/home.htm

Sports and mental imagery: http://www.vanderbilt.edu/AnS/psychology/health_psychology/mentalimagery.html

Stephen Kosslyn's home page: http://www.wjh.harvard.edu/~kwn/

Zenon Pylyshyn's home page: http://ruccs.rutgers.edu/faculty/pylyshyn.html

Notes

One of the main reasons that computational models of imagery have been relatively rare is that the programming tools currently available are much better suited for verbal representations than for visual ones. In addition to the array representations advocated by Glasgow and Papadias (1992), graph representations can be useful for capturing some aspects of visual representations (Wong, Lu, and Rioux 1989). Croft and Thagard (2002) use scene graphs and the Java 3-D programming language to model visual analogies.

7 Connections

Near the end of the nineteenth century, Santiago Ramón y Cajal discovered that the brain consists of discrete cells. These neurons signal each other through contacts at specialized points called synapses. Figure 7.1 shows a simplified picture of neurons connected by synapses. The human brain has about 100 billion neurons, many of which connect to thousands of other neurons, forming neural networks.

In the early days of computational models of thinking in the 1950s and 1960s, there was much interest in modeling how neural networks might contribute to thought. But this work waned in the 1970s, as the attention of researchers in artificial intelligence and psychology shifted almost entirely to rule-based and concept-based representations. In the 1980s, however, there was a dramatic rebound of computational modeling inspired by the neuronal structure of the brain (e.g., Hinton and Anderson 1981; Rumelhart and McClelland 1986). This research is often called connectionist, because it emphasizes the importance of connections among simple neuronlike structures, but is also sometimes discussed in terms of neural networks or parallel distributed processing (PDP). A wealth of connectionist models of mind and brain have been developed, but I will concentrate on two classes of models. The first class is concerned with *local* representations in which neuronlike structures are given an identifiable interpretation in terms of specifiable concepts or propositions. The second class is concerned with *distributed* representations in networks that learn how to represent concepts or propositions in more complex ways that distribute meaning over complexes of neuronlike structures.

Both local and distributed representations can be used to perform *parallel constraint satisfaction.* Many cognitive tasks can be understood computationally in terms of processing that simultaneously satisfies numerous

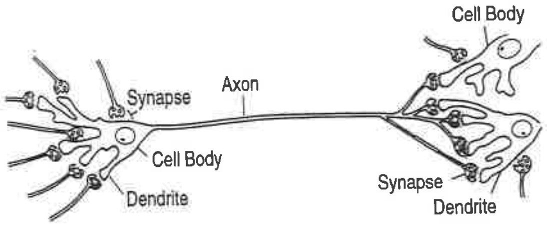


Figure 7.1
Neurons connected by synapses. The electrical signals flow into the dendrites and out through the axon. Adapted with permission from Rumelhart and McClelland 1986, vol. 2, p. 337.

constraints. As an initial example of a constraint satisfaction problem, consider the task faced by university administrators when they put together a new class schedule. Some of the constraints they face are inviolable: they cannot put two classes in the same room at the same time, and a student or professor cannot simultaneously be in two different classes. In contrast, many of the constraints are soft ones, involving preferences of professors and students concerning when and where their classes will take place. Coming up with a schedule that takes into account the various constraints imposed by classroom availability and the preferences of professors and students is a daunting task that is rarely accomplished in optimal fashion. Administrators typically take a previous term's schedule and adapt it as needed to handle new problems. But constraint satisfaction problems can be solved in a more general way if all the constraints are simultaneously taken into account.

Explicit models of parallel constraint satisfaction were first developed for computer vision. Marr and Poggio (1976) proposed what they called a "cooperative" algorithm for stereoscopic vision. Two eyes form slightly different images of the world: how does the brain match the two images and construct a coherent combined image? Marr and Poggio noticed that matching is governed by several constraints involving how points in one image can be put into correspondence with points in another. Creating a coherent image is then a matter of satisfying the constraints on matching points across the two images. To accomplish this task computationally, Marr and Poggio proposed using a parallel, interconnected network of processors in which the interconnections represented the constraints.

Similar networks were subsequently used by Feldman (1981) to model visual representations in memory and by McClelland and Rumelhart (1981) to model letter perception. Look back at the Necker cube presented in figure 6.1. Parallel constraint satisfaction provides a mechanism for resolving the ambiguity inherent in the Necker cube. Each of the two global interpretations can be defined in terms of a set of more elementary interpretations of the elements of the drawing. For example, under one interpretation the top-left corner in the drawing is the front-top-left corner of the cube, whereas under the other interpretation the same point is interpreted as the back-top-left corner. Furthermore, the possible local interpretations are highly interdependent, tending to either support or compete with each other in accord with the structural relations embodied in the canonical cube.

Human interpretations of the Necker cube can be modeled by a simple connectionist network that uses units to represent interpretations of the corners and links between units to represent compatibilities and incompatibilities between interpretations. In this network, parallel constraint satisfaction converges on one or the other of the two possible views, activating a subset of units that collectively represent a coherent interpretation, and deactivating the others. Research in the past decade has shown that parallel constraint satisfaction applies to many kinds of high-level cognition, not just to visual perception.

Representational Power

Connectionist networks constitute very simple representations, since they consist only of units and links. The units are analogous to neurons and have a degree of activation that corresponds roughly to the frequency with which neurons fire in order to send signals to other neurons. In local connectionist networks, the units have a specifiable interpretation such as particular concepts or propositions. The activation of a unit can be interpreted as a judgment about the applicability of a concept or the truth of a proposition. Links can be one-way, with activation flowing from one unit to another, or symmetric, with activation flowing back and forth between two units. Links are either excitatory, with one unit raising the activation of another, or inhibitory, with one unit suppressing the activation of another. Figure 7.2 gives a simple example of a local network that might be involved

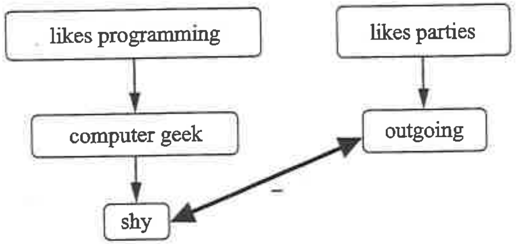


Figure 7.2
Simple local network with excitatory links (thin lines) and an inhibitory link (thick line with minus sign). Which of the excitatory links could plausibly be symmetric?

in making inferences about a fellow student. You meet Alice and learn that she likes programming, so you think she might be a computer geek and therefore shy. On the other hand, you learn that she likes parties, which suggests that she is outgoing. In forming a coherent impression of her, you have to decide whether she is actually shy or outgoing. The network in figure 7.2 uses a unit to represent each trait and has one-way excitatory links that make activation flow from the observed behaviors to the inferred traits. It also has a symmetric inhibitory link between shy and outgoing, reflecting the fact that it is hard to be both. The distributed networks described below include units that do not have such specific interpretations.

To understand the nature of distributed representations, we can use a visual analogy developed by Kosslyn and Koenig (1992, 20). Figure 7.3 shows an octopus network that accomplishes the task of communicating to seagulls the presence of fish near the bottom of the tidal pool. The octopi in the bottom row detect fish and signal to the octopi in the middle row by squeezing their tentacles, and the octopi in the middle row similarly signal to those in the top row, who in turn can throw up tentacles to inform the seagulls. This is a kind of *feedforward* network where information flows upward through the network. The bottom row of octopi can be thought of as an input layer, and the top row as an output layer, but what interpretation can be given to the octopi in the middle row? The information about how many fish there are is not encoded in any particular octopus, but rather is distributed over the whole network of octopi. Similarly, figure 7.4 depicts a feedforward neural network in which the hidden

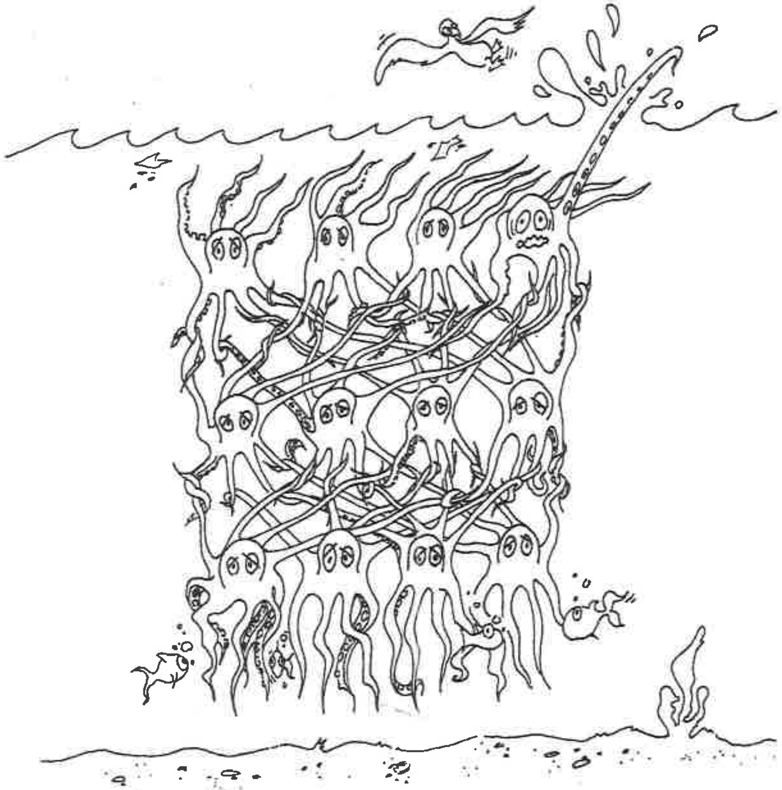


Figure 7.3
A visual analogy for a distributed processing network. Reprinted with the permission of The Free Press, an imprint of Simon and Schuster, from *Wet Mind: The New Cognitive Neuroscience*, by Stephen M. Kosslyn and Olivier Koenig. Copyright © 1992 by Stephen M. Kosslyn and Olivier Koenig.

(neither input nor output) units in the middle layer have no initial interpretation. They acquire an interpretation through adjustments in the weights that connect them to other units, by a learning process discussed below. In *recurrent* networks, activation from the output units feeds back into the input units.

Concepts can be viewed as distributed representations in networks. A network that is trained to respond accurately to stimuli can acquire concepts that apply to the stimuli. For example, if a network has as input units

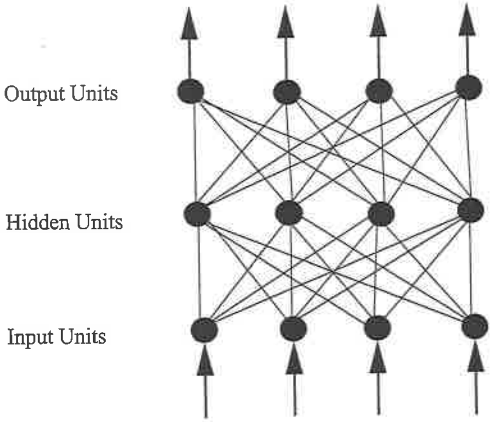


Figure 7.4
A feedforward computer model, with input, hidden, and output units.

features of animals, and output units that identify kinds of animals such as dog and cat, then the network can acquire the concept of a dog or a cat. The concept does not consist of any particular node; rather, it consists of a typical pattern of activation of units that occurs when a typical set of features is given as input. The notion of a concept as a pattern of activation of nodes in a distributed network is very different from the characterization of concepts given in chapter 4, but shares with it the claim that a concept is a prototype rather than a set of necessary and sufficient conditions.

Links between units suffice for representing simple associations such as that computer geeks are shy and shy people are not outgoing. But they lack the representational power to capture more complex kinds of rules, such as that anyone who likes a computer geek is also a computer geek. In the logical symbolism presented in chapter 2, this would be something like

$$(x)\ \{(\exists y)\ [\text{geek}\ (y)\ \&\ \text{likes}\ (x, y)] \rightarrow \text{geek}\ (x)\}.$$

In words: "For any x, if there is a y such that y is a geek and x likes y, then x is a geek." Relations such as "likes" and complex logical relations are difficult to represent in connectionist networks, although ingenious attempts are underway to increase their representational power beyond that of the simple local network in figure 7.2. One promising technique is to use *synchrony* to link units that represent associated elements: a unit or package of units that represents the x that does the liking can be made to fire with the same temporal pattern as the x that likes computers (Shastri and Ajjanagadde 1993; Shastri 1999; Hummel and Holyoak 1997). Another way of representing relational information is to use vectors, which are lists of numbers that can be understood as the firing rates of groups of neurons. For example, the vector (0.3, 0.4, 0.2) can be interpreted as the relative firing activity of three neurons. Vectors can be used to distinguish between agents (e.g., what does the liking) and objects (e.g., ones that are liked). Such vectors can be combined to represent highly complex relational information needed for analogical reasoning (Smolensky 1990; Eliasmith and Thagard 2001). See chapter 9 for more discussion of neuronal representations.

Neural networks provide powerful sensory representations that make possible many more tastes and aromas than we can typically express in words (Churchland 1995). The tongue has four types of taste sensors, for sweet, sour, salty, and bitter. Consider a system that has a unit corresponding to each of these sensors, with each unit capable of ten distinct levels of activation. Then the system can discriminate $10^4 = 10{,}000$ different tastes, each corresponding to a different pattern of activation.

Computational Power

Problem Solving

Neural networks provide powerful computational tools for performing parallel constraint satisfaction. Consider the problem in figure 7.2, where the task is to decide whether Alice is outgoing or shy. This problem has both positive constraints, such as between *likes parties* and *outgoing*, and negative constraints, such as between *outgoing* and *shy*.

Once the concepts and constraints are specified, implementing this kind of model in a parallel network is easy. First, concepts such as *outgoing* are represented by units. Second, positive internal constraints are represented by excitatory connections: if two concepts are related by a positive constraint, then the units representing the elements should be linked by an excitatory link. Third, negative internal constraints are represented by inhibitory connections: if two concepts are related by a negative constraint, then the units representing the elements should be linked by an inhibitory link. Fourth, an external constraint can be captured by linking units representing elements that satisfy the external constraint to a special unit that affects the units to which it is linked either positively (by virtue of excitatory links) or negatively (by virtue of inhibitory links). In the Alice example, the external constraints are that you know that she likes programming and likes parties, so there will be links between the special unit and the units representing these two elements.

The neural network computes by spreading activation between units that are linked to each other. A unit with an excitatory link to an active unit will gain activation from it, whereas a unit with an inhibitory link to an active unit will have its own activation decreased. Some units are activated as others are deactivated, with the result depending on the interconnections among the units. A problem solution consists of when a group of units, such as those in the Alice problem, is activated by the set containing *outgoing*, while correctively deactivating the set containing *shy*. In the network in figure 7.2, *outgoing* will win out over *shy* because outgoing is more directly connected to the external information that Alice likes parties.

Constraints can be satisfied in parallel by repeatedly passing activation among all the units, until after some number of cycles of activity all units have reached stable activation levels. This process is called *relaxation*, by analogy to physical processes that involve objects gradually achieving a stable shape or temperature. Achieving stability is called *settling*. Relaxing the network means adjusting the activation of all units based on the units to which they are connected until all units have stable high or low activations.

Planning Although decisions among competing plans are naturally understood in terms of parallel constraint satisfaction, constructing plans is usually a more sequential process understood in terms of rules or analogies. Your plan to graduate can be expressed in terms of a set of rules concerning what sequence of courses will give you enough courses of the required kinds. But connectionist networks can implement simple kinds of rule-based systems. Touretzky and Hinton (1988) constructed a rule-based system that uses distributed representations. It treats the process of matching the *IF* part of a rule as a kind of parallel constraint satisfaction. However, the resulting system can match only clauses with simple predicates, not relations. Nelson, Thagard, and Hardy (1994) use local representations to implement rule matching and analogy application as parallel constraint satisfaction. The resulting system models plan construction, such as how Juliet in Shakespeare's play planned to meet Romeo. Thus, connectionist systems can be indirectly relevant to modeling solutions of planning problems.

Decision We can understand the process of making a decision in terms of parallel constraint satisfaction (Thagard and Millgram 1995; see also Mannes and Kintsch 1991). The elements of a decision are various actions and goals. The positive internal constraints come from facilitation relations: if an action facilitates a goal, then the action and goal tend to go together. The negative internal constraints come from incompatibility relations, when two actions or goals cannot be performed or satisfied together, as when a student cannot take two courses at the same time. The external constraint on decision making comes from goal priority: some goals are inherently desirable, providing a positive constraint. Once the elements and constraints have been specified for a particular decision problem, a constraint network can be formed such as that seen in figure 7.5.

Suppose you are facing the difficult problem of deciding what to do after graduation. Perhaps your options include going to graduate school or taking an entry-level position with a large corporation. The constraints you face are first that you cannot do both and moreover that the different options fit better with different goals that you have. Immediate employment may solve your current financial problems, but may not necessarily provide an interesting long-term career. Moreover, perhaps there are aspects of your field that you want to learn more about. On the other hand, you might be tired of taking classes. Figure 7.5 shows a simple network that captures part of what is involved in the decision. Units represent the various options and goals, and pluses and minuses indicate the excitatory and inhibitory links that embody the fundamental constraints. If a unit settles with high activation, this is interpreted as acceptance of the goal or action that it represents, whereas deactivation represents rejection. The unit representing graduate school has stronger excitatory links and therefore will get more activation than the unit representing taking a job, which will be deactivated because of the inhibitory link with the unit for graduate school.

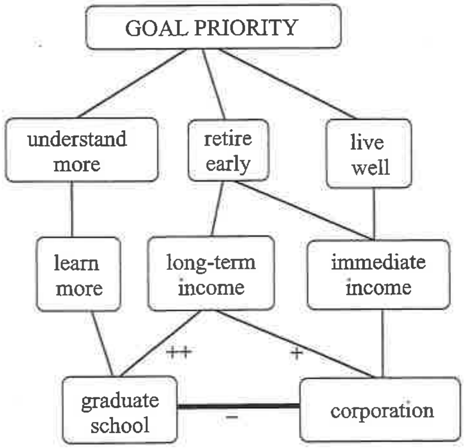


Figure 7.5
A constraint network for decision making. Boxes represent units, thin lines represent positive constraints based on facilitation (symmetric excitatory links), and the thin line with a minus represents a negative constraint (inhibitory link). The "goal priority" special unit pumps activation to the other nodes that have to compete for it.

Analogy can also be useful in decision making, since a past case where something like A helped to bring about something like B may help one to see that A facilitates B. But reasoning with analogies may itself depend on parallel constraint satisfaction. Chapter 5 described Holyoak's and my view that retrieving and mapping analogs involves the constraints of similarity, structure, and purpose (Holyoak and Thagard 1995). The computational models we have implemented perform parallel satisfaction of these constraints using mechanisms similar to the ones just described for decision making.

Explanation Churchland (1989) has contended that explanation should be understood as activation of prototypes encoded in distributed networks. Understanding why a particular bird has a long neck can come via activation of a set of nodes representing *swan*, which include the prototypical expectation that swans have long necks. On this view, inference to the best explanation is just activation of the most appropriate prototype.

Using local networks, inference to the best explanation has been modeled via a theory of explanatory coherence (Thagard 1989, 1992, 2000). Suppose you are expecting to meet your friend Fred at the cafeteria, but Fred does not show up. Your knowledge of Fred and your general knowledge about other students may suggest various hypotheses that could explain why Fred does not show up, but you would still have to decide which hypothesis is most plausible. Perhaps Fred decided he had to study, or maybe he went dancing with someone. An extra piece of evidence that Fred was spotted in the library would clearly support one hypothesis over the other. Figure 7.6 shows a network that captures some of the relevant information as used in the program ECHO that I wrote to model explanatory coherence. Units representing pieces of evidence are linked to a special evidence unit that activates them, and activation spreads out to other units. There is an inhibitory link connecting the units representing the two competing hypotheses that Fred is in the library and that he went dancing. Choice of the best explanation can involve not only the evidence for particular hypotheses, but also explanations of why those hypotheses might be true. For example, Fred's motive for studying is that he wants high grades; alternatively, the reason he went dancing might be that he likes to party. Settling the network will provide a coherent interpretation of his behavior. In the network in figure 7.6, the network will settle with the unit for "Fred is studying" activated because it has more sources of activation than its competitor, the unit for "Fred went dancing."

Learning

Given the simple structure of connectionist networks, there are two basic ways in which learning can take place: add new units, or change the weights on the links between units. Work to date has concentrated on the second kind of learning. A biologically plausible kind of weight learning was proposed by Hebb (1949). He speculated that when two brain cells or systems are active at the same time, they should become associated with each other. This kind of learning has been observed in real neurons and has been modeled computationally in various ways. The idea is that if unit (neuron) A and unit B are both active at the same time, then the weight on the link between them should increase. For example, in a local network

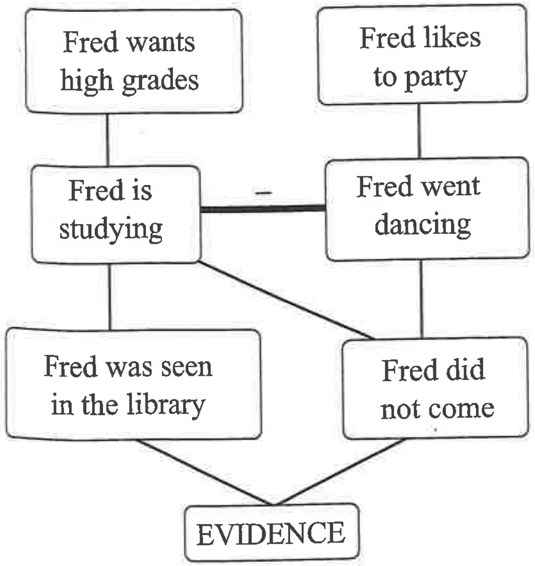


Figure 7.6
Network for picking the best explanation of why Fred did not show up. The thin lines are symmetric excitatory links and the thick line marked with a minus is a symmetric inhibitory link.

that has units representing both dancing and partying, if these units are frequently active at the same time, then the link between them will become stronger and stronger, implementing an association between dancing and partying. This kind of learning is unsupervised in that it does not require any teacher to tell the network when it has right or wrong answers.

The most common kind of learning in feedforward networks with distributed representations uses a technique called *backpropagation*. Figure 7.7 shows a simple network with input, hidden, and output units that is supposed to learn about social stereotypes on campus. After training, the network should be able to classify students: given a set of features activated in the input layer, it should activate an appropriate stereotype at the output layer. For example, a student who plays sports and parties (input layer) could be identified as a jock (output layer). Backpropagation can be used to train the network by adjusting the weights that connect the different

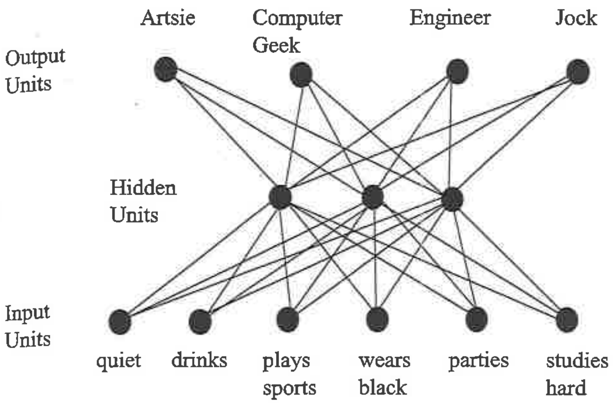


Figure 7.7
A network that can be trained to classify students.

units, through the following steps (see Towell and Shavlik 1994; for full details, see Rumelhart and McClelland 1986):

1. Assign weights randomly to the links between units.
2. Activate input units based on features of what you want the network to learn about.
3. Spread activation forward through the network to the hidden units and then to the output units.
4. Determine errors by calculating the difference between the computed activation of the output units and the desired activation of the output units. For example, if activation of *quiet* and *studies hard* activated *jock*, this result would be an error.
5. Propagate errors backward down the links, changing the weights in such a way that the errors will be reduced.
6. Eventually, after many examples have been presented to the network, it will correctly classify different kinds of students.

Backpropagation models have had many successful applications, in both psychology and in engineering. They do more than simply identify rules such as *IF someone plays sports THEN he or she is a jock*. Networks trained by backpropagation can identify statistical associations between input and output features that are more subtle than rules. Nevertheless, backpropagation has a number of drawbacks as a model of human learning. First, it requires a supervisor to say whether an error has been made. Much learning—for example, of language—seems to occur without much explicit supervision. Neural network models of unsupervised learning are discussed in Hinton and Sejnowski (1999). Second, backpropagation tends to be slow, requiring many hundreds or thousands of examples to train a simple network. For some kinds of human learning large numbers of trials seem appropriate, but people can also sometimes learn from very few examples. McClelland, NcNaughton, and O'Reilly (1995) advocate complementary learning systems that use both a slow-learning component for semantics as well as a fast-learning one for object names and other information.

Language

Early connectionist models of language involved visual and auditory perception. McClelland and Rumelhart (1981) showed how word recognition can be understood as a parallel constraint satisfaction problem. Suppose you spilled coffee on this page so that some of the letters were partly covered. You would probably still be able to figure out what many of the words were, by using visible letters and the overall context. For example, in figure 7.8 it is possible to determine the ambiguous middle letter in each word, using both the presented information about the shape of the letter and the overall context given by the word that the letter appears in. Interconnected units can represent hypotheses about what letters are present and about what words are present, and relaxing the network can pick the best overall interpretation. McClelland and Elman (1986) developed a similar model of speech perception.

Just as connectionist networks can be used to disambiguate letters and sounds, they can be used also to disambiguate word meanings. Kintsch (1988, 1998) proposed a "construction-integration" model of discourse comprehension that could explain, for example, how the word "bank" is sometimes taken to mean a financial institution and at other times taken to mean the edge of a river. Unlike what happens under the view of concepts described in chapter 4, meaning is not built into a concept but must instead be created in particular contexts by interacting elements. Figure 7.9 shows part of a network that might be useful for determining the appropriate meaning of "bank" in a particular context. Which interpretation gets activated depends on how input information will affect the various units and links.

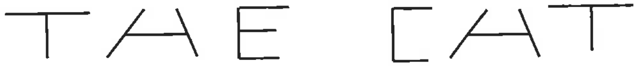


Figure 7.8
Context makes possible identification of identical structures as different letters.

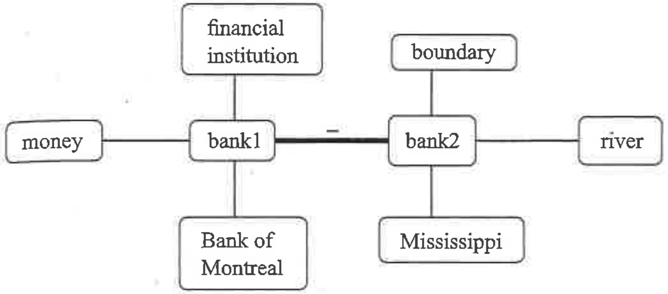


Figure 7.9
Meaning of "bank" is determined by activation flow in the network. Thin lines are symmetric excitatory links. Thick lines are symmetric inhibitory links.

Rumelhart and McClelland (1986) developed a parallel distributed processing model of how children learn to form the past tense of English verbs without forming explicit rules. One explanation of why young children erroneously use past tenses such as "goed" and "hitted" is that they have formed a rule that produces past tenses by simply adding "ed." Errors arise when this rule is applied too generally to include irregular verbs. But Rumelhart and McClelland showed how a connectionist network can be trained to reproduce the children's error using distributed representations rather than rules. In response, Pinker and Prince (1988) argued that the connectionist model is psychologically implausible in that it forms past tenses quite differently from how children do. MacWhinney and Leinbach (1991) replied with a new connectionist model designed to overcome these objections, and Ling and Marinov (1993) countered with a nonconnectionist model that they claimed is at least as psychologically realistic. The debate continues with Pinker and Ullman (2002) advocating a "words and rules" theory of language processing. McClelland and Patterson (2002) defend the connectionist approach.

Psychological Plausibility

Connectionist models have furnished explanations of many psychological phenomena. McClelland and Rumelhart's (1981) model of word perception described above has explained the results of several experiments. For example, Rumelhart and McClelland (1982) described psychological experiments that confirmed their model's predictions concerning how the duration of context letters affects the perceptibility of a word. McClelland and Elman (1986) described various speech perception phenomena such as temporal effects that are explained by their model. Similarly, Kintsch's (1988) model of discourse comprehension has been confirmed by experiments in which students verified sentences of various types (Kintsch et al. 1990).

The local connectionist models of analogical mapping and retrieval not only have been used to simulate the results of previous psychological experiments, but also have suggested new ones (Holyoak and Thagard 1995; Spellman and Holyoak 1993; Wharton et al. 1994). For example, Spellman and Holyoak (1993) were able to show that the purpose of an analogy has an effect on analogical mapping in a way that Holyoak's and my computer models simulate. Similarly, to test my connectionist model of how explanatory hypotheses are evaluated, Read and Marcus-Newhall (1993) and Schank and Ranney (1991, 1992) created experiments that compared judgments of human subjects favorably with those generated by the program ECHO. Ziva Kunda and I used a simple local connectionist model to account for a dozen experimental results concerning how people form impressions of other people (Kunda and Thagard 1996).

Backpropagation techniques have simulated many psychological processes. For example, Seidenberg and McClelland (1989) used backpropagation to model visual word recognition in a way that simulates many aspects of human performance, including how words vary in processing difficulty, how novel items are pronounced, and how people make the transition from beginning to skilled reading. St. John (1992) used backpropagation to produce distributed representations that simulate many aspects of discourse comprehension. Connectionist learning mechanisms are now used to explain many aspects of human development, such as why children are quick to learn some things but slow to learn others (Bates and Elman 2002).

Neurological Plausibility

How neurologically plausible are local connectionist networks? The artificial networks in this chapter are similar to brain structure in that they have simple elements that excite and inhibit each other. But real neural networks are much more complicated, with billions of neurons and trillions of connections. Moreover, real neurons are much more complex than the units in artificial networks, which merely pass activation to each other. Neurons have dozens of neurotransmitters that provide chemical links between them, so the brain must be considered in chemical as well as electrical terms. Real neurons undergo changes in synaptic and nonsynaptic properties that go beyond what is modeled in artificial neural networks. See chapter 9 for discussion of neurons that are much more like those found in the brain.

In local representations, each unit has a specifiable conceptual or propositional interpretation, but each neuron in the brain does not have such a local interpretation. At best, we can think of each artificial unit as representing a *neuronal group*, a complex of neurons that work together to play a processing role. Thinking of units as like neuronal groups rather than like neurons also overcomes another difference between units and neurons: many local networks use symmetric links between units, whereas synapses connecting neurons are one-way. But neuronal groups often have neural pathways that allow them to influence each other. Unlike units in artificial neural networks, a real neuron has excitatory links to other neurons or inhibitory links to other neurons, but not a mixture. The brain clearly distributes its representations over far more neurons than are found in artificial neural networks, local or distributed.

Hebbian learning that strengthens synapses between similarly active neurons has been observed in the brain, which also experiences various other kinds of learning by synapse adjustment (Churchland and Sejnowski 1992, chap. 5). However, backpropagation learning does not correspond to any process that scientists have observed in the brain. Actual neural networks do have the feedforward character of backpropagation networks, but there is no known neurological mechanism by which the same pathways that feed activation forward can also be used to propagate error correction backward. O'Reilly and Munakata (2000, 162) describe an algorithm that is an approximation to backpropagation but is more biologically plausible.

Most connectionist models are thus only a very rough approximation to the behavior of real neurons. Nevertheless, the analogy between the brain and the computational mind has so far been very fruitful, and computer models that are more authentically brainlike are under development. Chapter 9 describes computational models that are more neurologically realistic than the ones presented in this chapter.

Practical Applicability

Connectionist models of learning and performance have had some interesting educational applications. Adams (1990) provides a connectionist-style description of the various kinds of knowledge required for reading. Figure 7.10 shows the interrelations among orthography, word meanings, and the broader context in which a word occurs. To read a piece of text, you need to process letters into words and simultaneously take into account meaning and context. In the terms of this chapter, reading is a kind of parallel constraint satisfaction where the constraints simultaneously involve spelling and meaning and context. Any narrow approach to teaching reading that ignores some of these constraints—for example, by neglecting phonics or by neglecting meaning and context—will make learning to read more difficult.

Design is naturally thought of in terms of parallel constraint satisfaction. For example, an architect's design for a building must take into account numerous constraints such as cost, the intended use of the building, its surroundings, and aesthetic considerations. Backpropagation techniques have been used to assist engineers in predicting the stresses and strains of materials needed for buildings (Allen 1992).

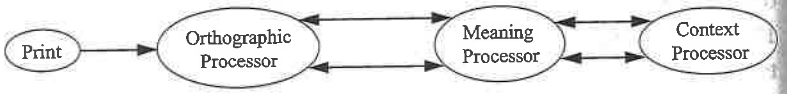


Figure 7.10
Multiple processors required for reading (Adams 1990, 138). See also Seidenberg and McClelland 1989.

Connectionist models are widely used in intelligent systems. The backpropagation algorithm has had many engineering applications—for example, in training networks to recognize bombs, underwater objects, and handwriting. One bank trained an artificial neural network to identify which of its customers were likely to default on loans. Other networks have been trained to interpret the results of medical tests and predict the occurrence of disease. Widrow, Rumelhart, and Lehr (1994) survey applications of neural networks in industry.

Summary

Connectionist networks consisting of simple nodes and links are very useful for understanding psychological processes that involve parallel constraint satisfaction. Such processes include aspects of vision, decision making, explanation selection, and meaning making in language comprehension. Connectionist models can simulate learning by methods that include Hebbian learning and backpropagation. The explanatory schema for the connectionist approach is as follows:

Explanation target

Why do people have a particular kind of intelligent behavior?

Explanatory pattern

People have representations that involve simple processing units linked to each other by excitatory and inhibitory connections.

People have processes that spread activation between the units via their connections, as well as processes for modifying the connections.

Applying spreading activation and learning to the units produces the behavior.

Simulations of various psychological experiments have shown the psychological relevance of the connectionist models, which are, however, only rough approximations to actual neural networks.

Discussion Questions

1. What is the difference between a local and a distributed representation?
2. How do units in artificial neural networks differ from natural neurons?

3. How do connectionist explanations of psychological phenomena differ from rule-based explanations?

4. What psychological phenomena are most naturally explained in connectionist terms?

5. What psychological phenomena are most difficult for connectionists to explain?

Further Reading

Introductions to neural network modeling include Bechtel and Abrahamsen 2002, Churchland and Sejnowski 1992, O'Reilly and Munakata 2000, and Rumelhart and McClelland 1986. McClelland and Rumelhart 1989 provides detailed instructions for doing your own modeling. Anderson and Rosenfeld 1988 includes some classic papers on neural networks, and Anderson and Rosenfeld 1998 contains interviews with many pioneering researchers in the field. Elman et al. 1996 applies connectionist ideas to the problem of innateness. Tesar and Smolensky 2000 discusses language acquisition from the perspective of a theory that grew out of connectionism.

Web Sites

Jeff Elman's home page: http://crl.ucsd.edu/~elman/

Jay McClelland's home page: http://www.cnbc.cmu.edu/~jlm/

Software for neural network modeling: http://www.cnbc.cmu.edu/Resources/PDP++//PDP++.html

Notes

The kind of spreading activation between concepts as discussed in chapter 4 is narrower than the kind discussed in this chapter, which includes inhibitory as well as excitatory mechanisms, and includes the activation of hidden units that do not represent whole concepts.

To compute activations of the units in a connectionist network, each unit is given a starting activation and repeated cycles of updating begin. There are many ways this can be done. In one technique, on each cycle the activation of a unit *j*, a_j, is updated according to the following equation:

$$
\begin{aligned}
a_j(t+1) = {} & a_j(t)(1-d) + \\
& net_j(max - a_j(t)) \quad \text{if } net_j > 0 \\
& net_j(a_j(t) - min) \quad \text{otherwise.}
\end{aligned}
$$

Here *d* is a decay parameter (that decrements each unit at every cycle, *min* is the minimum activation (–1), *max* is the maximum activation (1). Based on the weight w_{ij} between each unit *i* and *j*, we can calculate net_j, the net input to a unit, by the equation

$$net_j = \Sigma_i w_{ij} a_i(t).$$

8 Review and Evaluation

Cognitive science is about the same age as rock and roll; both emerged from diverse sources in the mid-1950s. Like rock music, cognitive science has changed in many ways through the development of new ideas and techniques. This chapter briefly summarizes the achievements of cognitive science, comparing and evaluating the representational and computational power of the six approaches described in chapters 2–7. It concludes by sketching a series of important challenges for CRUM, the Computational-Representational Understanding of Mind.

The Achievements of Cognitive Science

Scientific understanding of problem solving, learning, and language is enormously more sophisticated now than it was fifty years ago when behaviorism reigned. We know how to design complex systems that make logical inferences. Rule- and concept-based systems have successfully modeled various aspects of problem solving and language use. In the past couple of decades, analogical thinking has been increasingly understood through a combination of psychological experiments and computational modeling. Imagery has been transformed from a topic at the fringes of scientific investigation to a subject of highly sophisticated psychological, neurological, and computational research. Connectionist models of learning and parallel constraint satisfaction have furnished explanations of numerous psychological phenomena.

One accomplishment that has eluded cognitive science is a unified theory that explains the full range of psychological phenomena, in the way that evolutionary and genetic theory unify biological phenomena, and relativity and quantum theory unify physical theory. Different

Table 8.1
Review of theoretical applications of computational approaches.

	Representation	Problem solving	Learning	Language
Logic	Propositions Operators Predicates Quantifiers	Deduction Probability	Generalization Abduction	Logical form
Rules	IF-THEN	Search Forward chaining Backward chaining	Chunking Generalization Abduction	Grammar Pronunciation Spelling
Concepts	Frames with slots Schemas Scripts	Matching Inheritance Spreading activation	Abstraction from examples Conceptual combination	Lexicon Semantics
Analogies	Target and source Causal relations	Retrieval Matching Adaptation	Storage Schema formation	Metaphor
Images	Visual, motor, etc.	Matching, manipulating	Imaginary practice	Image schemas
Connections	Units and links	Parallel constraint satisfaction	Backpropagation weight adjustment	Disambiguation Pronunciation

cognitive scientists argue that the mind is a logical system, a rule-based system, a concept-based system, an analogy-based system, an imagery-based system, and a connectionist system. The perspective of this book is that the best current answer to the final exam question "What kind of system is the mind?" is "All of the above." The mind is an extraordinarily complex system, supporting a very diverse range of kinds of thinking.

The different approaches to CRUM that were described in chapters 2–7 tend to capture different aspects of mind. Table 8.1 summarizes the different approaches and their theoretical applications. At this early stage of cognitive science research, theoretical diversity is a desirable feature rather than a flaw. Of course, we can hope that a Newton, Darwin, or Einstein of cognitive science will emerge to provide a simple, unified theory that incorporates all the insights to date. But progress can be made in understanding mind without such an overarching theory, which the complexity and diversity of mind might make unattainable. One premise of cognitive science is that progress will require more than the isolated efforts of researchers in particular disciplines. Integrated, cross-disciplinary effort will continue to be essential in understanding the nature of mind.

Table 8.2
Practical applications of cognitive science.

	Education	Design	Systems
Logic	Critical thinking	Codes	Logic programming
Rules	Arithmetic, skill acquisition	Computer-human interaction	Most expert systems
Concepts	Problem schemas	Building specifications	CYC, frame-based expert systems
Analogies	Problem solving	Case-based design	Case-based expert systems
Images	Visual problem solving	Diagrams	A few expert systems
Connections	Reading	Constraint satisfaction	Trained expert systems

Cognitive science has also had substantial applications to education, design, and intelligent systems. Different versions of CRUM have illuminated different aspects of applied thinking. We saw, for example, that rule-based and analogical models are useful in understanding how students solve problems, and connectionist parallel constraint satisfaction models have important implications for teaching reading. Design requires a diversity of cognitive processes, from deductive inference to imagery. Intelligent systems that mimic human abilities have drawn on a variety of kinds of representations and processes, especially rule-based, analogical (case-based), and connectionist (backpropagation) systems. Table 8.2 summarizes how the different approaches have been practically applied.

Comparative Evaluation

For a deeper review of the six different approaches to representation and computation, we can evaluate their comparative advantages and disadvantages, continuing to use the criteria of representational power, computational power, psychological plausibility, neurological plausibility, and practical applicability. This comparison supports the contention that no single approach currently deserves to be seen as the theoretical basis for all of cognitive science.

Representational Power

We saw that formal logic has considerable representational power, generating complex propositions with operators such as "not" and "or" and quantifiers such as "all" and "some." Computer models that restrict themselves to rules, concepts, analogies, images, or connections have difficulty representing intricate propositions such as "No students' supervisors are responsible for some of their students' problems or worries." Even so, formal logic does not capture all the subtleties of natural language, so we have to conclude that no current computational model has the representational power to capture all of human thought.

Connectionist models have an advantage over verbal representations in that they have more flexibility in capturing a broader range of sensory experience. Patterns of activation of units can represent many tastes and smells to which verbal representations only approximate. On the other hand, connectionism has struggled with the challenge of figuring out how simple neuronlike units can represent complex relations such as those naturally included in computational models based on logic, rules, or analogies. Since the brain with its billions of neurons somehow manages to produce language, we know that a system based on interacting units can produce complex inferences, but discovering how will require connectionist models with substantially more representational power than those now available.

Computational systems that employ rules abandon the expressiveness of formal logic for a simplified format of IF-THEN rules that have computational advantages. Like propositions in formal logic, rules are concise and independent representations. In contrast, concepts, analogies, and images all bundle information together into organized structures. A concept collects a package of information about a kind of thing, and an analog collects a package of information about a situation. Images provide their own special kind of packaging since they are intimately connected with sensory functions such as vision. A visual image vividly ties together interconnected information that can be difficult to represent verbally.

In sum, a unified theory of mental representation needs to postulate structures that among them have (1) the sensory richness of images and connections, (2) the organizing capabilities of concepts, analogs, and images, and (3) the verbal expressiveness of rules and propositions in formal logic.

Computational Power

In developing a computational model, we need to be concerned with speed and flexibility as well as abstract computational potential. There are many ways to perform computations, but for cognitive science we need computational techniques that have the speed and flexibility necessary for both psychological plausibility and practical applicability. Inference viewed as logical deduction can be elegant, but rule-based systems that emphasize heuristic search have exhibited superior performance in many domains. The effectiveness of rule-based systems has led some theorists such as Newell (1990) to advocate a unified theory of cognition based on rules. But other computational mechanisms include matching of whole structures in applying concepts, analogies, and images. Concept-based and connectionist systems implement different kinds of spreading activation. Although much human problem solving can be construed as heuristic search in a rule-based system, there are many problem solutions that are better described in terms of processes like schema application, analogical mapping, and parallel constraint satisfaction.

Similarly, human learning is not restricted to a single mechanism such as rule-based chunking. A comprehensive theory will have to account for learning of rules and concepts from examples and from combinations of other rules and concepts. It should encompass both quick, one-shot learning such as when people abductively form new hypotheses, and slow, multiple-trial learning such as when children learn to balance. Rule-based chunking and connectionist weight adjustment are both powerful learning mechanisms, but neither captures the full range of human learning capabilities.

Similarly, cognitive science still lacks a comprehensive theory of language learning and use, although different approaches have shed considerable light on different aspects of language. Some aspects of grammar and pronunciation, for example, are plausibly described in terms of rules, but rule-based approaches have helped little with understanding the nature of the lexicon or the role of metaphor in language production and comprehension. Language thus seems to depend on concepts, analogies, and images as well as on rules. Perhaps connectionism will eventually provide a neurally inspired way of saying how all these aspects are exhibited by a single system. But no comprehensive connectionist theory of language has emerged, even though connectionist models of learning and parallel constraint satisfaction have been very successful in some linguistic applications such as word sense disambiguation.

Psychological Plausibility

Each of the six approaches to representation and computation has inspired psychological experiments as well as computational models. These experiments have addressed numerous controversial issues that continue to inspire lively debate. Are syllogistic and other kinds of logical inference done by applying logical rules, or by some more concrete method such as mental models? Is the process by which people learn to form the past tense of English verbs best described in connectionist terms or in terms of rules? Decades of experimental psychology have identified many phenomena that a general theory of mind will have to explain, but the situation so far is that different experimental results fit best with different representational theories. Rule-based models apply well to some cognitive tasks such as playing tic-tac-toe, but do not tell us much about other cases of problem solving where analogies are more prominent. Experiments support the importance of images in human thinking, but many phenomena do not seem to involve images. Although the connectionist simulations described in chapter 7 are successful in accounting for a diverse range of psychological phenomena, it would be premature to suppose that all other kinds of models are unnecessary. The connectionist models apply well to cognitive tasks that are naturally understood in terms of incremental learning and parallel constraint satisfaction. But the generation of units and constraints may require rule-based and other mechanisms that connectionist models have not yet addressed.

It would be wonderful to have a unified theory of cognition that could account for all psychological phenomena observed so far. But progress can also be made locally, applying particular theories of representation and computation to particular psychological phenomena. Cognitive science has made substantial progress in developing rich computational models of many kinds of human performance observed in psychological experiments. Discovering how the various kinds of representation and thinking fit together will undoubtedly require more experiments as well as more integrated models of the sort discussed in chapter 14.

Neurological Plausibility

When the first edition of this book appeared in 1996, there was considerable neurological evidence linking mental imagery with the visual system in the brain, but a lack of neurological evidence for logic, rules, concepts, and analogies. Thanks to new scanning techniques for observing the operations of the brain, cognitive neuroscience has been the fastest developing part of cognitive science. Chapters 2–5 of the current edition cite some relevant neurological studies. Connectionist models gain some neurological plausibility from the analogy between artificial neural networks and the brain, although current connectionist ideas are only rough approximations to how the brain works. Chapter 9 describes computational models that are more neurologically realistic.

Practical Applicability

Constructing a unified cognitive theory requires reconciling the conflicting claims of various cognitive scientists who hold that the mind is fundamentally a rule-based system, or a connectionist system, and so on. But accomplishing practical goals of improving education, design, and intelligent systems can proceed in a more piecemeal fashion, selectively applying insights from different approaches to cognitive science wherever they appear relevant.

Potentially, cognitive science is to education what biology is to medicine: a theoretical basis for practical remedies. Conceptions of the mind as using rules, concepts (schemas), and analogies have already contributed to understanding how people solve problems. Images are also relevant to problem solving as is evident in the usefulness of diagrams in many domains. Connectionist ideas are just starting to have an impact on educational theory and practice, and conceiving of processes such as reading in terms of parallel constraint satisfaction suggests ways of improving teaching.

To date, understanding the process of design has been most furthered by attending to the roles of rules, concepts, analogies, and images in creative design. Most expert systems that have had industrial applications have been rule-based systems, but case-based (analogical) and connectionist systems are proving increasingly useful. A manager hoping to develop an intelligent system should look carefully at the nature of the task to be accomplished and the knowledge available, critically considering what kinds of representation and computation are most appropriate.

Some advocates of particular approaches to cognitive science boldly assert that the mind is a rule-based system, or that the mind is a connectionist system, and so on. The fact that all current accounts of representation and computation have disadvantages as well as advantages suggests the need for combinations and integrations of the various approaches (see chapter 14). Some critics of CRUM have argued, however, that all of these computational approaches are inherently limited in what they can tell us about the mind.

Challenges for Cognitive Science

Review of the major approaches taken by advocates of CRUM shows that it explains much about the nature of human problem solving, learning, and language. Although CRUM has had considerable success in illuminating the nature of mind, there remain skeptics who believe that it is fundamentally misguided and neglects crucial aspects of thinking—for example, consciousness and emotional experience. Chapters 9–13 discuss seven important challenges for CRUM:

1 *The brain challenge* CRUM ignores crucial facts about how thinking is performed by the brain.

2 *The emotion challenge* CRUM neglects the important role of emotions in human thinking.

3 *The consciousness challenge* CRUM ignores the importance of consciousness in human thinking.

4 *The body challenge* CRUM neglects the contribution of the body to human thought and action.

5 *The world challenge* CRUM disregards the significant role of physical environments in human thinking.

6 *The dynamic systems challenge* The mind is a dynamic system, not a computational system.

7 *The social challenge* Human thought is inherently social in ways that CRUM ignores.

These challenges pose serious problems for CRUM and for the whole enterprise of cognitive science. There are four possible responses to them:

1. *Deny* the claims that underlie the challenge.
2. *Expand CRUM* to enable it to deal with the problems posed by the challenge, adding new computational and representational ideas.
3. *Supplement CRUM* with noncomputational, nonrepresentational considerations that together with CRUM can meet the challenge.
4. *Abandon CRUM.*

I will argue that none of the challenges provides reason to abandon CRUM. Several of them show, however, that CRUM needs to be expanded and supplemented, particularly in ways that integrate it with biological and social factors. Supplementing is different from expanding in that it requires introducing concepts and hypotheses that go beyond the computational-representational explanation pattern. Chapters 9–14 describe numerous ways in which cognitive science is currently being expanded to deal with gaps in older versions of CRUM.

The Mind–Body Problem

Because the challenges discussed in the following chapters raise important general questions about the nature of mind and body, it will be helpful first to outline the major philosophical views about how mind and body are related. The commonsense view of persons is that they consist of two components: a body and a mind. This view is called *dualism*, since it assumes that each of us consists of two fundamentally different substances, one physical and the other mental or spiritual. Anyone whose religious views imply that a person survives after death is a dualist, since the mind can survive the body's demise only if it is something nonphysical. Although dualism is probably the most widely held view of mind, it is philosophically problematic. What evidence do we have that there is mind independent of body? If mind and body are two different substances, how do they interact? Dualism makes mind a fundamentally mysterious entity beyond scientific investigation.

In contrast to dualism, *materialism* claims that mind is not a different kind of substance from the physical matter that constitutes the body. Philosophers have defended several versions of materialism. *Reductive materialism* claims that every mental state such as being conscious of the smell of donuts is a physical state of the brain. Thus, the mental can be reduced to the physical. More radically, *eliminative materialism* claims that we should not try to identify all the aspects of our mental experience with brain events, since our commonsense views of the mind may be fundamentally wrong. Instead, as neuroscience develops, we can hope to acquire a much richer theory of mind that may replace and eliminate commonsense notions such as consciousness and belief.

Both reductive and eliminative materialism assume that understanding the mind depends fundamentally on understanding the brain. However, the computational approach to mind has frequently been associated with a different view called *functionalism*, according to which mental states are not necessarily brain states, but rather are physical states that are related to each other through causal relations that can hold among various kinds of matter. For example, an intelligent robot might be viewed as having mental states even though its thinking depends on silicon chips rather than on biological neurons. Similarly, we might encounter intelligent aliens from other planets whose mental abilities depend on very different biological structures than human brains.

These four views—dualism, reductive materialism, eliminative materialism, and functionalism—have been the favorites in recent philosophy of mind. Another view, idealism, was popular in the nineteenth century. It holds that everything in the universe is mental and nothing is material.

Summary

The Computational-Representational Understanding of Mind has contributed to much theoretical understanding and practical application. But no single approach has emerged as the clearly most powerful explanation of human cognitive capacities. Different approaches have different representational and computational advantages and disadvantages. Psychological plausibility is shared among various approaches that have successfully modeled different kinds of thinking. But CRUM faces challenges that charge it with neglecting important aspects of mind.

Discussion Questions

1. What are the most impressive achievements of cognitive science? In what directions does it still have the furthest to go?
2. What other challenges does CRUM need to face?

3. What are the impediments to a unified theory of the mind? Will we ever have one? Would we want one?

Notes

My preferred version of materialism is close to what Flanagan 1992 calls "constructive naturalism" and what Foss 1995 calls "methodological materialism." Paul Churchland (1989) and Patricia Smith Churchland 1986 defend eliminative materialism. On functionalism, see Johnson-Laird 1983 and Block 1978.

Alan Turing proposed an imitation game to answer the question of whether computers can think. In this game, which has come to be known as the Turing test, an investigator communicates by typing with both a person and a computer. If the investigator cannot tell which is the human and which is the computer, then we should judge the computer to be intelligent. This test is both too loose and too restrictive. It is too loose in that a cleverly constructed program might be able to fool us for a while even though it contains little intelligence. It is too restrictive in that the computer may fall short on some fairly trivial aspect of human experience but be capable of highly intelligent functioning in other areas.

Creativity is often cited as a challenge for CRUM, but earlier chapters described several mechanisms that can model some aspects of human creativity, including abduction, conceptual combination, and analogy. Another interesting challenge is whether CRUM, neuroscience, and/or the dynamic systems view discussed in chapter 12 explain why people dream (Flanagan 2000). Bruner (1990) poses what might be called the *narrative challenge*, claiming that computational and biological approaches to thinking neglect the importance of story interpretation in how people understand each other, but researchers such as Kintsch (1998) have much to say about narrative coherence.

Glossary

Note: Words in italics have their own entries in the glossary.

Abduction Reasoning that generates hypotheses to explain puzzling facts.

ACT "Adaptive Control of Thought"—A computational theory of thinking developed by John Anderson.

Affective computing Study of computing technology that relates to, arises from, or deliberately influences *emotions*.

Algorithm A step-by-step procedure for solving a problem.

Amygdala Almond-shaped part of the brain involved in *emotions* such as fear.

Analogy Mental process that makes connections between relations in two sets of objects.

Anthropology The study of the origins, distribution, social relations, and *culture* of human beings.

Artificial intelligence The study of how computers can be programmed to perceive, reason, and act.

Backprogagation Learning *algorithm* in *feedforward* networks that adjusts the strengths of the links between *neurons*.

Bayesian network A directed graph that that can be used to reason with probabilistic information.

Case-based reasoning Reasoning by *analogy*.

Chaos Property of a *dynamic system* that it is highly sensitive to small changes.

Cognitive grammar Approach to *linguistics* that rejects the traditional separation of syntax and semantics.

Cognitive science The interdisciplinary study of mind and intelligence.

Coma State of deep unconsciousness caused by disease or injury.

Computation Physical process with states that represent states of another system and with transitions between states that amount to operations on the *representations*.

Concept *Mental representation* of a class of objects or events that belong together, usually corresponding to a word.

Conceptual change Process in which *concepts* acquire new *meaning*.

Conceptual combination Process in which new *concepts* are constructed by joining or juxtaposing old ones.

Connectionism Approach to *cognitive science* that models thinking by artificial *neural networks*.

Consciousness Mental state involving attention, awareness, and qualitative experience.

Cortex Outer layer of the brain, responsible for many higher cognitive functions.

CRUM Computational-Representational Understanding of Mind: the hypothesis that thinking is performed by *computations* operating on *representations*.

Culture The way of life of a society, including beliefs and behaviors.

Data structure An organization of information in a computer program.

Deduction Reasoning from premises to a conclusion such that if the premises are true then the conclusion must also be true.

Distributed artificial intelligence Problem solving that requires communication among more than one computer, each of which possesses some intelligence.

Distributed cognition Problem solving that requires communication among more than one thinker.

Distributed representation *Neural networks* that use patterns of activity in multiple nodes or *neurons* to stand for objects or situations.

Dopamine *Neurotransmitter* involved in reward pathways in the brain.

Dualism Philosophical view that the mind consists of two separate substances, soul and body.

Dynamic (dynamical) system Collection of interacting objects whose changes are describable by mathematical equations.

Electroencephalogram (EEG) Recording of electrical activity in the brain.

Embodiment Property of having a body and experiencing the world by means of it.

Emotion Positive or negative mental state that combines physiological input with cognitive appraisal.

Emotional intelligence Ability to deal effectively with the *emotions* of oneself and others.

Empiricism The philosophical view that knowledge comes primarily from sensory experience.

Explanation schema *Mental representation* of a pattern of causal connections.

Feedforward network Artificial *neural network* in which the flow of activity is in one direction, from input *neurons* to output neurons.

Frame Data structure that represents a *concept* or *schema*.

Functionalism Version of *materialism* according to which mental states are defined by their functional relations, not by any particular kind of physical realization.

Hebbian learning Process in *neural networks* that strengthens the association between two *neurons* that are simultaneously active.

Hippocampus Brain region involved in the acquisition of memories.

Image Mental structure that is similar to what it represents.

Induction Reasoning that introduces uncertainty.

Inheritance Form of inference in which information is transferred from a higher to a lower structure.

Innate A *representation* or process that is genetic rather than learned.

Insula (insular cortex) Brain region that integrates information from many bodily senses.

Intentionality Property of a *representation* or mental state that it is about some aspect of the world.

Lesion Abnormal change in an organ such as the brain.

Linguistics The study of language.

Link Connection between two artificial *neurons* that enables one to influence the activity of the other.

Local representation Artificial *neural network* in which each node stands for a single concept or proposition.

Logic The study of valid reasoning.

Magnetic resonance imaging (MRI and fMRI) Technique that uses magnets to produce images of the structure and function of organs.

Materialism Philosophical view that minds are purely physical.

Meaning The content of a *representation* that results from its relations to other representations and the world.

Mechanism System of interconnected parts that produces regular changes.

Memory Storage of information, either temporary (short-term or working memory) or permanent (long-term).

Mental model Mental structure that approximately stands for something in the world.

Mental representation A structure or process in the mind that stands for something.

Metaphor Use of language to understand and experience one kind of thing in terms of another.

Model Structure that approximately represents some objects or events.

Multiagent system Interacting collection of computers capable of intelligent action.

Neural network Interconnected group of *neurons*.

Neuron Nerve cell.

Neuroscience Study of the structure and functioning of brains.

Neurotransmitter Molecule that transmits nerve impulses across a *synapse*.

Parallel Process in which more than one computation is performed at the same time.

Parallel constraint satisfaction Process in which a problem is solved by using a parallel *algorithm* to find the best assignment of values to interconnected aspects of the problem.

Parallel distributed processing Approach to *cognitive science* that models thinking by artificial *neural networks* with *distributed representations*.

Philosophy Study of the fundamental nature of knowledge, existence, and morality.

Positron emission tomography (PET) Technique that uses radioactive isotopes to produces images of the chemical function of organs such as blood flow in the brain.

Prefrontal cortex Area of the brain at the front of the front of the *cortex*, responsible for the highest cognitive functions such as reasoning.

Production rule a representation of the form IF something THEN something.

Psychology Study of the minds of humans and other animals.

Rationalism The philosophical view that knowledge comes primarily by reasoning that is independent of sensory experience.

Recurrent network *Neural network* in which the output of some *neurons* feeds back via intervening connections to become input to them.

Relaxation Process in which an artificial *neural network* reaches a state of stable activations.

Representation A structure or activity that stands for something.

Robot Machine capable of performing complex physical acts similar to ones done by humans.

Rule A *mental representation* of the form IF something THEN something.

Schema A *mental representation* of a class of objects, events, or practices.

Search A computational process of looking for or carrying out a sequence of actions that lead to desired states.

Situated action Action that results from being embedded in a physical or social world.

SOAR "State, Operator, And Result"—A computational theory of thinking developed by Allen Newell and others.

Social cognition Study of how people think about each other.

Social epistemology Study of social practices that encourage or inhibit the development of knowledge.

Somatic marker Brain signal corresponding to states of the body relevant to *emotions*.

Source analog Set of objects, properties, and relations that suggests conclusions about a target analog.

Spike train Firing pattern of a *neuron*, consisting of a sequence of firing episodes.

Spreading activation Computational process in which the activity of one structure leads to the activity of an associated structure.

Syllogism Kind of *deduction* in which the premises and conclusions have forms such as "All A are B" and "No A are B."

Synapse Space in which a signal passes from one *neuron* to another.

Target analog Set of objects, properties, and relations that can be learned about by comparison to a source analog.

Theory Set of hypotheses that explain observations.

Thought experiment Use of the imagination to investigate nature.

Ventromedial prefrontal cortex The bottom-middle part of the *prefrontal cortex*.

Whorf hypothesis Conjecture that language determines how we perceive and think about the world.

References

Abraham, R. H., and C. D. Shaw. 1992. *Dynamics: The geometry of behavior.* 2nd ed. Redwood City, Calif.: Addison-Wesley.

Adams, M. J. 1990. *Beginning to read.* Cambridge, Mass.: The MIT Press.

Aitchison, J. 1987. *Words in the mind: An introduction to the mental lexicon.* Oxford: Basil Blackwell.

Akmajian, A., R. A. Demers, A. K. Farmer, and R. M. Harnish. 2001. *Linguistics: An introduction to language and communication.* 5th ed. Cambridge, Mass.: The MIT Press.

Allen, R. H., ed. 1992. *Expert systems for civil engineers: Knowledge representation.* New York: American Society of Civil Engineers.

Allman, J. M. 1999. *Evolving brains.* New York: Scientific American Library.

Anderson, J. A., and E. Rosenfeld, eds. 1988. *Neurocomputing.* Cambridge, Mass.: The MIT Press.

Anderson, J. A., and E. Rosenfeld, eds. 1998. *Talking nets: An oral history of neural networks.* Cambridge, Mass.: The MIT Press.

Anderson, J. R. 1983. *The architecture of cognition.* Cambridge, Mass.: Harvard University Press.

Anderson, J. R. 1993. *Rules of the mind.* Hillsdale, N.J.: Erlbaum.

Anderson, J. R. 2000. *Cognitive psychology and its implications.* 5th ed. New York: Worth.

Anderson, J. R., Y. Qin, V. A. Stenger, and C. S. Carter. Forthcoming. The relationship of three cortical regions to an information-processing model. *Cognitive Neuroscience* 16, 637–653.

Asada, M. et al. 2003. An overview of RoboCup-2002 Fukuoka/Busan. *AI Magazine* 24 (2, summer), 21–40.

Ashby, F. G., and E. Walrdron. 2000. The neuropsychological bases of category learning. *Current Directions in Psychological Science* 9, 10–14.