. in particular, can
ovides "a scientific
: "it will tell us not
but why" (Resnick
. concerted effort to
:e learning environ-
:he opening sentence
r class or economic
tools for developing
ıe utmost."

2

The Science of Mind: Analyzing Tasks, Behaviors, and Representations

The claim that our minds and computers are related—that they are both species of the same genus—might seem outrageous; however, as we come to understand this claim and the research methods based on it, we will begin to appreciate the similarities between mind and machine. From these similarities cognitive scientists derive theories that help explain how children learn and, consequently, how they could be taught more effectively.

A Balance-Scale Problem

Research on how children learn to solve balance-scale problems illustrates the main ideas, methods, and instructional applications of cognitive science.

Try to solve the balance-scale problem shown in figure 2.1. Assume the scale's arm is locked so that it can't rotate around the fulcrum. If I were to unlock the arm, what would happen? Would the scale tip left, tip right, or balance?

This is a tricky problem. Figure 2.2 gives a set of rules one might use to solve it. Each rule has an IF clause that states the conditions under which the rule is applicable and a THEN clause that states what to do under those conditions. To use these rules, find the rule whose conditions fit the pattern of weights and distances in the problem. You find that P4 is the only rule whose IF clause fits the problem. Its THEN clause tells you to compute torques for each side; that is, for each side, multiply the number of weights by their distance from the fulcrum. Doing that gives $t_1 = 5 \times 3 = 15$ for the left side and $t_2 = 4 \times 4 = 16$ for the right. These new data satisfy the condition for P7; executing its THEN clause gives the correct answer, "Right side down." Some readers might remember

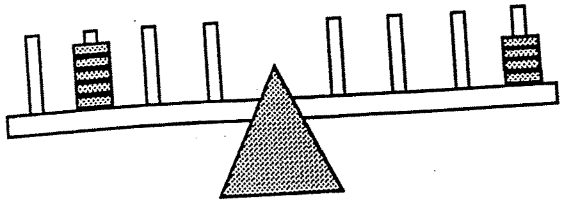


Figure 2.1
Will the scale tip left, tip right, or balance?

RULE IV

P1 **IF** weight is the same
THEN say "balance".

P2 **IF** side X has more weight
THEN say "X down".

P3 **IF** weight is the same **AND** side X has more distance
THEN say "X down".

P4 **IF** side X has more weight **AND** side X has less distance
THEN compute torques: $t_1 = w_1 \times d_1$; $t_2 = w_2 \times d_2$.

P5 **IF** side X has more weight **AND** side X has more distance
THEN say "X down".

P6 **IF** the torques are equal
THEN say "balance".

P7 **IF** side X has more torque
THEN say "X down".

Figure 2.2
The set of rules an expert might use to solve the balance-scale problem. (From Siegler and Klahr 1982, p. 198. Used with permission of Lawrence Erlbaum Associates.)

the THEN clause in P4 from high school physics as a version of the law of torques: Multiply weight by distance on each arm to find the torque, or rotational force; the side with the larger torque goes down. This simple law solves all balance-scale problems.

The set of rules is an English-language version of a computer program for solving balance-scale problems. It takes as input data about the weight on each side of the scale and the distance of the weight from the fulcrum. The output is the answer for a balance-scale problem: tip left, tip right, or balance. The program is a series of IF-THEN rules. Computer scientists call the IF clauses *conditions*, the THEN clauses *actions*, and the entire IF-THEN statement a *production rule*. They call computer programs written using only production rules *production systems*. Computing devices that execute production systems efficiently have a specific internal structure (or *architecture*, as computer scientists say).

Cognitive scientists claim that the human mind can be described as a computing device that builds and executes production-system programs. In fact, the rules in figure 2.2 are a production system an expert would use to solve balance-scale problems. Robert Siegler, a cognitive psychologist, showed that production systems can simulate human performance on such problems (Siegler 1976; Klahr and Siegler 1978; Siegler and Klahr 1982). He also showed that a series of increasingly complex production systems can model the way in which children gradually develop expertise on balance-scale problems from ages 5 through 17. Children learn, says Siegler, by adding better rules to their production systems. Proper instruction, he goes on to show, can help children acquire these better rules.

The Human Computer and How It Works

At the heart of the cognitive revolution was the realization that an adequate human psychology had to include the study of how the mind processes symbols. Computational theory gave psychologists a language and a framework for studying human symbol processing. Both minds and computers process symbols, use a small set of basic operations to manipulate them, and store them in memory. When we solve a balance-scale problem, we use a system of mental symbols to encode information about the problem, to manipulate that information, and to store the results of the manipulations in memory.

Symbol systems and basic operations, notions fundamental to cognitive theory, can be hard to understand. But we all understand how arithmetic works, and it provides a ready analogy for understanding symbols and operations. A symbol is an object that stands for or represents another object. In arithmetic, there are two kinds of symbols: numerals and operation signs. The content of arithmetic, what it is about, is numbers. Numerals are symbols for numbers. Each number has a unique numeral, and each numeral represents a unique number. In the Arabic system we can combine a few basic symbols (0, . . . ,9), using rules, to generate an infinity of numerals symbolizing an infinity of numbers. This ability to combine a few basic symbols systematically into more complex ones is a powerful feature of other symbol systems we use, such as the alphabet and speech sounds.

The arithmetic signs, +, −, ×, and ÷, are symbols that stand for operations on numbers. These four signs symbolize the basic processes of arithmetic. Like numerals, they can be combined to generate an infinity of other numeric operations. To find an average we combine addition and division; to figure out monthly mortgage payments we combine multiplication, addition, and division. Making more complex numerical operations out of simple ones gives arithmetic its power and its wide applicability.

Symbols and processes in computing work the same way. There are two kinds of symbols: those that stand for input or data (what the computation is about) and those that stand for operations on the data. In computing, the basic operations include recognizing when two symbols are the same, creating new symbols, storing a symbol in memory, and retrieving a symbol from memory. The basic operations are part of the computer's hardware or the primitive commands in a particular computer language. Using rules, we can combine the basic operations to form more complex operations. We call these more complex, elaborate operations *computer programs*. Depending on what the data are and how we combine the basic processes, we can make the computer into a word processor, a spreadsheet, or a flight simulator.

Cognitive scientists assume that the human mind works by applying elementary processes to symbol structures that represent the content of our thoughts. On the balance-scale problems, we use symbols to encode variables, such as sides, weights, and distances. Cognitive scientists call these symbol structures *mental representations*.

The idea of representation is fundamental to cognitive science. Representations are the symbol structures we construct to encode our experience, process it, and store it in our memories. Representations are the symbolic links between the external environment and our internal, mental world. The representations we construct in encoding our experiences have profound effects on our behavior and our learning.

In the balance-scale task, we use basic processes such as comparing two symbols (which side has more weight?), creating new symbols (finding the torques), and retrieving information from memory (what is 5 × 3?). When we combine a few basic processes in the right way, we get a production system with which to solve balance-scale problems.

The expert's production system for the balance scale is a combination of basic processes that operate on symbols representing objects in the environment. If we know the initial input to the production-system program, we can predict what the expert will say about any balance-scale problem. The program not only gives the same answer as the expert but also simulates the expert's performance by telling us exactly what the expert knows and does to arrive at that answer. The production system is a cognitive theory of expert performance on balance-scale problems.

The production system for the balance scale is an example of how a good cognitive theory of a task not only explains the task but also performs the task as a human would. The same approach and the same criterion for being a good cognitive theory apply in other domains—math, physics, reading, and even writing. What varies among the domains and among problems within the domains are the specific representations we use and the ways in which we combine our basic cognitive processes to operate on the representations. Different representations and combinations of operators allow us to play chess, solve physics problems, and write essays, just as different data programs turn a computer into a word processor or a spreadsheet. The challenge for the cognitive scientist is to identify the representations and elementary processes we use and to discover how we combine them to build the programs that guide our actions.

Processing and Storing Symbols: Human Memory Structures

Minds differ from digital computers in some obvious ways. Minds are made of neural tissue, computers from silicon and copper. Minds

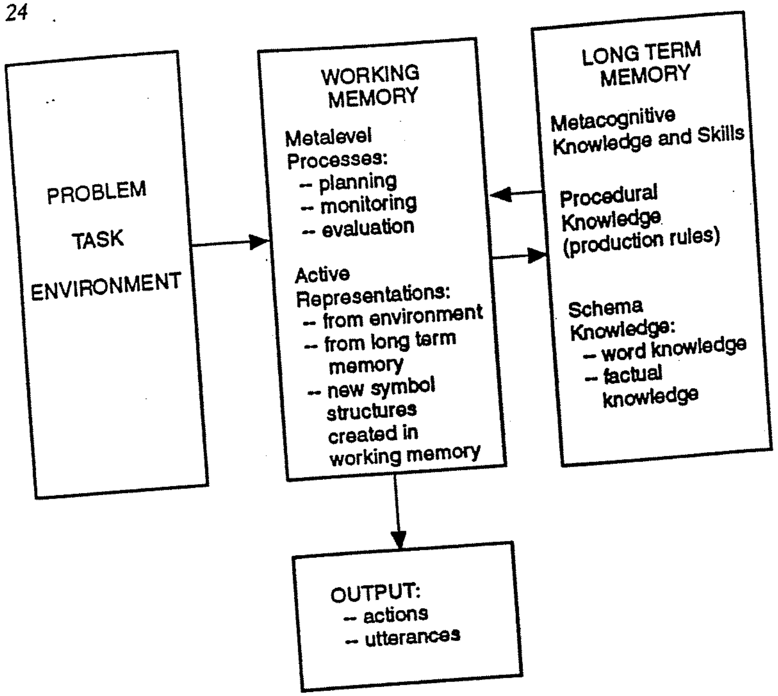


Figure 2.3
The standard picture of the human cognitive architecture.

evolved; computers are designed. Another important difference between minds and computers lies in their built-in capacities, characteristics, and memory structures—the attributes that enable them to construct and run programs. As a computer scientist would say, minds and computers have different *computational architectures*.

Borrowing from computing, cognitive scientists speak of our *cognitive architecture*, the built-in mental features that allow our minds to build and execute programs. Figure 2.3 gives the standard picture of the human cognitive architecture. Information from the external world (what cognitive psychologists sometimes call the *task environment*) comes to us through our sensory systems. Some of this information enters working memory, where we process it. We store some of this processed information for future use in long-term memory. Working memory is the part of our cognitive architecture where mental computations actually take place; in this respect it is much like a computer's central processing unit. Long-term memory is for

storage, like a computer's hard disk. Features of these two memory structures account for both the strengths and the weaknesses of human cognition.

Most of us, when we think of "memory," think of long-term memory—our permanent storehouse of knowledge and skills. Our long-term memory appears, for all practical purposes, to have unlimited capacity: no one has ever reported a case of an otherwise normal person who couldn't learn and remember new things. The most important feature of long-term memory for learning and instruction is not its capacity but its internal organization. Unlike a digital computer, you don't store a chunk of information in long-term memory by giving it an "address" in your brain, an address that you look up when you want to retrieve the information. Long-term memory has what psychologists call an *associative structure*. Symbol structures represent items or chunks of information in memory, and associative links tie the items together into networks of related information. We create associative links between chunks if we use the chunks together repeatedly, learn them together, or experience them together.

Cognitive psychologists have discovered that long-term memory is not a single entity; it comes in a variety of forms. At the most general level, they distinguish declarative from nondeclarative memory. Declarative memory contains a system for remembering specific events (what psychologists call *episodic memory*) and a system for remembering general facts and word meanings (*semantic memory*). We consciously recall items from declarative memory, and we can express or describe the items we retrieve. This is not so for the contents of nondeclarative memory. Among other things, nondeclarative memory contains our memory for motor, perceptual, and cognitive skills—our memory for procedures. The contents of nondeclarative memory are not always open to conscious recall, nor can they always be expressed or accurately described. Tennis players have a motor skill to hit backhands, but when they execute the skill they don't consciously recall the procedure; they just hit the backhand. As you read this text you are executing a complex motor, perceptual, and cognitive skill, yet you can't describe how you transform the marks on the page into meaningful prose.

To understand problem solving and high-order cognition, we can focus on semantic and procedural memory—our memories for facts and skills. Although semantic and procedural memory both have associative structures, their structures are slightly different. The expert rule system illustrated in figure 2.2 is an example of a procedural memory structure. The associations in procedural memory form rules. Individual rules represent associations between chunks of information, where the chunks are the conditions and actions in the rules. The expert has learned to associate certain actions with certain conditions. The expert also associates the seven rules together as a system because collectively the rules are useful for solving balance-scale problems. There are also implicit associations between rules. For example, the action of P4 generates the conditions for either rule P5 or rule P6. Sometimes rules used together repeatedly combine to form a single, more complex, new rule. The rules and their organization give the expert a way to move from chunk to chunk in long-term memory.

Psychologists call the associative structures in declarative memory *schemas*. Schemas are network structures that store our general knowledge about objects, events, or situations. Figure 2.4 illustrates how our general knowledge about animals might be stored as a schema in semantic memory. In this example, the central node is "animal." The "is a" links connect the major nodes in the hierarchy that organizes our biological knowledge. Both mammals and birds are animals; a canary is a bird, but a bear is not. The "has," "can," and "is" links associate the various biological types with important properties or features. When we learn something new about bears or canaries, the information isn't passively inscribed at the end of our memory tape; rather, we integrate the new item into a preexisting schema.

Our associative memory structures are like little theories we apply to negotiate and understand the world. The associative structures help us make predictions—as with the balance scale—and help us make inferences that go beyond what we literally experience. For example, if you tell me that Tweety is a canary, I can infer that he is yellow, is a bird, and has feathers. Our schemas also help us know what to expect in situations. My schema for a baseball game leads me to expect that I will spend around 3 hours at the ballpark, and that if I eat there my dinner will be hot dogs and soda, not Dover sole.

These associative structures do not simply provide a way to store information; they also influence what we notice, how we interpret it, and how we remember it. In one famous memory study,

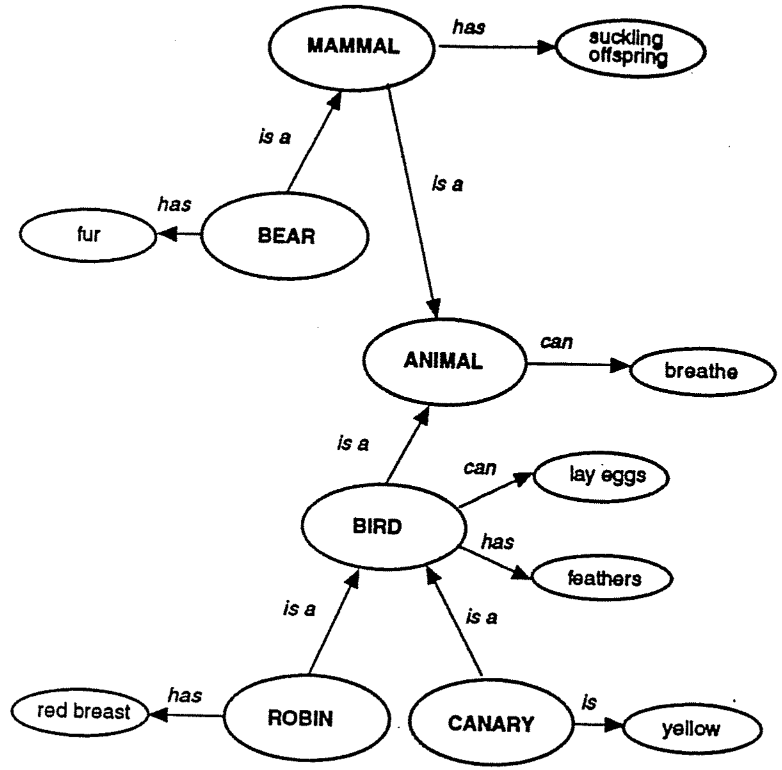


Figure 2.4
An example of a schema showing how knowledge about animals might be stored in long-term memory. The nodes represent concepts; the links represent relations among concepts. (From Just and Carpenter 1987, p. 66. Used with permission of Allyn and Bacon, Inc.)

a research assistant brought subjects, one by one, to a waiting room and told the subjects that this room was the experimenter's office (Brewer and Treyens 1981). After each subject had sat in the room for 35 seconds, the assistant took him to a seminar room and asked him to write down everything he could remember about the "office." All the subjects recalled correctly items that were in the room and were usually in most offices—a desk, a chair, etc. Yet only a third of the subjects remembered odd items, such as a skull, that were in the waiting room but not typically in an office. Conversely, a third of the subjects incorrectly reported remembering items that are usually in an office, such as books, but which were not present in the waiting room. The subjects' office schemas influenced what they noticed, what they remembered correctly, and what they "remembered" incorrectly about the waiting room.

Associative memory structures are powerful devices for organizing and deploying our skills and knowledge. Like other theories, they also actively influence what we perceive. Just as scientific theories influence what scientists see and consider important (for example, a social psychologist and an epidemiologist would notice different things about a group of coughing passengers on an airliner), our memory structures influence what we see and consider important. Prior knowledge influences what we notice and how we interpret new experiences. Thus, prior knowledge affects how we interpret school instruction and thus affects what we can learn. School instruction that ignores the influence of preexisting knowledge on learning can be highly ineffective.

If long-term memory is the storehouse, then working memory is the clearinghouse. *Working memory* is the term psychologists use to refer to the cognitive resources we use to execute mental operations and to remember the results of those operations for short periods of time (Baddeley 1992). Working memory contains all the symbol structures that are active and available for processing at any given time and keeps an internal record of the current state of mental activity. The inputs to working memory are symbols that encode information coming from the external world or symbol structures retrieved from long-term memory. When these structures are processed, the results can be new symbol structures in working memory, symbols for storage in long-term memory, or commands to the motor system to do or say something.

Working memory's most significant characteristic is its limited capacity. The capacity of short-term memory, as George Miller showed, is about seven plus or minus two chunks. In a short-term memory task, all we have to do is remember some information. Working memory is the information-processing descendant of short-term memory. Working-memory tasks combine the demand for remembering information with the demand for doing some processing on that information.

Our capacity to remember *and* process information is understandably less than our capacity to remember alone. You can convince yourself of this by some simple self-experimentation. For a short-term-memory task, cover the digits below with a piece of paper. Then uncover them one at a time, exposing one new digit every second. After 10 seconds, cover all the digits again and see how many you remember.

9 7 2 1 6 8 9 3 0 4

For a working-memory task, pick a paragraph in this book, read it aloud, and as you read try to remember the last word of every sentence. When you finish the paragraph, say aloud the words you tried to remember (Daneman and Carpenter 1980). Most people can do this for two sentences, but few people can do it for more than four. You will find that your short-term digit span is around seven plus or minus two digits, but that the capacity of your working memory, as measured by how many final words you can remember, is more like four plus or minus one.

Working memory can hold and process only a limited amount of information, and that for only short periods of time. We can quickly exceed its capacity, and when we do that any new information coming into working memory overwrites or obliterates what was previously there. Working-memory capacity is a limiting factor in our ability to process information. It is the bottleneck in our cognitive system. Skilled thinking, problem solving, and learning depend on how well we can manage this limited resource—on how efficiently we can store, process, and move information into and out of working memory.

How does the human computer work? How do these notions about symbol structures, representations, production systems, and memory structures fit together?

CYCLE	WORKING MEMORY CONTENTS	PRODUCTION
1	LEFT: w = 5, d = 3 RIGHT: w = 4, d = 4	P4 **IF** side X has more weight and X has less distance **THEN** compute torques ...
2	LEFT: torque = 15 RIGHT: torque = 16	P7 **IF** side X has more torques **THEN** say "Side X down."
3	SAY: "Right down."	No match, so halt.

Figure 2.5
An illustration of how an expert uses the rule in figure 2.2 to solve the balance-scale problem in figure 2.1. On each cycle the expert tries to match contents of working memory with the IF clause in a rule. When a match is found, that rule fires, changing the contents of working memory and starting a new cycle.

Using the production system illustrated in figure 2.2 to solve the balance-scale problem shown in figure 2.1 gives a simple example. Figure 2.5 sketches the process. Assume that an expert on balance-scale problems has stored the production system in long-term memory. When presented with the problem, the expert encodes relevant information about the problem into working memory. The expert forms an initial representation of the problem that includes symbols for side, weight, and distance. The first cycle starts with a search through the rules to find one whose conditions match the current contents of working memory. The contents of working memory match P4, which fires and changes the contents of working memory. Cycle 2 begins, and the only match is with P7. P7 fires, changing the contents of working memory, and cycle 3 begins. "SAY: 'Right down'" doesn't match the condition of any rule, so the program stops. Working memory sends the appropriate command to the brain's speech centers, and the expert says "Right down."

Production systems can become very complex, but the basic mode of operation remains the same. The system looks for matches between symbols active in working memory and conditions on production rules in long-term memory. When a match is found, that rule fires, modifying the contents of working memory—and the cycle begins again. When no match can be found, the program halts. That, in short, is how cognitive scientists think the human computer works.

Problems and Representations

Psychology is a science of human behavior that develops theories about how we react or respond in various situations or environments. According to cognitive science, all humans share the same basic cognitive architecture, although memory capacity and speed of processing may vary among individuals. Differences in our behavior arise from the ways in which our cognitive architectures, including individual differences in those capacities, interact with the environment. If cognitive scientists are to describe this interaction, then not only do they have to describe the computing device and its capacities carefully; they also have to describe the environment carefully.

To do the latter, cognitive scientists think of the external world in terms of *task environments*. A task environment is a problem plus the context in which a subject encounters the problem. For the balance-scale task, the environment consists of a balance scale and an experimenter who poses the problem by asking the subject for a prediction about what the scale will do.

Cognitive scientists use the word *problem* in a special way. The idea is simple, and it borrows from our everyday use of the word. As Newell and Simon wrote, "a person is confronted with a *problem* when he wants something and does not know immediately what series of actions he can perform to get it" (1972, p. 72). Cognitive psychologists elaborate and refine this general notion. They think of a problem as consisting of an *initial state* or situation and a *goal state* (i.e., what the person wants). To solve a problem, a person must figure out what to do to move from the initial state to the goal state. The things a person can do, the moves he or she can make in a problem situation, cognitive psychologists call *operators*. For example, a chess game is a problem in which the initial state is the opening position, the goal is checkmate, and the operators are legal moves. In solving a problem, then, we use operators to create a chain of working-memory states that begins with the initial state and ends with the goal state.

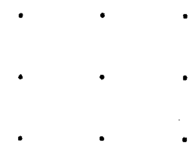
One way to view the development of cognitive science, especially as it relates to education, is to see how this characterization of problem solving was extended and modified as researchers applied it to increasingly complex domains. That is how this book is organized.

In the balance-scale problem, the initial state is the locked scale with weights on each arm. The goal state is the correct prediction of a unique outcome: left, right, or balance. Among the operators we could apply are the expert's production rules. As we have seen, these rules create a series of states in working memory leading from the initial state to a correct prediction.

The cognitivist's core notion of a problem applies directly to solving most school math and science problems. In a high school geometry problem, the "givens" are the initial state, "to prove" states the goal, and the operators are geometrical definitions, postulates, and theorems. There is a unique, well-defined goal, but there are various ways to move from the givens to that goal. In reading, the text is the given and the goal is to construct an interpretation of the text. However, for some kinds of texts—in contrast with solving geometry problems—there need not be a unique interpretation. That is what makes literature personally rewarding and intellectually challenging. Writing demands the solution of what cognitive scientists call *ill-defined problems*. With writing problems (for example, writing an essay), there is no unique solution and no standard, universal method of finding solutions. Often, it is only after we start solving an ill-defined problem that we have an idea of what an adequate solution might be. Teaching a classroom lesson presents an ill-defined problem that the teacher has to solve on the spot, where every student-teacher interaction can change the teacher's goals and choice of operators. Many everyday tasks (finding a job, planning a trip) and most creative tasks (writing a symphony, doing medical research) present ill-defined problems.

As was mentioned above, representations are the link between the external world and our internal processing system. A person's problem representation is what the person encodes about the problem from the task environment. It is the solver's interpretation or understanding of the problem—an interpretation based on experience and on beliefs about the major variables or factors relevant to the problem. Preexisting knowledge, stored as productions or schemas, guides the solver's interpretation.

A person's problem representation is seldom identical to the problem statement in the task environment. The famous nine-dot problem is the classic example. Draw four straight lines that pass through all nine dots without raising the pencil from the page:



Your initial representation dictated how hard or easy you found the problem. If you found the problem hard, most likely your representation (your interpretation of the problem statement) prohibited lines going outside the boundary defined by the dots. There was no such prohibition in the problem statement. If your representation allowed the four lines to go outside the boundary, you probably solved the problem easily. (One possible solution is shown in note 1 at the back of the book.)

What the nine-dot problem shows is that sometimes we include information in our representations that is not in the problem statement or the task environment. At other times, we might include in our representations information that is present in the task environment but that is not relevant to the problem. For example, we might encode the color of the weights in a balance-scale problem. Sometimes, too, we might fail to encode information that is present and relevant. As we will see, young children often don't encode distance information when doing balance-scale problems.

Our initial problem representations are important because they shape the course of our problem solving. The initial representation determines what we take to be the initial state and can influence what we take to be the goal and the legal operators. In this way, the initial representation constrains what cognitive psychologists call the solver's *problem space*. The problem space is the set of all possible knowledge states the solver can construct from the initial state using the legal operators. Using the operators, then, the solver can generate many possible paths, or chains of knowledge states. Some of the possible paths lead to the goal; others do not. Solving a problem consists of using the operators to find a path of knowledge states from the initial state to the goal state. A poor initial representation can make an easy problem hard or impossible. The poor initial representation of the nine-dot problem results in a choice of legal moves that can't lead to a solution. The problem space is so small that there are no paths from the initial state to the goal. A good initial representation and a suitable problem space, in contrast, can make an otherwise hard problem trivial.

Analyzing the Task

Knowing how you behaved in tackling the nine-dot problem, I could predict something about your psychology, your internal symbol processing. I could tell how you represented, or understood, the problem. How is this possible? The trick is that more information is available to me than just your performance on the problem. I have a complete understanding of the task and what it demands. I know as a matter of geometrical fact that there are only two options for drawing the straight lines: either they stay within the boundary or they go outside. I also know that there is no solution using only lines that fall inside the boundary. What I know about the task and about your problem-solving behavior allows me to figure out how you must have understood the task. I can figure out what representation you used.

In the same way, cognitive scientists can discover what representations and rules people use on more complex problems. Cognitive psychologists begin their research on problem solving with what they call a *task analysis*. They try to define what the major variables and causes are in a given type of problem. They try to figure out what knowledge and skills the problem demands, and given those demands, what ideal performance on the problem would be. Scientifically, task analysis is essential for solving the problem that cognitive scientists have set for themselves. We can think of what cognitive scientists are trying to do in terms of an equation:

Task demands + Subject's psychology = Behavior.

Most of the time, cognitive psychologists are trying to solve this equation for "Subject's psychology," the subject's unobservable mental processing. To do so they need values for the other two variables. They can observe a subject's behavior, and task analysis gives values for the "Task demands" variable. If they have possible values for the "Task demands" and "Behavior" variables, they can derive values for "Subject's psychology." Of course, we should not overinterpret this analogy. Doing cognitive research is not as simple as solving an

equation. Exacting task analysis and careful observation of behavior place constraints on the subject's psychology, but may not uniquely determine it. Task analysis plus behavior allows researchers to formulate intelligent hypotheses about subjects' psychology that they can put to further experimental tests. This is what Siegler did with children who were solving balance-scale problems.

Analyzing the Balance-Scale Task

In the late 1920s, the psychologist Jean Piaget opened up a new intellectual world with his scientific studies of children's thinking, which ranged across all domains of reasoning—logical, quantitative, spatial, and causal. Piaget held that children's thinking develops from infancy through adolescence in a series of discrete stages. At each successive stage, children acquire increasingly sophisticated, abstract, logical structures that guide their reasoning across all domains (Inhelder and Piaget 1958). Modern researchers have refined, corrected, and sometimes overturned Piaget's original interpretations and some aspects of his theory, but his groundbreaking research shaped the course and agenda of research on cognitive development. One of the tasks Piaget used in his studies was the balance scale; understandably, a body of research has grown up around that task.

The beauty of the balance-scale task for developmental psychology is that it is complex enough to be interesting but simple enough for exhaustive task analysis. Two variables are relevant: the amount of weight on each arm and the distance of the weight from the fulcrum. There are three discrete outcomes: tip left, tip right, and balance. There is a simple law, the law of torques, that solves all balance-scale problems, though few of us discover this law on our own. If weight and distance are the only two relevant variables and if the scale either tips or balances, there are only six possible kinds of balance-scale problem:

- balance problems—equal weight on each side and the weights at equal distances from the fulcrum;
- weight problems—unequal weight on each side and the weights equal distance from the fulcrum;
- distance problems—equal weight on each side and the weights at unequal distances from the fulcrum;
- conflict-weight—one side has more weight, the other side has its weight at a greater distance from the fulcrum, and the side with greater weight goes down;

- conflict-distance—one side has more weight, the other side has its weight at a greater distance from the fulcrum, and the side with greater distance goes down;
- conflict-balance—one side has more weight, the other side has its weight at a greater distance from the fulcrum, and the scale balances.

Robert Siegler called the last three types "conflict" problems, because when one side has more weight but the other side has its weight farther from the fulcrum one can have conflicting intuitions about which variable dominates. (The problem illustrated in figure 2.1 is a conflict-distance problem: there is more weight on the left side, the weight is farther from the fulcrum on the right side, and the right side goes down.)

These six possibilities cover all possible cases for how weight and distance influence the action of the scale. The six cases provide a complete theory, or task analysis, of the balance scale. Notice that the six problem types place varying demands on the solver. For a balance problem or a weight problem, a solver need only consider weight. For the conflict problems, a solver has to pay attention to weight, distance, and the ways in which weight and distance interact.

Siegler formulated some psychological hypotheses about how people might solve balance-scale problems. Using the information from the task analysis, he could test his hypotheses by giving subjects problems and observing their performance. Siegler called his hypotheses "rules" and formulated them as four production-system programs. His rules I–III are given in figure 2.6; his rule IV is the expert's production system of figure 2.2 above.

The rules make different assumptions about how and when people use weight or distance information to solve the problems. Rule I considers only weight. Rule II considers distance, but only when the weights on the two sides are equal (P3). Rule III attempts to integrate weight and distance information (P4 and P5). Rule IV introduces the law of torques (P4) when one side has more weight but less distance.

Knowing the task and having hypotheses about the subjects' psychology gave Siegler values for two of the three variables in the cognitivist's equation that interrelates task, psychology, and behavior. This allowed him to generate values for the behavior variable. He could predict how subjects would perform. Siegler analyzed how the four rules worked on the six kinds of problems and determined that each rule would give a distinct pattern of right and wrong

RULE I

P1 **IF wieght is the same**
THEN say "balance".

P2 **IF side X has more weight**
THEN say "X down".

RULE II

P1 **IF** weight is the same
THEN say "balance".

P2 **IF** side X has more weight
THEN say "X down".

P3 **IF** weight is the same **AND** side X has more distance
THEN say "X down".

RULE III

P1 **IF** weight is the same
THEN say "balance".

P2 **IF** side X has more weight
THEN say "X down".

P3 **IF** weight is the same **AND** side X has more distance
THEN say "X down".

P4 **IF** side X has more weight **AND** side X has less distance
THEN make an educated guess.

P5 **IF** side X has more weight **AND** side X has more distance
THEN say "X down".

Figure 2.6
Siegler's rules I–III for the balance-scale task. (From Siegler and Klahr 1982, p. 198. Used with permission of Lawrence Erlbaum Associates.)

- conflict-distance—one side has more weight, the other side has its weight at a greater distance from the fulcrum, and the side with greater distance goes down;
- conflict-balance—one side has more weight, the other side has its weight at a greater distance from the fulcrum, and the scale balances.

Robert Siegler called the last three types "conflict" problems, because when one side has more weight but the other side has its weight farther from the fulcrum one can have conflicting intuitions about which variable dominates. (The problem illustrated in figure 2.1 is a conflict-distance problem: there is more weight on the left side, the weight is farther from the fulcrum on the right side, and the right side goes down.)

These six possibilities cover all possible cases for how weight and distance influence the action of the scale. The six cases provide a complete theory, or task analysis, of the balance scale. Notice that the six problem types place varying demands on the solver. For a balance problem or a weight problem, a solver need only consider weight. For the conflict problems, a solver has to pay attention to weight, distance, and the ways in which weight and distance interact.

Siegler formulated some psychological hypotheses about how people might solve balance-scale problems. Using the information from the task analysis, he could test his hypotheses by giving subjects problems and observing their performance. Siegler called his hypotheses "rules" and formulated them as four production-system programs. His rules I–III are given in figure 2.6; his rule IV is the expert's production system of figure 2.2 above.

The rules make different assumptions about how and when people use weight or distance information to solve the problems. Rule I considers only weight. Rule II considers distance, but only when the weights on the two sides are equal (P3). Rule III attempts to integrate weight and distance information (P4 and P5). Rule IV introduces the law of torques (P4) when one side has more weight but less distance.

Knowing the task and having hypotheses about the subjects' psychology gave Siegler values for two of the three variables in the cognitivist's equation that interrelates task, psychology, and behavior. This allowed him to generate values for the behavior variable. He could predict how subjects would perform. Siegler analyzed how the four rules worked on the six kinds of problems and determined that each rule would give a distinct pattern of right and wrong

RULE I

P1 IF wieght is the same
THEN say "balance".

P2 IF side X has more weight
THEN say "X down".

RULE II

P1 IF weight is the same
THEN say "balance".

P2 IF side X has more weight
THEN say "X down".

P3 IF weight is the same **AND** side X has more distance
THEN say "X down".

RULE III

P1 IF weight is the same
THEN say "balance".

P2 IF side X has more weight
THEN say "X down".

P3 IF weight is the same **AND** side X has more distance
THEN say "X down".

P4 IF side X has more weight **AND** side X has less distance
THEN make an educated guess.

P5 IF side X has more weight **AND** side X has more distance
THEN say "X down".

Figure 2.6
Siegler's rules I–III for the balance-scale task. (From Siegler and
p. 198. Used with permission of Lawrence Erlbaum Associates.)

answers. By giving a subject a set of problems that contain several examples of each problem type, Siegler could tell which rule the subject was using.

Figure 2.7 presents the pattern of performance Siegler predicted. Notice that if there are three outcomes for a balance-scale task—tip left, tip right, and balance—then 33 percent correct is chance performance; it amounts to a guess. Guessing is clearly distinguishable from always being wrong about a problem type (getting 0 percent correct). Notice, too, that sometimes adopting a more advanced rule can result in lower rather than higher performance on some problems. Going from rule II to rule III improves overall performance, but using rule III lowers performance on conflict-weight problems from 100 percent to 33 percent correct.

Finding Out What Children Know

If children use Siegler's rules, then the pattern of a child's responses to a set of balance-scale problems that contains all six types will reveal what rule that child uses. Children's responses will tell us what they know about the balance-scale task, including how they represent the problem. Siegler tested his hypotheses and predictions by giving a battery of 30 balance-scale problems to a group of 40 children that included equal numbers of 5-year-olds, 9-year-olds, 13-year-olds, and 17-year-olds. He showed each child a balance scale that had weights placed on it and asked the child to predict what the scale would do. As soon as the child made a prediction, Siegler rearranged the weights for the next problem. He did not let the children see if their predictions were correct, because he wanted to find out what they knew initially. He wanted to avoid giving the students feedback on their performance so he could be sure they weren't learning about the task during the experiment. He wanted to look at their learning, but only after he assessed their initial understanding.

The children's performance confirmed Siegler's hypotheses. Ninety percent of them made predictions that followed the pattern associated with one of the four rules. There was also a strong developmental trend. The 5-year-olds most often used rule I. The 9-year-olds used rule II or rule III. The 13- and 17-year-olds used rule III. Only two children, a 9-year-old and a 17-year-old, used rule IV. The developmental trend matched almost exactly the pattern of predictions in figure 2.7. The developmental trend, like the pattern of

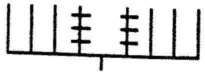
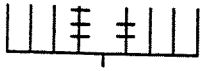
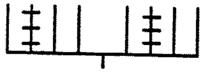
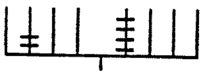
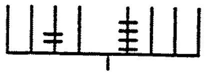
Problem Type	RULE I	RULE II	RULE III
 Balance	100	100	100
 Weight	100	100	100
 Distance	0	100	100
 Conflict-Weight	100	100	33
 Conflict-Distance	0	0	33
 Conflict-Balance	0	0	33

Figure 2.7
Siegler's predictions of the percentage of each type of balance-sc
subject would solve correctly using various rules. (Adapted from
p. 168. Used with permission of Lawrence Erlbaum Associates.)

predictions, included a decrease in performance on conflict-weight problems as the children grew older. The 5-year-old children, the majority of whom used rule I, answered 86 percent of the conflict-weight problems correctly. The 13-year-olds and the 17-year-olds, most of whom had progressed to rule III, answered only 50 percent of the conflict-weight problems correctly.

As these results confirm, Siegler's rules qualify as a cognitive and developmental theory for the balance scale. As a cognitive theory should, his rules explain behavior in terms of symbol structures that children have stored in their long-term memories. The individual rules tell us what knowledge children use. The production system tells us how they organize their knowledge. Chunks of information which children encode from the task environment or generate in working memory are the conditions that cause the rules to fire. When written in a suitable computer language, the rules can be run as programs on computers, and they simulate human performance. As a good cognitive theory should, the theory embodied in Siegler's rules performs the task it explains and explains the task in terms of representations and mental processes.

Taken together, Siegler's four rules constitute a developmental theory that explains development in terms of changes in knowledge structures and problem representations. By age 5, most children are using rule I. By age 13, almost all are using rule III. Few children spontaneously progress to rule IV, which represents expert performance on the balance scale. Thus, the rules chart a course of normal development on the task, from novice to expert performance.

Siegler's rules also tell us what cognitive changes underlie the transition from novice to expert. On tasks like the balance scale, children progress through a series of partial understandings that gradually approach mastery. Performance improves, or learning occurs, when children add more effective production rules to the theories they have stored in long-term memory. Chunking plays a role in this process. Children chunk information about weight and distance to construct more complex condition clauses and build more sophisticated rules. The balance-scale task presents a simple example of how, as children develop or learn, the chunks or concepts they use in problem solving become larger and richer. Bigger chunks, more complex concepts, and better rules are ways in which experts differ from novices, not only on the balance-scale task, but in most subjects and problem domains.

Siegler's four rules, viewed as a developmental theo
a simple example of how, as Robert Glaser claimed, cogn
can give us developmental theories of performance ch
know what the developmental stages are and how they
level of detail provided by a cognitive theory, we ough
to design instruction to help children advance from one
next. We might be able to improve children's learnin
work on the balance-scale task, as we shall see below, is
of this as well.

Experts and Response-Time Studies

Those familiar with physics or computer programs mig
the claim that rule IV captures an expert's understanc
balance scale. After all, we know that the law of torques is
rule for solving balance-scale problems. We also know th
to write a computer program that computes torques:
weight and distance information for each side. Multiply
distance for each side. If the products are the same, print
otherwise print out which side has the greater torque.

Rule IV is more complex. According to this rule
resorts to computing torques only in certain situations.
cognitive scientist claim that experts use rule IV and do
compute torques on all problems?

We can't decide between the two theories by looki
and wrong answers, because both rule IV and the law
solve *all* problems correctly. But once we have two diffe
programs that solve all the same problems, we can a
different kind of data to tell which one experts really
computer programs, even programs that for the same inp
the same output, can differ in how long they take to co
computation. This can occur when one program contains
or takes more steps to find an answer for certain pattern
Rule IV and the law of torques might differ in just this w
we can look at how long it takes experts to solve certai
balance-scale problems and figure out from their respo
which of the two programs they use.

From the perspective of a response-time study, ever
scale problem falls into one of two groups. We can solv

weight, and distance problems without computing torques, but to solve most conflict problems we have to compute torques. If experts always compute torques, then they use the same program on all problems and their response times should be the same for all problems. If experts use rule IV, then they first try to solve the problem without computing torques, and they do the numerical computation only as a last resort. This means that experts' response times on conflict problems should be longer than their response times on balance, weight, and distance problems.

Siegler tested twelve adult experts and found that they solved balance, weight, and distance problems in 1.5 to 2 seconds. To solve conflict problems, the experts took 3 to 3.5 seconds. Using response-time data, we can conclude that experts don't compute torques on all problems. Experts use rule IV.

Our ability to describe problem solving at a detailed level sometimes allows us to compare theories as we would compare computer programs. Response-time measures give us additional data that might support one cognitive theory over another. This method of response-time analysis has been particularly useful in discovering the methods children use to solve simple arithmetic problems.

Why do experts use rule IV and not the simpler law of torques? Expert behavior on the balance-scale task is just one example of how we unconsciously adopt strategies to minimize demands on working memory. To solve balance, weight, and distance problems, all we have to remember is which weight and which distance is greater. That is, we need only two items of comparative, qualitative information. To compute torques we have to remember the exact values of four numbers. Experts use the less demanding qualitative solution whenever they can and avoid committing any more cognitive capacity to a task than is necessary. The ability to manage cognitive resources efficiently and often unconsciously when solving problems in a knowledge domain is a sign of expertise in that domain.

The Balance Scale and Learning

So far, we have seen how cognitive research can generate theories about children's knowledge and how they use it to solve problems. With theories like Siegler's that describe what goes on at discrete levels of performance, we also can begin to investigate how children

make transitions between levels; that is, we can study ho
learn and how they might learn most effectively.

To investigate how children learn about the balance sc
conducted a training study. Working with 5-year-olds a
olds, all of whom used rule I, he had each child make
for 16 problems. After each prediction, Siegler released t
the balance scale and let the child see if his or her pred
correct. This feedback experience gave the children an o
to learn about the balance scale. Two days later, the chil
retest with no feedback to see if they had learned anythin
training.

In this experiment, there were three training groups.
of 5- and 8-year-olds served as a control group. Their traini
consisted only of balance and weight problems—problems
solve using rule I. A second group had training on distance
where rule II, but not rule I, would work. A third group ha
on conflict problems, which require at least rule III for pe
even at chance levels. With this training, would the chil
anything? Would they progress from rule I to a more adva

As expected, the children in the control group made no
They learned nothing from training on problems they alre
how to solve. The children in the second group, both 5- a
olds trained on distance problems, did learn something.
from 16 problems was enough for these children to adva
rule I to rule II. The surprise came with the third group, th
who had training on conflict problems. The 8-year-olds in t
advanced two levels in their mastery of the balance scale,
I to rule III. *The 5-year-olds in this group either stayed at rule I*
so confused and erratic that it appeared they were no longer using

From a researcher's perspective, this is a troubling res
if we have detailed knowledge about children's initial under
we can't necessarily predict how children will respond to
There must be more involved in learning than an interaction
the children's current rule and the training they receive. E
ally, the result is even more worrisome. In most classrooms,
don't have the time or the tools to figure out what k
children have before instruction. Teachers have to give
lessons to everyone and expect, or hope, that all students re
instruction in approximately the same way. In view of
result, this is an unrealistic expectation.

How are 8-year-olds different from 5-year-olds? Why do the older children, but not the younger children, learn from training on conflict problems? To answer this question, the cognitive scientist needs finer-grained data than are provided by task analyses, response patterns, and response times. Cognitive scientists use a method called *protocol analysis* to collect such fine-grained data.

Problem solving occurs in working memory, and we are consciously aware of at least some of the information our working memory contains. Parts of working memory are highly auditory—we can hear what's going on there. When we want to remember a phone number or the items we have to pick up at the supermarket, we often keep the information active in working memory by repeating it over and over to ourselves. More important, in solving problems, or even doing daily tasks, we often silently tell ourselves what to do as we go along. While we cook, solve crossword puzzles, or do geometry problems, we silently talk our way to a solution.

Protocol analysis exploits this "talking to ourselves" feature of working memory. To collect fine-grained, moment-by-moment data on a subject's cognitive processing, researchers have the subject "think aloud" while solving a problem. They instruct the subject to say everything he is thinking while engaged in a task. They then transcribe and analyze the verbal protocols. Often, the protocols provide data for computer simulations of problem-solving behavior.

Protocol analysis is a fundamental method of cognitive research. Alan Schoenfeld (1987, p. 1) admits that "spending 100 hours analyzing a single 1-hour videotape for a problem-solving session, and perhaps 2 or 3 years writing computer programs that 'simulate' the behavior that appeared in that 1 hour of problem solving, must appear odd to someone looking from outside the discipline." However, this often meticulous obsession with detail is what gives cognitive research its unique strengths and sets it apart from earlier attempts to understand mental functioning.

To find out why the 8-year-olds learned and the 5-year-olds didn't, Siegler and his collaborators selected several children between 5 and 10 years old for in-depth study (Klahr and Siegler 1978). Each child had a training session with the balance scale that included conflict problems. In the training session the child was asked to make a prediction for each problem and to state his or her reasons for the

prediction. The experimenter then unlocked the scale's a
child observed the result. If the prediction was not bor
experimenter asked "Why do you think that happened
searchers videotaped the entire session with each child and
all the children's verbal responses.

Lisa, a typical 5-year-old, took 30 minutes to do 16
Protocols like Lisa's suggested that the younger childre
encoding or representing weight in their initial interpr
balance-scale problems. For example, when Lisa was give
problem (on the left side, one weight on peg 3; on the
one weight on peg 1), she predicted the scale would balan
would both stay up," she said. Asked why she though
answered "'Cause they are both the same." When she s
side tip down, she was genuinely puzzled: "Well, why are
the same thing and one's up and one's down?" Lisa did n
difference between the two sides. She was not includin
information in her initial representation of the problem. S
did not notice and encode distance information.

An 8-year-old's protocol gave very different data. Jan
a conflict-distance problem: on the left side, three weig
first peg; on the right side, two weights on the third
predicted incorrectly that the left side would go down. Wh
what really happens (right side down) and asked for an ex
she gave one involving both weight and distance. For h
and 2 on each side were "near" the fulcrum and pegs 3 a
"far" from the fulcrum. She stated a rule: "If far pegs hav
then that side will go down." She then pointed out th
problem the far pegs on the right side had weights but th
on the left had none, so the right side would go down. Ja
a perfect explanation, nor is her rule always true. Her
shows, though, that she, unlike Lisa, had noticed and enc
weight and distance information in her representation of the

On the basis of the protocols, the difference betwee
olds and 8-year-olds seemed to be that the younger childre
problems in terms of weight only, whereas the older child
see the problems in terms of weight at a distance from the
If the younger children were not encoding distance, they
learn from training on conflict problems that differences in
sometimes overcome differences in weight. They could no
the concepts or build the chunks they needed for the conc

P4 and P5 in rule III. On the other hand, the older children, even if they were using rule I, appeared to encode distance. They could learn from training on conflict problems how to use that information to build new productions and progress to rule III.

Protocol analysis is so detailed and time consuming that researchers usually do it on only a few subjects, but these detailed analyses often suggest hypotheses that can be tested on larger groups of subjects. That is what happened here. To test possible encoding differences between 5-year-olds and 8-year-olds, Siegler presented each child with 16 problems, one at a time. The child saw the pattern of weights for 10 seconds, after which the experimenter hid the scale from sight. The experimenter then asked the child to recreate the configuration, or "make the same problem" from memory, by placing weights on the pegs of a second, identical scale. The results confirmed the hypothesis suggested by the protocol analysis. Five-year-olds were much more accurate in encoding weight than distance. They reproduced weight information correctly 51 percent of the time, but distance information correctly only 16 percent of the time. Eight-year-olds were more highly accurate on both dimensions: 73 percent correct for weight and 56 percent correct for distance information. The 5-year-old rule I users weren't encoding enough about the problem into their representations to benefit from training on conflict problems. The 8-year-old rule I users, on the other hand, were encoding information about distance that they were not using spontaneously to make predictions about the balance scale's actions. Training on conflict problems prompted the older children to see the relevance of the distance information, incorporate it into a condition, and build new and better productions.

Can 5-year-olds learn to encode both weight and distance, or is it beyond their level of cognitive development? Siegler found that giving 5-year-olds more time to study the configurations or giving them more explicit instructions ("See how the weights are on the pegs? See how many are on each side and how far they are from the center on each side?") made no difference in their ability to reproduce the configurations from memory.

Only one intervention seemed to work. *The 5-year-olds had to be told explicitly what to encode and how to encode it.* The instructor had to tell them what was important and teach them a strategy for remembering it. The instructor taught the children to count the disks on the left side, count the pegs on the left side, and then rehearse the

result (i.e., say aloud "three weights on peg 4"); to repeat t
for the right side; and then to rehearse both results togeth
weights on peg 4 and two weights on peg 3"). The instru
told the children to try to reproduce the pattern their
described. The instructor guided each child through this st
seven problems. With each problem, the children took mo
sibility for executing the strategy.

After this training, the 5-year-olds' performance on re
ing distance information from memory improved. They
rectly reproduced weight information 52 percent of the
distance information 51 percent of the time. Although t
apparently encoded the information, they, like the 8-year-
users, did not spontaneously start using it. They continu
rule I. However, when these 5-year-olds were given tra
conflict problems, they too progressed from rule I to rule
had to be taught explicitly what representation to use in
learn from the training experience.

The results of this study exemplify features of learning
common to almost all school subjects. Students learn by m
long-term memory structures, here called production systen
modify their structures when they encounter problems thei
rules can't solve. Some children modify their structures
neously; that is how children normally develop through
four rules. But by giving appropriate training we can facili
dren's development. For some children, presenting anomalo
lems is enough. Like the 8-year-olds confronted with
problems, some children can build better rules when challeng
hard problems. Other children can't. Some children have ina
initial representations of the problem. Children have to no
information they need and encode it if they are to build bett

We have seen, too, how long-term memory structures,
schemas and production systems, can influence what we no
call, and remember. The existing rules and the initial represe
affect one another. Effective instruction must break into and
this interaction. Breaking into and changing the interactio
requires detailed, explicit instruction on what the initial rep
tion should be. Often this instruction also has to include teac
effective strategy for encoding and remembering. Students wh
learn spontaneously from new experiences need direct inst
about the relevant facts *and* about the strategies to use. Teachi

facts or teaching strategies in isolation from the facts won't work. The difficulties children have in learning about the balance scale are, as we shall see, highly similar to the difficulties they encounter in learning mathematics, science, and literacy skills.

From Rule III to Rule IV

The transition from rule III to rule IV is also of educational interest. Siegler almost did not include 17-year-olds in his original study. The principal of the school where he was doing the research assured him that high school juniors and seniors had already studied the balance scale and the law of torques at least twice and "knew all there was to know about it." Siegler tested the high school students anyway. Much to everyone's surprise, only 10 percent of the high school students used rule IV spontaneously and only 20 percent discovered it after a training session. This suggests that rule IV is deceptively difficult, that high school science education is inadequate, or (most likely) both.

What kind of instruction or training sessions might help older students learn rule IV? On the basis of task analysis and how the balance scale works, Siegler conjectured that there were at least two points where students might have trouble: they might not realize that balance-scale problems have quantitative, mathematical solutions; and, even if they did, they might have trouble figuring out which algebraic equation to apply to the four variables to find and compare torques. To address the first point, training should emphasize the quantitative nature of the task. Rather than just asking on each training problem "What do you think will happen?" one should say, for example, "Three weights on the third peg versus two weights on the fourth peg; what do you think will happen?" The second point could be addressed by giving the student an external memory aid. Each time the experimenter presents a training problem, he could give the student a diagram of the problem on a piece of paper. The student could keep the diagrams and refer to them during the training session. Then, when the student developed a hypothesis about a possible equation, he or she could check the hypothesis using the data from all the previous problems as shown on the diagrams.

In an experiment, Siegler gave 13- and 17-year-olds training experiences that included hints on quantitative encoding, or the ex-

ternal memory aid for hypothesis checking, or both. Mos
olds progressed to rule IV with the help of either the enco
or the external memory aid. The 13-year-olds progressed
only if they had both kinds of help. The 17-year-olds also
last error much earlier in the training session. Cognitively
training helped students in both age groups learn rule I
13-year-olds needed more help and learned more slowly. T
instructional manipulation helped high school students le
they had failed to master on two previous occasions in th
science instruction.

Cognitive Research and Effective Instruction

In a summary of their work with the balance-scale task, Si
Klahr (1982, p. 197) conclude that their results "show that a
of new knowledge depends in predictable ways upon the i
of existing knowledge, encoding processes, and the ins
environment." Their summary, like their work, contains a
ments that make cognitive research applicable to educational
The work builds on and supports the assumption that hu
computers, are symbol processors. Task analysis, protocol
response-time studies, and training studies reveal how our
architecture works in solving problems.

The methods tell us about our psychology. They tell
how we organize knowledge in long-term memory and
knowledge interacts with our initial problem representa
working memory to guide our behavior. The research sugg
we learn by modifying existing memory structures, such as
tion systems. Learning can occur when new situations chall
current theories. In some cases, though, we can learn fr
experiences only if we receive explicit instruction about
represent, or interpret, those experiences.

We want our educational system to help children—
novices—become reasonably expert within certain dom
knowledge. To do this effectively, we have to know, in som
what stages children pass through on their mental journe
novice to expert. Cognitive science tells us how we can th
children progress from relative naiveté through a series o
understandings to eventual subject mastery.

Earlier in this chapter, we found it helpful to think of basic cognitive research as trying to solve an equation. If we have values for task demands and observed behavior, we can solve for the subjects' psychology; or with task analysis and psychological hypotheses we can, as Siegler did, predict behavior. Having data for any two variables lets us solve for the third. When we want to apply cognitive psychology to the design of effective instruction, we take the psychology and the behaviors as known and try to solve for the task-demand variable. We want to use what basic research tells us about our cognitive psychology, plus information about the behaviors or competencies we want students to have, to "solve for" (or generate hypotheses about) effective instructional tasks and learning environments. In view of what we know about children's cognitive psychology, what kinds of task environments will result in the expert behaviors we want? Those task environments are the instructional environments we should provide in our schools.

Siegler's work shows how cognitive science "provides an empirically based technology for determining people's existing knowledge, for specifying the form of likely future knowledge states, and for choosing the types of problems that lead from present to future knowledge" (Siegler and Klahr 1982, p. 134). The following chapters describe how researchers and teachers are applying this technology to improve classroom instruction. The tasks, representations, and production systems will become more complex—the progression from novice to expert can't be captured by four rules in every domain. However, our innate cognitive architecture remains the same no matter what domain we try to master, and the methods of cognitive science yield detailed information about how we think and learn. The lessons learned on the simple balance scale apply across the curriculum.

3

Intelligent Novices: Knowing How to Le

Imagine that a small, peaceful country is being threatened b
belligerent neighbor. The small country is unprepared hi
temperamentally, and militarily to defend itself; howev
among its citizens the world's reigning chess champion. T
minister decides that his country's only chance is to outv
gressive neighbor. Reasoning that the chess champion is a fo
strategic thinker and a deft tactician—a highly intelligen
skilled problem solver—the prime minister asks him to as
sponsibility for defending the country. Can the chess cham
his country from invasion?

This scenario is not a plot from a Franz Lehar operet
thought experiment devised by David Perkins and Gavriel
(1989). As they point out, our predictions about the chess ch
performance as national security chief depend on what w
intelligence and expertise are. If the goal of education is to
our children into intelligent subject-matter experts, our pr
about the chess champion, based on what we believe abou
gence and expertise, have implications for what we should c
schools.

Since the mid 1950s cognitive science has contribute
formulation and evolution of theories of intelligence, and s
understanding of what causes skilled cognitive performance a
should be taught in schools. In this chapter, we will review h
understanding of intelligence and expertise has evolved over
two decades and see how these theories have influenced edu
policy and practice.

Four theories will figure in this story.

The oldest theory maintains that a student builds up hi
intellect by mastering formal disciplines, such as Latin, Greek

and maybe chess. These subjects build minds as barbells build muscles. On this theory the chess champion might succeed in the national security field. If this theory is correct, these formal disciplines should figure centrally in school instruction.

In the early years of the cognitive revolution, it appeared that general skills and reasoning abilities might be at the heart of human intelligence and skilled performance. If this is so, again the chess champion might succeed, and schools should teach these general thinking and problem-solving skills—maybe even in separate critical-thinking and study-skills classes.

By the mid 1970s, cognitive research suggested that general domain-independent skills couldn't adequately account for human expertise. Researchers then began to think that the key to intelligence in a domain was extensive experience with and knowledge about that domain. Expertise was domain specific. This suggested that the chess expert was doomed to failure, and that schools should teach the knowledge, skills, and representations needed to solve problems within specific domains.

In the early 1980s researchers turned their attention to other apparent features of expert performance. They noticed that there were intelligent novices—people who learned new fields and solved novel problems more expertly than most, regardless of how much domain-specific knowledge they possessed. Intelligent novices controlled and monitored their thought processes and made use of general, domain-independent strategies and skills where appropriate. This suggested that there was more to expert performance than just domain-specific knowledge and skills.

Perkins and Salomon call this latest theory or view the "new synthesis," because it incorporates what was correct about the earlier views, while pointing out that none of the earlier theories alone provides an adequate basis for effective educational practice. According to the new synthesis, we should combine the learning of domain-specific subject matter with the learning of general thinking skills, while also making sure that children learn to monitor and control their thinking and learning.

The new synthesis introduces an important new idea into discussions about educational reform. The first three theories of intelligence emphasize *what* we should teach in our schools—formal disciplines, general thinking and learning skills, or domain-specific knowledge and skills. The new synthesis, as we shall see, implies

that we should be as concerned with *how* we teach as we traditionally have been concerned with what we teach. The most recent research shows that if we can apply the new synthesis in the classroom, we should be able to teach school subjects as high-order cognitive skills and help children become intelligent novices and expert learners.

Transfer

What connects the chess champion, theories of intelligence, and schooling is a phenomenon psychologists and educators call *transfer*. We generally believe that learning a certain skill or subject area can help us learn a related one. If we first learn tennis, we should be able to learn squash more easily. If we learn Spanish as a second language, we should be able to learn Italian as a third language more easily. Knowledge from the first skill or domain should transfer to the second, so there is less to learn. Notice, though, that in neither of these examples are we simply applying previously learned knowledge. Squash isn't tennis, and Italian isn't Spanish. In these situations, we are using old skills or knowledge in novel situations where we also have to learn new things. One cognitive scientist describes it this way: "Transfer means applying old knowledge in a setting sufficiently novel that it also requires learning new knowledge." (Larkin 1989, p. 283)

If this description is correct, we should be able to tell when transfer occurs. If knowledge transfers from task A to task B, then people who have learned A should be able to learn B more rapidly than people who did not first learn A. A tennis player should be able to learn squash more rapidly than a person with no prior experience at racquet sports.

Transfer is central to designing and developing effective instruction. Problems of transfer pervade schooling. Teachers want to teach lessons so that students can transfer what they have learned during class instruction to solve new problems at the end of a chapter. We want that learning to transfer to the unit, semester, or standardized test. Most important, we want school learning to transfer to real-world problem solving at home and on the job. If this is our goal, what and how should we teach?

If we want to teach so as to promote transfer of knowledge, we have to answer a prior question: What kinds of knowledge and skills, if any, transfer between tasks? What, if anything, might transfer from chess to national security? Theories of intelligence and expertise suggest answers to this question. Theories differ in their claims about what, whether, and when knowledge transfers from one task or knowledge domain to another. Of the theories outlined above, the first says that general mental strength transfers, the second that general skills and strategies transfer, and the third that expertise is domain specific (so that we might find some transfer within a domain, but little or none across domains). The new synthesis suggests that transfer can occur within and across domains, but only if we teach students appropriately.

Formal Disciplines and Mental Fitness

Our oldest theory of expertise and intelligence goes back to the classical Greeks, who believed that mastering formal disciplines, such as arithmetic and geometry, would improve general intelligence and reasoning ability. By the eighteenth century, scholars had added grammar, mnemonics, Greek, and Latin to the list of disciplines that build mental fitness. The theory was that these difficult formal disciplines would build general mental strength, just as rigorous physical exercise builds physical strength. On this theory, if we believe that chess is a formal discipline on a par with logic and geometry, we might favor the chess master's chances.

Edward Thorndike's careful studies of learning and of what knowledge transfers from one subject to another were among scientific psychology's first contributions to education. (See Thorndike and Woodworth 1901.) At the turn of the twentieth century, when Thorndike did his work, the prevailing view, derived from the ancient Greeks, was that learning formal disciplines improved general mental functioning. Thorndike, however, noted that no one had presented scientific evidence to support this view. Thorndike reasoned that if learning Latin strengthens general mental functioning, then students who had learned Latin should be able to learn other subjects more quickly. He found no evidence of this. Having learned one formal discipline did not result in more efficient learning in other domains. Mental "strength" in one domain didn't transfer to mental strength in others. Thorndike's results contributed to the demise of this ancient theory of intelligence and to a decline in the teaching of formal disciplines as mental calisthenics.

However, in some experiments where two subject domains shared surface similarity, Thorndike did observe faster learning in the second domain. He proposed a theory of "identical elements" to explain this. Thorndike suggested that where two domains share common *elements of knowledge*—not formal rigor—a person who has learned one of them might be able to learn the second more quickly. But because psychologists at the turn of the century had no precise way to describe and identify "elements," Thorndike couldn't test his theory rigorously. The methods he needed were those that cognitive psychologists developed more than 50 years later.

Elements, Productions, and Transfer

Once psychologists accepted the assumption that our minds process symbols, and once they realized they could study minds as information-processing devices, it became possible to test theories such as Thorndike's. Psychologists, using the framework of computational theory, could describe "elements" as symbol structures and devise problem-solving simulations and experiments to see which symbol structures two disciplines might share.

Production systems are among the things that allow psychologists to test modern versions of Thorndike's theory. If minds are devices that execute production systems, and if (as on the balance scale) learning occurs when we add new productions to long-term memory, then we might be able to formulate and test Thorndike's claim. We can think of each individual production rule as a piece of knowledge needed for a task; we can think of it as one of Thorndike's elements. If so, the transfer of learning from one task to another should be directly related to the number of productions the tasks share.

M. K ngley and John R. Anderson (1985) performed an elegant study to test this hypothesis. They studied the way in which secretarial students learned to use three different text editors or word processors. Two of the editors, ED and EDT, were line editors that allowed the user to edit one line of text at a time. EMACS was a screen editor, more like a standard word processor, that allowed the user to edit a document one screen at a time.

As is typical of cognitive scientists, Singley and Anderson first did a careful task analysis of the three editors. The two screen editors used different names for the editing commands and differed in how the user located a line to edit. Once past these superficial differences, however, they found that the production systems used to simulate expert performance on the two line editors were nearly identical. However, the production system that simulated expert performance on EMACS, the screen editor, was almost entirely different from those for the line editors. Thus, there was considerable production overlap between the line editors and almost none between the line editors and the screen editor.

How did this affect learning? Students who learned either line editor first took as long to learn the screen editor as students who started out on the screen editor. Skill on the line editors didn't transfer to the screen editor. In the case of the two line editors, students who learned one learned the second much more quickly. There was considerable transfer between the two line editors. Anderson (1985, p. 241) estimates that learning one of the line editors eliminated up to 90 percent of the work normally needed to learn the second. Singley and Anderson concluded that the amount of production overlap between two skills predicts the amount of transfer between the skills.

Relying on computational theory, production systems, and task analysis allowed cognitive science to make precise scientific sense of Thorndike's hypothesis. The information-processing approach can give us fine-grained representations—in this case, productions—of Thorndike's common elements. Cognitive research gives us methods for stating and testing claims about the transfer of knowledge between tasks.

General Methods and Intelligent Behavior

Cognitive scientists started applying computational insights to issues of expertise, intelligence, and transfer in the late 1950s. To understand their initial approach, recall the model of problem solving presented in chapter 2. As we saw there, problems have initial states and goal states. The solver chooses operators that create a chain of knowledge states linking the initial state to the goal. The operators, themselves composed of basic information processes, combine to form procedures or programs that guide problem-solving behavior.

How do we choose operators when solving problems? Cognitive scientists say that we use *methods* or *strategies* to choose them. Imagine we are playing chess. One method I might use to choose a move,

or an operator, is to pick a piece at random or move the first piece I happen to touch. At the other extreme, I might choose my moves by following an opening I have studied in a chess book. An intermediate option would be to use a method based on general chess principles: I might choose my moves so as to control the center, defend my pieces, and attack yours. The same spectrum of methods or strategies is available for balance-scale problems. I might generate a prediction by randomly choosing among left, right, and balance. At the other extreme, I might use Siegler's rule IV, having studied it in a book.

These methods differ in ways that are more interesting than how often I would win the game or make a correct balance-scale prediction. First, they differ in how widely applicable they are. Second, they differ in what I have to know to use them. The method of choosing an operator at random works for any problem: I don't have to know anything about chess or balance scales to use random choice. In contrast, following a line from a chess book or using rule IV works only for chess or balance-scale problems. Furthermore, I have to know a lot about chess or balance scales to use a book or Siegler's rule. Cognitive psychologists call methods that are widely applicable and that require little or no specific knowledge *weak methods*. They call methods that are situation-specific and domain-specific *strong methods*. Random choice is a weak method; rule IV is a strong one. Psychologists see all strategies, procedures, and skills as falling somewhere on the continuum between weak and strong methods.

In the early days of computer and cognitive science, there were divergent views about how to make computers or people more intelligent. Some thought the key to understanding intelligent behavior, for both machines and humans, lay in developing and understanding weak methods that were applicable across many problem domains. Others thought the better scientific bet was to study the knowledge needed in specific domains and find the specific strong methods that experts used.

Initial successes in artificial intelligence research suggested that weak methods were the way to go. Logic Theorist could prove logical theorems. A second-generation machine, called General Problem Solver, could solve problems in a variety of domains, including algebra, geometry, and chess (Ernst and Newell 1969). These programs used weak methods such as *hill climbing* and *means-end analysis*.

Hill climbing is a weak method that chooses intelligent next moves on a problem if the problem requires progress along a single dimension. If you were trying to find the top of a hill in the dark, you would keep taking steps that tended in an upward direction. When you couldn't take any more upward steps, you would stop, assuming you had reached the top of the hill. The children's game of helping a playmate find a hidden object by giving clues of "hotter" and "colder" as the playmate moves toward or away from the object is a hill-climbing game.

Means-end analysis, the method General Problem Solver used, is more complex. Hill climbing considers only one difference between the current and goal states—in the children's game, all that matters is distance from the hidden object. Means-end analysis identifies several differences between the current situation and the goal, then picks an action or an operator that will reduce one or more of those differences. If more than one action or operator could be used, means-end analysis chooses the one whose conditions of applicability best match the current situation. Sometimes, after choosing the action best suited to the situation, one still can't execute the action because the conditions aren't right. In this case, means-end analysis establishes a subgoal to create conditions that permit the chosen action.

The wide applicability of means-end analysis is suggested by the example Newell and Simon (1972, p. 416) used:

> I want to take my son to nursery school. What is the difference between what I have and what I want? One of distance. What changes distance? My automobile. My automobile won't work. What is needed to make it work? A new battery. What has new batteries? An auto repair shop. I want the shop to put in a new battery; but the shop doesn't know I need one. What is the difficulty? One of communication. What allows communication? A telephone . . . and so on.

The solver here looks at where he is and where he wants to be, then works back and forth between the ends and the means to achieve those ends until he has a set of actions, or operators, that achieve the goal. His finding a telephone starts a chain of events that results in his son's arriving at nursery school. Students often use means-end analysis to solve school math and science problems.

In the late 1960s and the early 1970s, programs such as General Problem Solver suggested that general skills might be fundamental

to human expertise and intelligence. If we could identify and teach such general skills, maybe we could improve human problem-solving performance both in and out of school. If general methods, skills, and strategies are the basis of the chess champion's expertise, maybe he can succeed at solving diplomatic problems.

Experts' Domain-Specific Knowledge

Research to extend the initial insight about the contribution of weak methods to human intelligence met with frustration. The research community soon realized there was a serious limitation on the early successes. The early AI programs, including General Problem Solver, simulated intelligent performance on games or formal, logical problems (e.g., solving number puzzles or proving logical theorems)—but playing games and solving formal problems demand little factual knowledge about the world. To succeed at these tasks requires knowledge of little more than the rules of the game or the rules of the formal system. However, problem solving in other domains, such as physics or medical diagnosis, requires considerably more factual knowledge. Just think about what you have to know to play tic-tac-toe versus what you have to know to solve a physics problem. When cognitive scientists started to explore some of these other more knowledge-rich areas, physics and medicine among them, they found that weak methods didn't work so well. In knowledge-rich domains, they found that strong methods tailor-made to work on specific, well-organized knowledge bases worked better.

Weak methods might work in domains where there is little factual knowledge, but one can't generalize from these puzzle domains to a general theory of human intelligence and expertise. As Marvin Minsky and Seymour Papert (1974, p. 59) wrote, "It is by no means obvious that very smart people are that way directly because of the superior power of their general methods—as compared with average people." Maybe expert, intelligent behavior depends crucially on the knowledge people have, how they organize it, and the specific methods they learn or develop to process it.

Data from psychological experiments also undermined the primacy of weak, general methods for human expertise. One of the most influential experiments was William Chase and Herb Simon's (1973) study of novice and expert chess players, which followed on earlier work by A. D. De Groot (1965). Chase and Simon showed positions from *actual* chess games to subjects for 5 to 10 seconds and asked the subjects to reproduce the positions from memory. Each position contained 25 chess pieces. Expert players could accurately place 90 percent of the pieces, novices only 20 percent. Chase and Simon then had the subjects repeat the experiment, but this time the "positions" consisted of 25 pieces placed randomly on the board. These were generally not positions that would occur in an actual game. The experts were no better than the novices at reproducing the random positions: both experts and novices could place only five or six pieces correctly.

Other researchers replicated the Chase-Simon experiment in a variety of domains, using children, college students, and adults. The results were always the same: Experts had better memories for items in their area of expertise, but not for items in general. This shows, first, that mastering a mentally demanding game does not improve mental strength in general. The improved memory performance is domain specific. Chess isn't analogous to a barbell for the mind. Second, it shows that if memory *strategies* account for the expert's improved memory capacity, the strategies aren't general strategies or weak methods. Chess experts have better memories for genuine chess positions, but not for random patterns of chess pieces or for strings of words or digits. Thus, experts aren't using some general memory strategy that transfers from chess positions to random patterns of pieces or to digit strings.

From long experience at the game, chess experts have developed an extensive knowledge base of perceptual patterns, or chunks. Cognitive scientists estimate that chess experts learn about 50,000 chunks, and that it takes about 10 years to learn them. Chunking explains the difference between novice and expert performance. Reproducing chessboard configurations after 5 to 10 seconds of study is a working-memory task, because there is not enough time to code and store the information in long-term memory. When doing this task, novices see the chessboard in terms of individual pieces. They can store only the positions of five or six pieces in working memory—numbers close to what we found our own working memory spans to be on the basis of the experiment discussed in chapter 2. Experts see "chunks," or patterns, of several pieces. If each chunk contains four or five pieces and if the expert can hold five such chunks in working memory, then the expert can reproduce accurately the positions of 20 to 25 individual pieces. Chase and Simon even found that when

experts reproduced the positions on the board, they did it in chunks. They rapidly placed four or five pieces, then paused before reproducing the next chunk.

Expertise, these studies suggest, depends on highly organized, domain-specific knowledge that can arise only after extensive experience and practice in the domain. Siegler's balance-scale study (chapter 2) is another example. Under normal conditions, it takes a child at least 17 years to become expert at balance-scale problems. More knowledge about and experience with the balance scale results in more sophisticated, expert-like performance. Chunking helps children develop more complex rules that contribute to their growing expertise on the balance-scale task.

Other studies of problem solving also argue against general strategies. Try to solve the two problems illustrated in figure 3.1. Philip Johnson-Laird (1983, p. 30) found an interesting difference in individuals' abilities to solve them. This is interesting because formally, or logically, they are the same problem. The same general strategy or formal rule solves both.

The correct answers are "E and 7" and "Manchester and car." Many people answer, incorrectly, that they have to turn only E, or else E and 4, in the first problem. You do have to turn E, because if that card has an odd number on the other side the rule is false. You don't have to turn 4, because even if that card had a consonant on the other side it doesn't matter; the rule doesn't say anything about what is on the other side of a consonant card. You have to turn 7, because a vowel on the other side of that card would make the rule false. The same problem-solving strategy works for the second problem, and for any "if-then" rule as logicians interpret such statements. (According to the laws of logic, an "if-then" statement is false only in the case where the "if" clause is true and the "then" clause is false; in every other case the statement is true.)

The two problems in figure 3.1 differ only in their subject matter. The first problem is an abstract one about letters and numbers, but the second one deals with a possible real-life situation. Johnson-Laird's subjects were much better at the second problem. Only 12 percent of them said they would turn over the "7" card to test the first rule, but over 60 percent said they would turn over the "car" card to test the second rule. Furthermore, he found that giving subjects experience with real-life if-then problems didn't improve their performance on more abstract versions. Apparently, most of

1. Each of the following four cards has a number on one side and a letter on the other side. Pick the cards you have to turn over to find out if this general rule is true or false:

If a card has a vowel on one side then it has an even number on the other side.

E	K	4	7

2. Each of the following cards has a destination on one side and a mode of transport on the other side. Pick the cards you have to turn over to find out if this statement is true or false:

Every time I go to Manchester I travel by train.

Manchester	Sheffield	Train	Car

Figure 3.1
These two problems are logically the same; they differ only in their content. (From Johnson-Laird 1983.)

us don't use a general rule or strategy to solve these problems. If we did, we would use the same rule to solve all such problems. There would be no difference between performance on number-letter problems and on destination-transportation problems. In other experiments, researchers report that the ability to transfer a solution from one version of a problem to another occurs only if the experimenter *explicitly tells* subjects that the two problems are the same. For some reason, what the problem is about and how familiar we are with that content affects our problem solving. It seems as if domain-specific knowledge *does* contribute to expert performance.

It also appears that the ability to use general strategies at all depends on the subject's having a knowledge base on which the

strategies can work. When an experimenter asks subjects to memorize lists of words (e.g., "dog, gold, carrots, diamond, cat, peas"), subjects rarely repeat the words in that order. Usually, subjects say something like "dog, cat, carrots, peas, gold, diamond." To remember the words, subjects group them into meaningful categories—here animal, vegetable, and mineral. Psychologists call this often-unconscious strategy *clustering*. Clustering helps us remember things by exploiting the schema structure of long-term memory; we remember the words by associating them with the appropriate schemas.

When college students and young children were the subjects in such experiments, psychologists found that the college students recalled more words and did more clustering. Initially, psychologists attributed young children's poor performance to their inability to use the clustering strategy. Later it was discovered that if the word list included things young children know more about than college students, the results would be different (Lindberg 1980). If the experimenter used a 30-word list that included names of children's television celebrities, cartoon stars, and comic book characters, young children recalled more and used more clustering than the college students. Thus, there is an interaction between knowledge and strategy use—between facts and skills. Subjects are more likely to use a general memory strategy the more they know about a domain or a topic. Strategies can help us process knowledge, but first we have to have the knowledge to process.

While granting the possibility that strategies might play some role in problem solving, by the mid 1970s many cognitive scientists had come to believe that domain-specific knowledge and strong methods are the bases of expertise and intelligence—that the chess champion would fail to deter the belligerent neighbor nation. If they are right, there may not be a simple way to make people better general problem solvers. Siegler (1985, p. 184) sums up what this means for education: "Seen from this perspective, much of the task of education in problem solving may be to identify the encoding that we would like people to have on specific problems, and then to devise instructional methods to help them attain it." In other words, the educational challenge might be to identify the representations we want students to have in *specific domains* and then develop methods and curricula to teach those representations.

Weak Methods in the Schools

Despite this theoretical shift within the cognitive science community from favoring general strategies to arguing for domain-specific knowledge and skills, some educators still advocate teaching weak, general strategies. If children need thinking and learning skills but don't have them, they argue, the best educational strategy might be to teach those skills directly. Maybe teaching weak, general methods—skills that apply across the curriculum—can serve as a shortcut to higher intelligence and better school performance. Unfortunately, weak methods as they are traditionally taught seem to have little impact on student learning.

Traditional study and learning skills, though less general than means-end analysis, are general, weak methods. These skills are often the staples of junior high language-arts classes—taking notes, outlining, underlining, and figuring out words from context. We all believe that these strategies work, but research on traditional study skills shows that these skills are no more effective than simply reading and rereading a text. (See Anderson 1980.) Other researchers who have looked at the impact of teaching general reading-comprehension skills have found that teaching these skills increases students' awareness of the skills. However, even when students say they use the skills, the skills have little effect on their reading comprehension (Paris and Jacobs 1984). Similarly, the teaching of study or memorization skills in one subject has no impact on students' performance in other subjects. Often, it doesn't even occur to students to use the skills when studying a different subject. Typically, children will use strategies immediately after instruction, but will not use them on later occasions unless explicitly told to do so by the instructor.

Research shows that either the teaching of traditional study skills has no impact on learning or else the skills fail to transfer from the learning context to other situations. Either way, teaching these general skills is not the path to expertise and enhanced academic performance.

A wide variety of books and commercially available courses attempt to teach general cognitive and thinking skills. (For reviews and evaluations see Nickerson et al. 1985, Segal et al. 1985, and Chipman et al. 1985.) Analysis and evaluation of these programs again fail to support the belief that the teaching of general skills enhances students' overall performance.

Most of these programs teach general skills in stand-alone courses, separate from subject-matter instruction. The assumption is that students would find it too difficult to learn how to think and to learn subject content simultaneously. Like the early AI and cognitive science that inspire them, the courses contain many formal problems, logical puzzles, and games. The assumption is that the general, weak methods that work on these problems will work on problems in all subject domains.

A few of these programs, such as the Productive Thinking Program (Covington 1985) and Instrumental Enrichment (Feuerstein et al. 1985), have undergone extensive evaluation. The evaluations consistently report that students improve on problems like those contained in the course materials but show only limited improvement on novel problems or problems unlike those in the materials (Mansfield et al. 1978; Savell et al. 1986). The programs provide extensive practice on the specific kinds of problems that their designers want children to master. Children do improve on those problems, but this is different from developing *general* cognitive skills. After reviewing the effectiveness of several thinking-skills programs, one group of psychologists concluded that "there is no strong evidence that students in any of these thinking-skills programs improved in tasks that were dissimilar to those already explicitly practiced" (Bransford et al. 1985, p. 202). Students in the programs don't become more intelligent generally; the general problem-solving and thinking skills they learn do not transfer to novel problems. Rather, the programs help students become experts in the domain of puzzle problems.

The evaluations of these programs undercut the basic assumption about the power of weak methods in another way, too. If general skills, or weak methods, are the stuff of intelligence, then teaching these skills to students who had not previously used them should improve their performance. This doesn't happen. The programs don't help all students who were initially naive about the general skills taught. Typically, these programs help low-performing students most, average students some, and more able students hardly at all (Nickerson et al. 1985, p. 325).

Although we should not dismiss approaches that might help low-achieving students, this inverse pattern—low achievers benefiting most and high achievers hardly at all—is exactly what we would expect if school performance depends on domain-specific knowledge and strong methods. Low-performing students have neither general cognitive skills nor domain-specific knowledge. Teaching low achievers general skills can only help. The higher the level of initial performance, though, the more domain-specific knowledge a child has. If you have domain-specific knowledge and strong methods to go with it, why use weak ones? If you know all the standard variations on the Queen's Gambit Declined, why choose chess moves at random; why even rely on general chess principles? Teaching general cognitive skills to able students (even able students who haven't heard of those skills) doesn't improve their performance, because they are already relative experts. Able students already have domain-specific knowledge and use strong methods.

Evidence from the laboratory and the classroom argues against a fundamental role for weak methods and general skills in expertise and learning. Weak methods, in the guise of study skills, thinking-skills curricula, or critical-thinking programs, are not a short-cut to improved educational outcomes.

By the mid 1970s, then, most cognitive theorists recognized that domain-specific knowledge and strong methods were keys to expert performance and human intelligence. At that point, many would have bet against the chess champion's having a successful diplomatic career.

This message was picked up by some educators, and it has even reached the general public. E. D. Hirsch's *Cultural Literacy* (1987) is a thoughtful, sustained, and highly popular presentation of how domain-specific knowledge and skills are fundamental to literacy. Chapter 2 of Hirsch's book is an extended discussion of how cognitive research supports this educational philosophy. According to Hirsch, the research should make us skeptical of attempts to teach reading, writing, and critical thinking as general cognitive skills applicable to novel problems. Skilled performance in these subjects, like skilled performance in Simon and Chase's chess studies, demands an extensive store of domain-specific knowledge. "General programs contrived to teach general skills are ineffective," Hirsch argues (p. 61).

Hirsch characterizes the "critical thinking" movement—an attempt to teach weak methods—as a well-intentioned program "to take children beyond the minimal basic skills required by state guidelines and to encourage the teaching of 'higher order' skills" (1987, p. 132). The danger, as he sees it, is that advocates of higher-order thinking tend to ignore the importance of "mere facts." According

to Hirsch, "we should direct our attention undeviatingly toward what schools teach" (p. 19).

There are also dangers associated with arguments, such as Hirsch's, for the primacy of domain-specific knowledge and skills. It is easy to oversimplify and misinterpret what the research means for educational practice. Certainly the research implies that we can't ignore "mere facts" in school instruction—domain knowledge is essential. But, conversely, curricula that merely transmit facts aren't desirable either. Cognitive research also implies that we have to be as concerned with how we teach as we are with what we teach. The danger with cultural literacy is embracing the what to the detriment of the how. Lists of proper nouns, such as appear in the appendix to Hirsch's book, might help outline curricular content, but they say nothing about how to teach that content effectively. Researchers have known for a long time that teaching word meanings to children can increase vocabulary knowledge, but more vocabulary knowledge doesn't necessarily improve reading comprehension. If better reading comprehension is the goal, how one teaches vocabulary matters. Similarly, current social studies texts may present the facts about geography or history, but fail to teach course content so that students have an understanding of geography or history. As we will see, how texts present the facts is vitally important.

Finally, the majority of the cognitive research Hirsch cites was done in the 1970s. But cognitive research didn't stop then. The prevailing view of 20 years ago was not the final word, nor should it necessarily guide educational practice. Research that started to appear in the early 1980s suggests that domain-specific knowledge and skills are necessary for expert performance but may not be sufficient. There is more to intelligence and expert performance than domain knowledge.

Metacognition

Around 1980, cognitive scientists introduced a new element, called *metacognition*, into discussions of intelligence and expert performance. Metacognition is the ability to think about thinking, to be consciously aware of oneself as a problem solver, and to monitor and control one's mental processing.

John Flavell, one of the developers of this notion, described metacognition as the fourth and highest level of mental activity

(Flavell and Wellman 1977). At the lowest level are the hard-wired, basic processes such as matching the contents of working memory to conditions on production rules. At the next level are things like knowing 9 × 7 = 63, being able to recall your mother's maiden name, and having command of sufficient schemas or facts to be culturally literate. At the third level are strategies, weak or strong methods, which we voluntarily and consciously use. For example, you might repeat a phone number silently to keep it active in working memory, or you might use Siegler's rule IV to solve a balance-scale problem. The fourth level is the metacognitive level—the knowledge, awareness, and control of the three lower levels. It is our conscious awareness of ourselves (and, by extension, others) as problem solvers.

Research on metacognition has shown that knowledge, awareness, and control of mental abilities develop with age and experience. As children mature, for example, they develop a much better sense of how many items they can hold in short-term memory. Four-year-old children can usually hold about three items, but will predict they can remember eight. Adolescents have a short-term memory capacity of about six items and accurately predict this capacity (Yussen and Levy 1975).

Children's performance on other memory tasks also provides evidence for metacognitive development. When experimenters give children a list of items to study and tell the children they can take as much time as they need to memorize the list, older children perform better than younger children (Flavell 1979). Although young children might lack effective memory strategies or lack background knowledge needed for the task, there is an independent metacognitive trend as well. Preschool children, after taking as much time as they want to study the list, think they have learned it completely but do very poorly when tested. In contrast, when elementary school children say they have learned the list, they can recall it accurately. The younger children don't know how to use study time effectively and have no idea if they have learned the list or not. It seems that the younger children don't know how to learn and don't know when they have learned.

Children's understanding of texts and stories shows a similar developmental trend. Even young children grasp the essential gist of a story. If given sufficient time to study a text, children at every age can remember more about it. Children younger than about 12,

though, do not use the study time effectively. They remember more about the text, but tend to remember more details or isolated ideas from the text. They don't remember more about the text's themes or about how those themes interrelate. In short, before age 12 children don't seem to know what kinds of things are important for better understanding of texts and can't direct their mental energy to those things. The younger children lack important reading comprehension strategies, or, if they have the strategies, they lack control over them. They have weaknesses at Flavell's third and fourth cognitive levels. In contrast, children of age 12 and older usually remember more of the text's important ideas after additional study. The older children know what is important in texts, have strategies for reading texts and studying that are directed at those important features, know how and when to use the strategies, and can monitor their use of them. They can control their cognitive activity—they have metacognitive skills. Ann Brown and Judy Deloache, who reported some of these results, conclude that "one main aspect of 'what develops' is metacognition—the voluntary control an individual has over his own cognitive processes," and that "the growth of metacognitive abilities underlies many of the behavioral changes that take place with development" (Brown and DeLoache 1978, p. 26).

Hirsch emphasizes the necessity of domain-specific knowledge in learning and doesn't mention metacognition explicitly. Nonetheless, the importance of metacognition is implicit in his diagnosis of literacy problems. Although domain-specific knowledge contributes to expertise in all domains, in reading (as Hirsch carefully explains) background knowledge—knowledge that goes beyond what is literally printed on the page—is crucial for comprehension. Teaching the schemas of cultural literacy is intended to give students the background knowledge needed to be culturally literate. Note that such knowledge would fall into level 2 of Flavell's taxonomy: facts stored for recall in long-term memory.

But Hirsch alludes to knowledge that literate individuals have that would fall into Flavell's fourth level: "In effective reading, one must not only call up one's own schematic associations but also *monitor* [my italics] whether they are appropriate ones shared by the wider speech community." Literate adults do this automatically, but "young children and other semi-literates do not confidently know what other members of the speech community can be expected to know." They lack "readily accessible information about what is shared by others" (p. 68).

Calling up one's one schematic associations is a level-2 cognitive process. Monitoring their appropriateness, on the other hand, is a level-4, metacognitive process. Similarly, knowing about or estimating what other members of the speech community might know is also a metacognitive task; it involves the ability to envision other people as problem solvers whose minds work similarly to one's own. This is just to say that reading comprehension involves more than extensive cultural background knowledge. Minimally, reading comprehension also requires metacognitive monitoring skills. If students lack these skills, no amount of cultural knowledge on its own can make them literate.

Metacognition and Intelligent Novices

Metacognition is an important addition to a theory of expertise and intelligence. The results of the research discussed so far—the studies of memory, learning skills, and reading—are consistent with the contention that metacognitive skills are high-order skills but domain-specific skills nonetheless. Clearly, it is possible for a person to be expert and metacognitively sophisticated in one domain but not in others. Our cultural stereotype of the absent-minded professor—a scientist or scholar who is expert and metacognitively capable in an academic domain but inept and unaware outside that academic specialty, particularly in everyday life—derives from this possibility.

Other results, though, suggest that metacognitive skills are general skills—skills that some people can apply across domains and in domains where they have little prior background knowledge. Everyday experience suggests that there are *intelligent novices:* some novices learn new domains more quickly than other novices. Research tells us that one thing that makes some novices more intelligent than others is their metacognitive skills.

As part of an experiment, John Bransford, an expert cognitive psychologist who has done work on math learning, tried to learn physics from a textbook with the help of an expert physicist. He kept a diary of his learning experiences and recorded the skills and strategies most useful to him (Brown et al. 1983). Among the things he listed were (1) awareness of the difference between understanding and memorizing material and knowledge of which mental strategies

to use in each case; (2) ability to recognize which parts of the text were difficult, which dictated where to start reading and how much time to spend; (3) awareness of the need to take problems and examples from the text, order them randomly, and then try to solve them; (4) knowing when he didn't understand, so he could seek help from the expert; and (5) knowing when the expert's explanations solved his immediate learning problem. These are all metacognitive skills; they all involve awareness and control of the learning problem that Bransford was trying to solve. Bransford might have learned these skills originally in one domain (cognitive psychology), but he could apply them as a novice when trying to learn a second domain (physics).

This self-experiment led Bransford and his colleagues to examine in a more controlled way the differences between expert and less-skilled learners. They found that the behavior of intelligent novices contrasted markedly with that of the less skilled. Intelligent novices used many of the same strategies Bransford had used to learn physics. Less-skilled learners used few, if any, of them. The less-skilled did not always appreciate the difference between memorization and comprehension and seemed to be unaware that different learning strategies should be used in each case (Bransford et al. 1986; Bransford and Stein 1984). These students were less likely to notice whether texts were easy or difficult, and thus were less able to adjust their strategies and their study time accordingly (Bransford et al. 1982). Less-able learners were unlikely to use self-tests and self-questioning as sources of feedback to correct misconceptions and inappropriate learning strategies (Brown et al. 1983; Stein et al. 1982).

Hirsch, in his discussion of reading, notes how expert readers "monitor" their schematic associations. Monitoring comprehension is also a metacognitive skill. Ellen Markman (1985) studied this skill by having students in grades 3 through 6 read short passages which they had never seen before and which contained obvious contradictions. For example, a passage about ants might say in one place that ants navigate by leaving a chemical trail which they can smell and in another place that ants have no sense of smell. Most of the younger children and even a few of the older ones were oblivious to the inconsistencies; they weren't monitoring their comprehension. Children did improve on the task with age, so Markman first interpreted the results in terms of developmental differences between younger and older children. Subsequent research supported a more general conclusion: that the ability to apply this metacognitive skill differentiated strong from weak learners at all ages.

The ability to monitor comprehension is an essential learning skill. Often poor students are totally unaware that they don't comprehend class material. If they aren't aware that they have a learning problem, they can't take steps to overcome it.

Everyday experience suggests there are intelligent novices. Research tells us that metacognitive skills contribute to these expert learning performances. Some people develop these skills naturally; others do not. Those who do can become intelligent novices; those who don't may have difficulty learning.

Metacognition and Education

The importance of metacognition for education is that a child is, in effect, a universal novice, constantly confronted with novel learning tasks. In such a situation it would be most beneficial to be an intelligent novice. What is encouraging is that the research also shows that it is possible to teach children metacognitive skills and when to use them. If we can do this, we will be able to help children become intelligent novices; we will be able to teach them how to learn.

Just as there are basic math and reading skills, there are basic metacognitive skills. Among the basic metacognitive skills are the abilities to predict the results of one's own problem-solving actions, to check the results of one's own actions (Did it work?), to monitor one's progress toward a solution (How am I doing?), and to test how reasonable one's actions and solutions are against the larger reality (Does this make sense?). For example, a metacognitively adept chess player tries to predict the consequences of a series of moves, checks the results of those moves, and monitors whether those moves might contribute to a possible checkmate. Such a player also checks possible strategies against the larger reality. In a game against a higher-rated opponent, a metacognitively aware player would not look for an easy mating combination early in the game; a quick reality check would convince him that such a strategy doesn't make sense. Brown and DeLoache (1978, p. 15) call these skills "the basic characteristics of efficient thought." To become efficient thinkers—intelligent novices—students have to learn the skills and learn when to use them. Although a student might first learn the skills in the context of some specific subject matter (as Bransford first learned them in psychology), once he or she has learned them, the student can apply the skills in any learning situation—*if* the student has also learned that these skills are applicable and useful in any learning situation. Cognitive scientists call instruction that teaches students metacognitive skills and when to use them ***metacognitively aware* instruction**.

How does one teach like this? One can think of metacognitive skills as the ability to be critical of one's own problem solving. Metacognitively aware instruction attempts to transfer the critic's role from the teacher to the student. The transfer occurs in stages. Initially the teacher models the critic's role for the students. Gradually, the students begin to share this critical, metacognitive role with the teacher; eventually they can take on the role themselves, with the teacher standing by to provide coaching when the students falter. As the children become more metacognitively adept—more self-critical—the teacher cedes the critic's role entirely to them. Researchers describe this transition from the teacher's modeling and control to the students' control as *scaffolding*. Instruction creates a scaffold to support learning, and then the scaffold is gradually dismantled as the students become increasingly self-critical.

One problem with metacognitive skills is that they are usually covert and implicit in expert performance. To teach these skills and when to use them, the instructor has to make metacognition overt and explicit. One effective way to do this is in group learning situations where teachers and students engage in dialogues about their joint learning and problem solving. Almost all children are able conversationalists and can play the dialogue game. Appropriately guided by the teacher, the dialogue can become a social, collaborative form of "thinking aloud" in which each member of the group makes his or her thinking overt. In such situations, first the teacher and then the students describe their problem-solving strategies, present their reasoning to the group, and then defend and justify it against criticisms. These group dialogues make reasoning, planning, and monitoring public and shared. The children begin to see cognitive processes in action and understand how they can be monitored and controlled.

In experimental situations, Brown, working with various collaborators, has used metacognitively aware instruction to teach memory strategies to mildly retarded students (Brown et al. 1979), text-summarizing skills to college students (Brown et al. 1981), and analogical reasoning to children as young as 3 and 4 years (Brown 1989), all with great success. As we will see, the approach also works in the classroom, where metacognitively aware instruction can improve students' understanding of scientific reasoning, reading comprehension, and the writing process.

The Final Element: General Skills Again

The final element leading to the new synthesis is general skills—they just won't go away. Besides metacognitive ability, there is another trait that intelligent novices show. Research scientists are an extreme but illustrative example. Scientists don't just apply their extensive domain knowledge to solve standard, textbook problems; rather, they formulate new problems and discover new solutions. Deep, domain-specific knowledge is fundamental to their performance, but so are general skills and strategies. Einstein attributed his interest in Brownian motion to childhood experiences watching the patterns formed by smoke rising from his uncle's pipe. My graduate adviser, a biophysicist, would help us graduate students understand an abstract geometrical theorem about how objects can move in space by referring to it as "the hatpin-through-the-grapefruit theorem."

On the frontiers of science, or in any creative endeavor, everyone is a novice in the sense that prior knowledge is not directly applicable. Here, too, it helps to be an intelligent novice. Scientists, scholars, artists, and skilled managers all have to take what they know and stretch it to pose and answer novel problems. They have to transfer prior learning to new situations. Intelligent novices, like Einstein and my adviser, often use general strategies to do this. They use strategies such as modeling and analogy—weak methods applicable across many domains—in their attempts to apply what they know flexibly and creatively in new ways. Their extensive domain-specific knowledge interacts with general strategies to help them acquire new knowledge. General strategies do seem to have a role in intelligence and in expert performance.

General strategies should also be helpful to those universal novices, children. Although research on the teaching of such study skills as memory strategies, underlining, and note taking has shown that after instruction children do not apply these skills spontaneously, it is still hard to accept that such skills are not useful. Maybe the

problem is not with general skills but with how we have traditionally tried to teach them.

Let us review briefly what we think we know about intelligence and expertise. First, we have seen the importance of domain-specific knowledge for expert performance. The scientist's use of general strategies is based on deep understanding of at least one scientific domain. We can agree with the advocates of domain specificity that general programs contrived to teach general skills are ineffective, but that leaves open the possibility of teaching general skills within specific subject-matter instruction. General strategies do need a knowledge base on which to work, but once learned in a specific context they should be applicable in other domains. Second, both from the Johnson-Laird experiment (figure 3.1) and from research on study skills, we have seen that adults and children have difficulty transferring a skill or a strategy from one context to a similar context. In some cases, subjects could make the transfer between contexts only after the experimenter *told* them that the strategy applied in the new situation.

Perhaps, just as children have to be taught metacognitive skills and when to use them, they have to be taught general learning strategies and when to use them. Perhaps, then, previous attempts to teach general skills failed because course designers and instructors overestimated children's ability to generalize from one learning situation to another. Maybe children don't see how and why the situations are similar. In general-strategy and learning-skill instruction, rather than assume that students see the similarities between various learning situations, perhaps we should explicitly tell them how and why the situations are similar. This has led cognitive scientists to think that general-strategy instruction has a place in schools, but that strategy instruction has to be *informed*. By this they mean that strategy instruction should include explicit descriptions of the strategies, instruction about *when* the strategies are useful, and an explanation of *why* they are useful.

Paris et al. (1982) ran an experiment in which they compared informed instruction with a more traditional approach to strategy instruction. On each of the first two days of the experiment, they had 7- and 8-year-old children study sets of 24 pictures. After a period of study, they asked the children to remember as many of the pictures as they could. On average, the children could recall 12 or 13 pictures. On the third day, all the children were taught memory strategies: naming or labeling each item, sorting the items into related groups (clustering), learning the items one group at a time, and then testing themselves by trying to recall the items in groups (a metacognitive skill).

Half the children (the control group) saw the instructor demonstrate the memory strategies and were permitted to practice them, but were given no explanations of why the strategies worked and no feedback on their performance when they tried to use them. In short, the control group received traditional strategy instruction.

The other children (the experimental group) received the same instruction as the control group but in addition were told why the strategies worked and when to use them. Also, when the children in this group used a strategy they received immediate feedback on how successful they had been with it. This group received informed instruction.

Immediately after learning the strategies, children in the control group could recall on average 16 pictures and children in the experimental group 19. The experiment was continued for two more days. By the fifth day, children in the experimental group could still recall 16 items. These children continued to use the memory strategies spontaneously. Without being told, they continued to label, sort, cluster, and self-test. In contrast, by the fifth day children in the control group had fallen to the pre-instruction level of 12 to 13 items and had reverted to passive, pre-instruction learning strategies, such as looking at the pictures and trying to remember them. This experiment shows that children will use a strategy spontaneously—they will transfer it to a new situation—if they understand why it works and when it can help them learn. Informed strategy instruction works; traditional instruction doesn't.

General thinking, learning, and study strategies are important elements of intelligence and expert performance, and now it seems we may know how to teach them. According to Brown (1985, p. 335), "ideal cognitive skills training programs include practice in the specific task-appropriate strategies, direct instruction in the orchestrating, overseeing and monitoring of these skills, and information concerning the significance of those activities." Such instruction recognizes the necessity of domain-specific knowledge in that the strategies are specific, task appropriate, and integrated into subject-matter learning. The instruction is also metacognitively aware, in that the children receive direct instruction about how to

monitor and control their problem solving. It is informed in that children learn why the strategies work.

The New Synthesis and the Teaching of High-Order Skills

The new synthesis suggests that domain-specific knowledge, metacognitive skills, and general strategies are all elements of human intelligence and expert performance.

Where does this leave our chess champion? His detailed knowledge of chess won't help him in his new diplomatic career, because by definition it is domain specific. On the other hand, even if he has some potentially relevant general skills, he may not be able to transfer them readily to new problem domains—not everyone can do that. Much depends on whether the champion is an intelligent novice or not. Is his expertise narrowly confined to chess, or does he have the metacognitive insight to be an effective, rapid learner? If the latter, then maybe with some tutoring in foreign affairs and some on-the-job experience he could rapidly become a national security expert. Some chess champions—like some college professors and some school children—are naturally intelligent novices; others aren't. Whether our chess champion will succeed depends on what cognitive skills he has beyond his chess expertise.

We are just beginning to see what the new synthesis might mean for educational practice. Many of the examples in the following chapters illustrate how this latest theory of human intelligence might be applied in the classroom. For education, the most important implication of the theory is that how we teach is as important as what we teach. Domain-specific knowledge and skills are essential to expertise; however, school instruction must also be metacognitively aware and informed.

Most important, innovative classroom practices based on the new synthesis can help us achieve our goal of teaching high-order skills to all students. In chapter 1, we initially identified high-order skills as the skills that students need to achieve the higher NAEP proficiency levels. We observed that these higher proficiencies demand that students solve complex problems, for which there often are no standard solution procedures and no single correct answer. Students with high-order skills can use their knowledge flexibly to solve ill-structured, novel problems.

This characterization of high-order skills relies primarily on the kinds of problems students can solve and on students' observed behaviors. But "high-order" also refers to the thought processes needed to solve such problems and guide such behaviors. Susan Chipman, Program Manager for Cognitive Science at the Office of Naval Research, argues that "behind our choice of the term 'higher-order,' there are strong intuitions about the way in which our cognitive activities are structured and controlled" (Chipman 1992). These intuitions link high-order skills with our current theory of intelligence and expert performance.

First, Chipman points out that higher-order skills in a subject domain, such as those needed to solve ill-structured complex problems, are skills grounded in deep factual and procedural knowledge of the domain. As the new synthesis implies, high-order skills require extensive domain knowledge.

Second, she notes that students who genuinely possess high-order skills in a subject domain not only have the requisite factual and procedural knowledge, they also can recognize *when* the knowledge is applicable and can use it appropriately. It is this feature of high-order skills that accounts for the flexible, spontaneous use of knowledge in novel situations. This connects high-order skills with the notion of transfer. High-order skills should transfer from school learning to real-world situations and allow students to use what they already know to learn new things more rapidly. The key to transfer, and so to high-order skills, is knowing when to use knowledge. If we want to teach high-order skills, then, as the new synthesis says, the instruction should be informed.

Third, implicit in our use of "high-order," according to Chipman, are intuitions about how students control and monitor their cognitive skills. High-order skills, in this sense, involve awareness of what is happening in working memory, of how those processes determine eventual action, and of how to control those processes. Thus, metacognitive abilities are implicit in our notion of high-order skills. For this reason, if we want students to acquire high-order skills, instruction must be metacognitively explicit.

In short, high-order skills require extensive domain knowledge, understanding when to use the knowledge, and metacognitive monitoring and control. Students who have these things can solve novel, ambiguous problems; students who have high-order skills are intelligent novices.

For these reasons, instruction based on elements of the new synthesis is our best educational bet if we want all students to have the knowledge and skills that in past generations have been confined to the college-bound elite—if we want all students to acquire high-order skills. Educational practice grounded in cognitive theory, Lauren Resnick (1986, p. 43) writes, "would transform the whole curriculum in fundamental ways. It would treat the development of higher-order skills as the paramount goal of *all* schooling."

Transforming the curriculum to meet this goal won't be easy. We will have to rethink, or at least reevaluate, much of our received wisdom about educational policy, practice, and standards and about teacher training. We will have to restructure our schools—starting in the classrooms, where teachers interact with students. We will need teachers who can create and maintain learning environments where students can become intelligent novices. Many of us will have to change our representations of what schools and schooling are.

Admittedly, there is much we still don't know about how our minds work, how children best learn, and how to design better schools. Nonetheless, as a first step, we can start applying what we already know to improve instruction in what Resnick calls the "enabling" or "tool" domains: mathematics, science, reading, and writing. Mastery of these domains is necessary for advanced learning in more specialized subjects. We can teach these enabling domains as high-order cognitive skills to all students, as the following chapters will show.