4 Cognitive Technologies for Mathematics Education

Roy D. Pea
Educational Communication and Technology
New York University

This chapter begins with a sociohistorical perspective on the roles played by cognitive technologies as reorganizers rather than amplifiers of mind. Informed by patterns of the past, perhaps we can better understand the transformational roles of advanced technologies in mathematical thinking and education. Computers are doing far more than making it easier or faster to do what we are already doing. The sociohistorical context may also illuminate promising directions for research and practice on computers in mathematics education and make sense of the drastic reformulations in the aims and methods of mathematics education wrought by computers.

The chapter then proposes an heuristic taxonomy of seven functions whose incorporation into educational technologies may promote mathematical thinking. It distinguishes two types of functions: *purpose* functions, which may affect whether students choose to think mathematically, and *process* functions, which may support the component mental activities of mathematical thinking. My hope is that the functions falling into these two categories will apply to all cognitive technologies, that they will help students to think mathematically, and that they can be used both retroactively to assess existing software and proactively to guide software development efforts. Definitions and examples of software are provided throughout the chapter to illustrate the functions.

The central role that mathematical thinking should play in mathematics education is now receiving more attention, both among educators and in the research community (e.g. Schoenfeld, 1985a; Silver, 1985). As Schoenfeld says, "You understand how to think mathematically when you are resourceful, flexible, and efficient in your ability to deal with new problems in mathematics" (1985a, p. 2). The growing alignment of mathematics learning with mathematical thinking is a significant shift in education.

THEMES OF CHANGE IN THE AIMS AND METHODS OF MATH EDUCATION

There is no question but that information technologies, in particular the computer, have radical implications for our methods and are already changing them. But, perhaps more importantly, we are also coming to see that they are changing our *aims* and thereby what we consider the goals of mathematical understanding and thinking to which our educational processes are directed.

Mathematics educators, represented by such organizations as NCTM, are fundamentally rethinking their aims and means. In particular, mathematics activities are becoming significant in a much wider variety of contexts than ever before. The reason for this expansion is the widespread availability of powerful mathematical tools that simplify numerical and symbolic calculations, graphing and modeling, and many of the mental operations involved in mathematical thinking. For example, many classrooms now have available programmable calculators, computer languages, simulation and modeling languages, spreadsheets, algebraic equation solvers such as TK!SOLVER, symbolic manipulation packages and software for data analysis and graphing. The drudgery of remembering and practicing cumbersome algorithms is now often supplanted by activities quite different in nature: selecting appropriate computer programs and data entry.

Why have these revolutionary changes occurred? How can we use them as a guide in the design, testing, and use of the new technologies, so that we can enhance both the processes of mathematics education and our understanding of how it occurs? In other words, what are the beacons that will help light the way as we consider the role of cognitive technologies in mathematics education?

AN HISTORICAL APPROACH TO MENTAL ROLES FOR COGNITIVE TECHNOLOGIES

An historical approach will help us consider how the powers of information technologies can best serve mathematics education and research. It will help us look beyond the information age to understand the transformational roles of cognitive technologies and to illuminate their potential as tools of mentation. Long before computers appeared, technical instruments such as written language expanded human intelligence to a remarkable extent. I take as axiomatic that intelligence is not a quality of the mind alone, but a product of the relation between mental structures and the tools of the intellect provided by the culture (Bruner, 1966; Cole & Griffin, 1980; Luria, 1976, 1979; Olson, 1976, 1985; Olson & Bruner, 1974; Pea, 1985b; Vygotsky, 1962, 1978). Let us call these tools cognitive technologies.

A cognitive technology is any medium that helps transcend the limitations of the mind (e.g., attention to goals, short-term memory span) in thinking, learning, and problem-solving activities. Cognitive technologies have had remarkable consequences on the varieties of intelligence, the functions of human thinking, and past intellectual achievements (e.g., Cassirer, 1944; Goodman, 1976). They include all symbol systems, including writing systems, logics, mathematical notation systems, models, theories, film and other pictorial media, and now symbolic computer languages. The technologies that have received perhaps the most attention as cognitive tools are *written language* (Goody, 1977; Greenfield, 1972; Olson, 1977; Ong, 1982; Scribner & Cole, 1981), and *systems of mathematical notation,* such as algebra or calculus (Cassirer, 1910, 1957; Kaput, 1985, in press; Kline, 1972) and number symbols (Menninger, 1969).

Contrast for a moment what it meant to learn math with a chalk and board, where one erased after each problem, with what it meant to use paper and pencil, where one could save and inspect one's work. This example reminds us that under the broad rubric of the "cognitive technologies" for mathematics, we must include entities as diverse as the chalk and board, the pencil and paper, the computer and screen, and the symbol systems within which mathematical discoveries have been made and that have led to the creation of new symbol systems. *Each has transformed how mathematics can be done and how mathematics education can be accomplished.* It would be interesting to explore, if space allowed, the particular ways in which mathematics and mathematics education changed with the introduction of each medium.

A common feature of all these cognitive technologies is that they make *external* the intermediate products of thinking (e.g., outputs of component steps in solving a complex algebraic equation), which can then be analyzed, reflected upon, and discussed. Transient and private thought processes subject to the distortions and limitations of attention and memory are "captured" and embodied in a communicable medium that persists, providing material records that can become objects of analysis in their own right—conceptual building blocks rather than shifting sands. Vygotsky (1978) heralded these tools as the "extracortical organizers of thought," because they help organize thinking outside the physical confines of the brain.

We are now seeing, in ways described throughout the chapter, how computers are an especially potent type of cognitive technology for learning to think mathematically: They can operate not only with numbers, but also with symbols—the fundamental currency of human thought. Computers are universal machines for storing and dynamically manipulating symbols. Capable of real-time programmable interactions with human users, computers may provide the most extraordinary cognitive technologies thus devised. But what can we learn from the history of noncomputer-based cognitive technologies that will inform our current inquiries?

Cognitive technologies, such as written languages, are commonly thought of as *cultural amplifiers* of the intellect, to use Jerome Bruner's (1966, p. xii) phrase. They are viewed as cultural means for empowering human cognitive capacities. Greenfield and Bruner (1969) observed that cultures with technologies such as written language and mathematical formalisms will "push cognitive growth better, earlier, and longer than others" (p. 654). We find similarly upbeat predictions embodied in a widespread belief that computer technologies will inevitably and profoundly amplify human mental powers (Pea & Kurland, 1984).

This amplifier metaphor for cognitive technologies has led to many research programs, particularly on the cognitive consequences of literacy and schooling (e.g. on formal logical reasoning) in the several decades since Bruner and his colleagues published *Studies in Cognitive Growth* (e.g., Greenfield, 1972; Olson, 1976; Scribner & Cole, 1981). The metaphor persists in the contemporary work on electronic technologies by John Seeley Brown of Xerox PARC, who, in a recent paper, described his prototype software systems for writing and doing mathematics as "idea amplifiers" (Brown, 1984a). For example, AlgebraLand, created by Brown and his colleagues (Brown, 1984b), is a software program in which students are freed from hand calculations associated with executing different algebraic operations and allowed to focus on high level problem-solving strategies they select for the computer to perform. AlgebraLand is said to enable students "to explore the problem space faster," as they learn equation solving skills. Although *quantitative* metrics, such as the efficiency and speed of learning, may truly describe changes that occur in problem solving with electronic tools, more profound changes—as I will later describe for the AlgebraLand example—may be missed if we confine ourselves to the amplification perspective.

There is a different tradition that may be characterized as the *cultural-historical* study of cognitive technologies. This perspective is most familiar to psychologists and educators today in the influential work of Vygotsky (1978). Vygotsky offered an account of the development of higher mental functions, such as planning and numerical reasoning, as being baed on the "internalization" of self-regulatory activities that first take place in the social interaction between children and adults. The historical roots of Vygotsky's orientation provide an illuminating framework for the roles of computer technologies in mathematical thinking and learning. Influenced by the writings of Vico, Spinoza, and Hegel, Marx and Engels developed a novel and powerful theory of society now described as *historical,* or *dialectical, materialism.* According to this theory, human nature is not a product of environmental forces, but is of our own making as a society and is continually in the process of "becoming." Humankind is reshaped through a dialectic, or "conversation," of reciprocal influences: *Our productive activities change the world, thereby changing the ways in which the world can change us.* By shaping nature and how our interactions with it are mediated, we change ourselves. As the biologist Stephen Jay Gould observes (1980), such "cultural evolution," in contrast to Darwinian biological evolution, is defined by the transmission of skills, knowledge, and learned behavior across generations. It is one of the ways that we as a species have transcended nature.

Seen from this cultural-historical perspective, *labor* is the factor mediating the relationship of human beings to nature. By creating and using physical instruments (such as machinery) that make our interaction with nature less and less direct, we reshape our own, human nature. The change is fundamental: Using different instruments of work (e.g. a plow rather than the hand) changes the functional organization, or system characteristics, of the human relationship to work. Not only is the work finished more quickly, but the actions necessary to accomplish the required task have changed.

In an attempt to integrate accounts of individual and cultural changes, the Soviet theorists L. S. Vygotsky (e.g., 1962, 1978) and A. R. Luria (1976, 1979) generalized the historical materialism that Marx and Engels developed for physical instruments. They applied it to an historical analysis of symbolic tools, such as written language, that serve as instruments for redefining culture and human nature. What Vygotsky recognized was that "mental processes always involve signs, just as action on the environment always involves physical instruments (if only a human hand)" (Scribner & Cole, 1981, p. 8). A similar instrumental and dialectical perspective is reflected in recent studies of the "child as a cultural invention" (Kessel & Siegel, 1983; Kessen, 1979; White, 1983). Take, for instance, Wartofsky's (1983) description of the shift in perspective:

> Children are, or become, what they are taken to be by others, and what they come to take themselves to be, in the course of their social communication and interactions with others. In this sense, I take "child" to be a social and

historical kind, rather than a natural kind, and therefore also a constructed kind rather than one given, so to speak, by nature in some fixed or essential form. (p. 190)

Applied to mathematics education, this sociohistorical perspective highlights not the *constancy* of the mathematical understandings of which children are capable at particular ages, but how what we take for granted as limits are redefined by the child's use of new cognitive technologies for learning and doing mathematics. Similarly, Cole and Griffin (1980) noted how symbolic technologies qualitatively change the structure of the *functional system* for such mental activities as problem solving or memory.

The term "amplify" has other implications. It means to make more powerful, and to amplify in the scientific sense "refers specifically to the intensification of a signal (acoustic, electronic), *which does not undergo change in its basic structure*" (Cole & Griffin, 1980, p. 349). Thus, the amplifier metaphor for the roles of technologies in mathematical thinking leads one to unidimensional, quantitative theorizing about the effects of cognitive technologies. A pencil seems to amplify the power of a sixth grader's memory for a long list of words when only the *outcome* of the list length is considered. But it would be distortive to go on to say that the *mental process* of remembering that leads to the outcome is amplified by the pencil. The pencil does not amplify a fixed mental capacity called memory; it restructures the functional system of remembering and thereby leads to a more powerful outcome (at least in terms of the number of items memorized). Similar preoccupations with amplification led researchers to make quantitative comparisons of enhancements in the learning of basic math facts that are brought about by software and print media, rather than to consider the fundamental changes in arithmetical thinking that accompany the usage of programmable calculator functions (Conference Board of the Mathematical Sciences, 1983; Fey, 1984; National Science Board, 1983).

Olson (1976) makes similar arguments about the capacity of written language to restructure thinking processes. For example, written language facilitates the logical analysis of arguments for consistency–contradiction because print provides a means of storing and communicating cultural knowledge. It transcends the memory limitations of oral language. What this means is that technologies do not simply either *amplify*, like a radio amplifier, the mental powers of the learner or speed up and make the process of reaching previously chosen educational goals more efficient. The standard image of the cognitive effects of computer use is one-directional: that of the child seated at a computer terminal and undergoing certain changes of mind as a direct function of interaction with the machine. The relatively small number of variables to measure makes this image seductive for the researcher. But since the technologies change the system of thinking activities in which the technologies play a role, their effects are much more complex and often indirect. Like print, they transcend the memory limitations of oral language. Complicating matters even more is that the *specific* restructurings of cognitive technologies are seldom predictable; they have emergent properties that are discovered only through experimentation.

I espouse a quite different theory about the cognitive effects of computers than that just described. My theory is consistent with questions based in a two-directional image that other mathematics educators and researchers (e.g., Kaput, 1985, in press) are posing, such as: What are the *new* things you can do with technologies that you could not do before or that weren't *practical* to do? Once you begin to use the technology, what totally new things do you realize might be *possible* to do? By "two-directional image," I mean that not only do computers affect people, but people affect computers. This is true in two senses. In one sense, we all affect computers and the learning opportunities they afford students in education by how we *interpret* them and by what we define as appropriate practices with them; as these interpretations change over time, we change the effects the computers can have by changing what we do with them. (Consider how we began in schools, with drill and practice and computer literacy activities, and now emphasize the uses of computers as tools, such as word processors, spreadsheets, database management systems.) In another sense, we affect computers when we study their use, reflect on what we see happening, and then act to change it in ways we prefer or see as necessary to get the effects we want. Such software engineering is fundamentally a dialectical process between humans and machines. We define the educational goals (either tacitly or explicitly) and then create the learning activities that work toward these goals. We then try to create the appropriate software. We experiment and test, experiment and test, until we are satisfied . . . which we tend never to be. Experimentation is a spiral process toward the unknown. Through experimentation, new goals and new ideas for learning activities emerge. And so on it goes—we create our own history by remaking the tools with which we learn and think, and we simultaneously change our goals for their use.

COGNITIVE TECHNOLOGIES IN MATHEMATICS EDUCATION

How does the idea of cognitive technologies relate to mathematics education? A few historical notes prepare the stage. We may recall Ernst Mach's (1893/1960) statement, in his seminal work on the science of mechanics earlier this century, that the purpose of mathematics should be to save mental effort. Thus arithmetic procedures allow one to bypass counting procedures, and algebra substitutes "relations for values, symbolizes and definitively fixes all numerical operations that follow the same rule" (p. 583). When numerical operations are symbolized by mechanical operations with symbols, he notes, "our brain energy is spared for more important tasks" (p. 584), such as discovery or planning. Although overly neural in his explanation, his point about freeing up mental capacity by making some of the functions of problem solving automatic is a central theme in cognitive science today.

Whitehead (1948) made a similar point: "By relieving the brain of all unnecessary work, a good notation sets it free to concentrate on more advanced problems, and in effect increases the mental power" (p. 39). He noted that a Greek mathematician would be astonished to learn that today a large proportion of the population can perform the division operation on even extremely large numbers (Menninger, 1969). He would be more astonished still to learn that with calculators, knowledge of long division algorithms is now altogether unnecessary. Further arguments about the transformational roles of symbolic notational systems in mathematical thinking are offered by Cajori (1929a, 1929b), Grabiner (1974), and, particularly, Kaput (in press).

Although long on insight, Mach and Whitehead lacked a cognitive psychology that explicated the *processes* through which new technologies could facilitate and reorganize mathematical thinking. What aspects of mathematical thinking can new cognitive technologies free up, catalyze, or uncover? The remainder of this chapter is devoted to exploring this central question.

A historical approach is critical because it enables us to see how looking only at the contemporary situation limits our thinking about what it *means* to think mathematically and to be mathematically educated (cf. Resnick & Resnick, 1977, on comparable historical redefinitions of "literacy" in American education). These questions become all the more significant when we realize that our cognitive and educational research conclusions to date on what student of a particular age or Piagetian developmental level can do in mathematics are restricted to the *static* medium of mathematical thinking with paper and pencil.[1] The dynamic and interactive media provided by computer software make gaining an intuitive understanding (traditionally the province of the professional mathematician) of the interrelationships among graphic, equational, and pictorial representations more accessible to the software user. Doors to mathematical thinking are opened, and more people may wander in.

[1]This argument is developed more fully with respect to cognitive development in general with new technologies in Pea (1985a).

Thus, the basic findings of mathematical education will need to be rewritten, so that they do not contain our imagination of what students might do, thereby hindering the development of new cognitive technologies for mathematics education.

TRANSCENDENT FUNCTIONS FOR COGNITIVE TECHNOLOGIES IN MATHEMATICS EDUCATION

Rationale

What strategy shall we choose for thinking about and selecting among cognitive technologies in mathematics education? I argue for the need to move beyond the familiar cookbooks of 1,001 things, in near random order, that one *can* do with a computer. Such lists are usually so vast as to be unusable in guiding the current choice and the future developments of mathematics educational technologies. Instead, we should seek out high leverage aspects of information technologies that promote the development of mathematical thinking skills. I thus propose a list of "transcendent functions" for cognitive technologies in mathematics education.

What is the status of such a list of functions? Incorporating them into a piece of software would certainly not be sufficient to promote mathematical thinking. The strategy is more *probabilistic*—other things being equal, more students are likely to think mathematically more frequently when technologies incorporate these functions. Some few students will become prodigious mathematical thinkers, whatever obstacles must be overcome in the mathematics education they face.[2] Others will not thrive without a richer environment for fostering mathematical thinking. This taxonomy is designed to serve as a heuristic, or guide. Assessments of whether it is useful will emerge from empirical research programs, not from intuitive conjecture. Indeed, until tighter connections can be drawn between theory and practice,[3] the list can only build on what we know from research in the cognitive sciences; it should not be limited by that research.

---

[2]It is more commonly true that prodigious mathematical thinkers have had a remarkable coalescence of supportive environmental conditions for their learning activities, e.g., suitable models, rich resource environment of learning materials, community of peers, and private tutoring (e.g., as described by Feldman, 1980).

[3]This situation is the rule in theory-practice relations in education (Champagne & Chaiklin, 1985; Suppes, 1978). For this reason, I have recently proposed (Pea, 1985b) the need for an activist research paradigm in educational technology, with the goal of simultaneously creating and studying changes in the processes and outcomes of human learning with new cognitive and educational tools.

Finally, why should we focus on transcendent functions? There are two major reasons. We would like to know what functions can be common to all mathematical cognitive technologies, so that each technology need not be created from the ground up, mathematical domain by mathematical domain. We would like the functions to be transcendent in the sense that they apply not only to arithmetic, or algebra, or calculus, but potentially across a wide array, if not all, of the disciplines of mathematical education, past, present, and future. The transcendent functions of mathematical cognitive technologies should thus survive changes in the K–12 math curricula, *since they exploit general features of what it means to think mathematically*—features that are at the core of the psychology of mathematics cognition and learning. These functions should be central regardless of the career emphasis of the students and regardless of their academic future. Lessons learned about these functions from research and practice should allow productive generalizations.

The transcendent functions to be highlighted are those presumed to have great impact on mathematical thinking. They neither begin nor end with the computer but arise in the course of teaching, as part of human interaction. Educational technologies thus only have a role within the contexts of human action and purpose. Nonetheless, interactive media may offer extensions of these critical functions. Let us consider what these extensions are and how they make the nature or variety of mathematical experience qualitatively different and more likely to precipitate mathematics learning and development.

These functions are by no means independent, nor is it possible to make them so. They define central tendencies with fuzzy boundaries, like concepts in general (Rosch & Mervis, 1975). They are also not presented in order of relative importance. I will illustrate by examples how many outstanding, recently developed mathematical educational technologies incorporate many of the functions. But very few of these programs reflect *all* of the functions. And only rare examples in classical computer-assisted instruction, where electronic versions of drill and practice activities have predominated, incorporate any of the functions.

One could approach the question of technologies for math education in quite different ways than the one proposed. One might imagine approaches that assume the dominant role for technology to be amplifier: to give students *more* practice, more *quickly,* in applying algorithms that can be carried out *faster* by computers than otherwise. One could discuss the best ways of using computers for teacher record-keeping, preparing problems for tests, or grading tests. In none of these approaches, however, can computers be considered *cognitive* technologies.

A different perspective on the roles of computer technologies in mathematics education is taken by Kelman et al. (1983) in their book, *Computers in Teaching Mathematics*. They describe various ways software can help create an effective environment for student problem solving in mathematics. Their comprehensive book is organized according to traditional software categories and curriculum objectives: computer-assisted instruction, problem solving, computer graphics, applied mathematics, computer science, programming and programming languages. The spirit of their recommendations is in harmony with the sketch I propose in this chapter, although their orientation is predominantly curricular rather than cognitive. My stress on transcendent functions is thus a complementary approach, taking as a starting point the root or foundational psychological processes embodied in software that engages mathematical thinking.

In my choice of software illustrations I have leaned heavily toward cases that manifest most clearly the specific loci supporting the seven Purpose or Process functions. Although programming languages, spreadsheets, simulation modeling languages such as MicroDynamo (Addison-Wesley), and symbolic calculators such as muMath (MicroSoft) and TK!Solver (Software Arts) can be central to thinking mathematically in an information age (e.g., Elgarten, Posamentier, and Moresh, 1983), I have seldom chosen them as examples. Although I take for granted the utility and power of these types of tools in the hands of a person committed to problem solving, their usefulness stems in part from the extent to which they incorporate the purpose and process functions. For example, Logo graphics programming provides the different mathematical representations of procedural text instructions and the graphics drawing it creates (Process Function 3); and simulation modeling languages and spreadsheets are excellent environments for mathematical exploration (Process Function 2), since hypothesis-testing and model development and refinement are central uses of these interactive software tools. But other environments in which these tools are used—for example, drill and practice on programming language syntax or abstract exercises to write programs to create fibonnaci number series need not offer much encouragement for mathematical thinking. In other words, the intrinsic value of such tools in helping students think mathematically is not a given. The stress on Functions remains central.

A GUIDING DICHOTOMY: PURPOSE AND PROCESS FUNCTIONS FOR COGNITIVE TECHNOLOGIES

How can technology support and promote thinking mathematically? In broad strokes, what appear to be the richest loci of potential cognitive and motivational support of technologies for math education?

We can think of two sides to the educational practices of mathematics learning and ask how software can help. The first side is the personal side—will students choose to commit themselves to learning to think mathematically? Mathematics educators have to some extent neglected the concepts of motivation and purpose (e.g., McLeod, 1985); that neglect may help explain girls' and minorities' documented lack of interest in mathematics. What students learn also depends on the *cognitive support* given them as they learn the many problem-solving skills involved in thinking mathematically.

My perspective on the functions necessary for cognitive technologies thus has two vantage points. First, students are purposive, goal-directed learners, who have the will (on any given occasion or over time) to learn to think mathematically or not. Then once they have embarked on mathematical thinking, they may be aided by technologies in mathematical thinking. For simplicity of exposition, we thus divide function types between: (a) those which promote PURPOSE—engaging students to think mathematically; and (b) those which promote PROCESS—aiding them once they do so.

Purpose Functions in Cognitive Technologies

What lies at the heart of cognitive technologies that help make mathematical thinking purposeful and help commit the learner to the pursuit of understanding? Cognitive technologies that accomplish these goals are based on a participatory link between self and knowledge rather than an arbitrary one. This organic relationship was central to John Dewey's pedagogical writings and integral to Piaget's constructivism: We must build on the child's interests, desires and concerns, and more generally, on the child's world view. But what exactly does this mean?

The key idea behind purpose functions is that they promote the formation of promathematics belief systems in students and thus ensure that students become mathematical thinkers who participate in and own what is learned. Students benefiting from purpose functions are no longer mere storage bins for or executors of "someone else's math." The implication is that technologies for mathematics education should be tools for promoting the student's self-perception as mathematical "agent," as subject or creator of mathematics (Papert, 1972, 1980). For example, Schoenfeld (1985a, 1985b) argues that the belief systems an individual holds can dramatically influence the very *possibilities* of mathematical education:

> Students abstract a "mathematical world view" both from their experiences with mathematical objects in the real world and from their classroom experiences with mathematics. . . . These perspectives affect the ways that students behave when confronted with a mathematical problem, both

influencing what they perceive to be important in the problem and what sets of ideas, or cognitive resources, they use (Schoenfeld, 1985, p. 157).

Although Schoenfeld's focus is broader than the point here, the student's mathematical world view includes the *self*: What am I in relation to this mathematical behavior I am producing? If students do not view themselves as mathematical thinkers, but only as recipients of the "inert" mathematical knowledge that others possess (Whitehead, 1929), then math education *for* thinking is going to be problematical—because the agent is missing.

In the prototypical educational setting, we often erroneously presuppose that we have engaged the student's learning commitment. But the student rarely sees significance in the learning; someone else has made all the decisions about scope and sequence, about the lesson for the day. The learning is meant to deal not with the student's problem or a problematic situation the teacher has helped highlight, but with someone else's. And the knowledge used to solve the problem is someone else's as well, something that someone else might have found useful at some other time. Even that past utility is seldom conveyed: students are almost never told how measurement activities were essential to building projects or making clothes, or how numeration systems were necessary for trade (McLellan & Dewey, 1895).

According to Dewey's (1933, 1938) scheme for the logic of inquiry, the prototypical system of delivering mathematical facts leaves out the necessary *first step* in problem solving: the identification of the problem, the tension that arises between what the student already knows and what he or she needs to know that drives subsequent problem-solving processes. It is interesting that Pólya (1957) also omits this first step; in other respects his phases of problem solving correspond to Dewey's seminal treatment: problem definition, plan creation, plan execution, plan evaluation, and reflection for generalization of what can be learned from this episode for the future (cf., Noddings, 1985). Perhaps the expert mathematician takes this first step for granted: For who could not notice mathematical problems? The world is full of them! But for the child meeting the formal systems that mathematics offers and the historically accrued problem-solving contexts for which mathematics has been found useful, the first step is a giant one, requiring support.

Purpose functions that help the student become a thinking subject can be incorporated into mathematically oriented educational technologies in many ways. Here, we go beyond Dewey to suggest other component features of mathematical agency:

1. *Ownership*. Agency is more likely when the student has primary ownership of the problem for which the knowledge is needed (or second-

ary ownership, i.e., identification with the actor in the problem setting, in an "as if it were me" simulation). A central pedagogical concern is to find ways to help people "own" their own thoughts and the problems through which they will learn. Kaput (1985) and Papert (1980) have provided suggestive examples from software mathematics discovery environments where the "epistemological context" is redefined: Authority for what is known must rest on proof by either the student or the teacher; it must not rest exclusively with the teacher and the text. Students can offer new problems to be solved, and they can also create new knowledge.

2. *Self-worth.* It is hard for students to be mathematical agents if they view opportunities for thinking as occasions for failure and diminished self-worth. Student performance depends partly on self-concept and self-evaluation (Harter, 1985). Research on the motivation to achieve by Dweck and colleagues (e.g., Dweck & Elliot, 1983) indicates that students tend to hold one of two dominant views of intelligence, and that the one held by each particular student helps determine his or her goals. On one hand, if the child views intelligence as an entity, a given quantity of something that one either has or has not, then the learning events arranged at school become opportunities for success or occasions for failure; if the child looks bad, his or her self-concept is negatively affected. On the other hand, if the child views intelligence as "incremental," then these same learning events are viewed as opportunities for acquiring new understanding. Although little is known about the ontogenesis of the detrimental entity view, it is apparent that this belief can hinder the possibility of mathematical agency and that software or thinking practices that foster an incremental world view should be sought.

3. *Knowledge for action.* A third condition for promoting mathematical agency is either that the mathematical knowledge and skills to be acquired have an impact on students' own lives or future careers or that knowledge actually facilitates their solution of real-world problems. New knowledge, whether problem-solving skills or new mathematical ideas, should EMPOWER children to understand or do something better than they could prior to its acquisition. That this condition is important is clear from research on the transferability of instructed thinking skills such as memory strategies (e.g., Campione & Brown, 1978). This research indicates easier transfer of the new skills to other problem settings if one simply explains the benefits of the skill to be learned, that is, that more material will be remembered if one learns this strategy.

Technologies for mathematical thinking that incorporate these Purpose functions should make clear the impact of the new knowledge on the students' lives.

To summarize: In characterizing the general category of Purpose functions for cognitive technologies, I have focused on the importance of linking the child-as-agent with the knowledge to be acquired instead of on the alleged motivational value (e.g., Lepper, 1985; Malone, 1981) of mathematical educational technologies. I have done so because it is inappropriate to think about technologies as artifacts that mechanistically induce motivation. That perspective has led to the extrinsic motivation characterizing most current learning-game software: bells and whistles are added that serve no function in the student's mathematical thinking. Furthermore, these extrinsic motivational features are not proagentive in the sense described earlier. Incorporating the purpose functions I have described into educational technologies could help strengthen *intrinsic* motivation. This can be done by building educational technologies based in specific types of functional and social environments.

Functional Environments That Promote Mathematical Thinking

These are environments that help motivate students to think mathematically by providing mathematics activities whose purposes go beyond "learning math." Whole problems, in which the mathematics to be learned is essential for dealing with the problems, are the focus. The mathematics becomes *functional*, since the technologies prompt the development of mathematical thinking as a means of solving a problem rather than as an end in itself. Systems that provide a functional environment help students interpret the world mathematically in a problem-solving context. Just as in real-life problem solving, associated curricula are not disembodied from purpose (Lesh, 1981). In other words, students see that the mathematics used has a point and can join in the learning activities that pursue the point.

An example is provided in a three-stage approach to algebra education using new technologies outlined in the recent *Computing and Education Report* (Fey, 1984, p. 24). In Stage 1, students begin with "problem situations for which algebra is useful." These types of problem situations—such as science problems of projectile motion and nonlinear profit or cost functions—offer "the best possible motivations." In Stage 2, they learn how to solve such problems using guess-and-test successive approximations—by hand, by graph, and by computer—as well as by means of formal computer tools such as TK!Solver and muMath. In Stage 3, they learn more formal techniques for solving quadratics, such as factoring formulas and the number and types of possible roots. Through such a sequence, students begin by seeing several applications immediately, not by learning techniques whose applications they will see only later. Similar sequences developed from mathematically complex musical or artistic creations are also possible.

Although such functional environments for learning mathematics can be created without computers, computers widen our options. Software may provide innovative, adventurelike problem-solving programs for which mathematical thinking is required of the players if they are to succeed. The five programs in Tom Snyder's (1982) *Search Series* (McGraw-Hill), for example, encourage group problem solving. In Energy Search, students manage an energy factory, collaboratively making interdependent decisions to seek out new energy sources. Geography Search sets students off on a New World search for the Lost City of Gold; climate, stars, suns, water depth, and wind direction, availability of provisions, location of pirates, and other considerations must figure in their navigation plans and progress.

In Bank Street's multimedia "Voyage of the Mimi" Project in Science and Math Education (Char, Hawkins, Wootten, Sheingold, & Roberts, 1983) video, software, and print media weave a narrative tale of young scientists and their student assistants engaged in whale research. Science problems and uses for mathematics and computers emerge and are tackled cooperatively during the group's adventures. One of the software programs, Rescue Mission (also created by Tom Snyder), simulates navigational instruments—such as radar and a direction finder—used on the Mimi vessel and the realistic problem of how to use navigation to save a whale trapped in a fishing net. To work together effectively during this software game, students need to learn how to plan and keep records of emerging data, work on speed-time-distance problems, reason geometrically, and estimate distances. It is in the context of needing to do these things that mathematics comes to serve a functional role.

Sunburst Corporation has also published numerous programs that highlight simulations of real-life events in which students use mathematics skills as aids to planning and problem solving in real-world situations. For example, Survival Math requires mathematical reasoning to solve real-world problems such as shopping for best buys, trip planning, and building construction, and The Whatsit Corporation requires students to run a business producing a product. While problems such as these can be solved on paper, the interactive, model-building features of the computer programs can motivate mathematical thinking much more effectively.

Social Environments for Mathematical Thinking

Social environments that establish an *interactive social context* for discussing, reflecting upon, and collaborating in the mathematical thinking necessary to solve a problem also motivate mathematical thinking.

Studies of mathematical problem solving, for example, by Noddings (1985), Pettito (1984a, 1984b), and Schoenfeld (1985b) indicate how useful *dialogues* among mathematics problem solvers can be in learning to think mathematically. Small group dialogues prompt disbelief, challenge, and the need for explicit mathematical argumentation; the group can bring more previous experience to bear on the problem than can any individual; and the need for an orderly problem-solving process is highlighted (Noddings, 1985). Cooperative learning research in other disciplines of schooling (e.g., Slavin, 1983; Slavin et al., 1985; Stodolsky, 1984) and the new focus in writing composition instruction that emphasizes thinking-aloud activities (Bereiter & Scardamalia, 1986) also focus on social environments. The computer can serve as a fundamental *mediational* tool for promoting dialogue and collaboration on mathematical problem solving. In mathematical learning, as in writing process activities (Grave & Stuart, 1985; Mehan, Moll, & Riel, 1985), social contexts can open up opportunities for the child to develop a distinctive "voice" and to internalize the critical thinking processes that get played out socially in dialogue.

To date, computers have rarely been used to facilitate this function explicitly. The record-keeping and tool functions of software could, however, effectively support collaborative processes in mathematics, just as they have in multiple text authoring environments (Brown, 1984b). This function is usually exploited only implicitly, as in Logo programming, where students often work together to create a graphics program. In doing so, they argue the comparative merits of strategies for solving the mathematics problems that are involved in the programming (Hawkins, Hamolsky, & Heide, 1983; Webb, 1984). The public nature of the computer screen and the ease of revision further encourage collaboration among students (Hawkins, Sheingold, Gearhart, & Berger, 1982). Self-esteem can also grow in a collaborative context when students view one of their peers as expert. There have been some instances in Logo programming research where students with little previous peer support and low self-esteem have emerged as "experts" (Sheingold, Hawkins, & Char, 1984; Papert, Watt, diSessa, & Weir, 1979).

Mathematics is often a social activity in the world. Explicitly recognizing and encouraging this in mathematics education would not only be educationally beneficial and more realistic, but would also make mathematics more enjoyable—sharable rather than sufferable. Mathematics educators should provide better tools for collaborating in mathematics problem solving and work towards promoting more instructionally relevant peer dialogue around mathematical thinking activities.

An example of a program that does just that is part of the Voyage of the Mimi Project in Science and Math Education at Bank Street (Char, Hawkins, Wootten, Sheingold, & Roberts, 1983), a line of software called the Bank Street Lab, developed in conjunction with TERC. It is composed of various kinds of group activities for conducting experiments involving Probe, a hardware device that plugs into the microcomputer and can measure and graph changes in light, heat, temperature, and sound over time. Students work together taking measurements and designing and carrying out experiments. Supplementary teacher materials suggest activities where students work in teams to apply mathematical thinking in making scientific discoveries.

Tom Snyder's programs (1982) also allow for small groups of students working cooperatively or competing against other groups.

Process Functions in Cognitive Technologies

A second set of categories of functions are those which help students understand and use the different mental activities involved in mathematical thinking. Although our understanding of the psychology of mathematics problem solving and learning is continually evolving, there are five different general categories of Process Functions that can be clearly identified for cognitive technologies in math education. Each provides important cognitive support:

- tools for developing conceptual fluency
- tools for mathematical exploration
- tools for integrating different mathematical representations
- tools for learning how to learn
- tools for learning problem-solving methods.

I will briefly define and illustrate each of these categories of functions as they may appear in mathematics software.

1. *Conceptual Fluency Tools*. Fluency tools are programs that free up the component problem-solving *processes* by helping students become more fluent in performing routine mathematical tasks that could be laborious and counterproductive to mathematical thinking. Computer technologies can promote fluency by allowing individually controlled practice on routine tasks and thus freeing up students' mental resources for problem-solving efforts.

There is ample room for debating what these component skills are in secondary school mathematics (e.g. Fey, 1984; Pollak, 1983) and in high schools (Maurer, 1984a, 1984b; Usiskin, 1980). Many software programs routinize irrelevant skills such as practice on long division algorithms. And the issue is made more complex by the fact that there are many mental functions, such as the component operations of numerical and symbolic calculation, that can now be entirely carried out by mathematics software (e.g., Kunkle & Burch, 1984; Pavelle, Rothstein, & Fitch, 1981;

Wilf, 1982; Williams, 1982). We need to focus on determining the skills and knowledge required to design the inputs and understand the outputs of these mathematical tools.

There are routine mathematical tasks that one should be able to do easily to make progress in mathematical thinking. For example, information technologies could improve the fluency of the estimation skills that are at the core of a revised early mathematics curriculum. There are games involving arithmetic estimation activities at which students become quickly proficient (e.g. Pettito, 1984a). The routinization of certain mathematical skills is equally appropriate at the highest level of mathematical achievement. What is at the fringe of mathematical thinking and creativity today is the slog work of tomorrow (Wilder, 1981). What is a creative invention at one point, such as Leibniz's calculus or Gauss's development of complex numbers, is likely to become so routine later that effective instruction makes it widely accessible.

The appropriate roles for such fluency tools is the subject of much current debate (Cole, 1985; Mehan et al., 1985; Patterson & Smith, 1985; Resnick, 1985). Schools frequently establish a two-tiered curriculum, in which basic computational skills are presumed to be necessary prerequisites for engaging in more complex, higher order thinking and problem solving. Limited to activities with little motivational significance in the first tier of this curriculum, many students never engage in the mathematical thinking characteristic of the second tier (Cole, 1985). But recent work in writing (in which such a two-tiered approach is common: Mehan et al., 1985; Simmons, in press) implies that so-called basic skills can be acquired in the *context* of more complex mathematical thinking activities. When a child's conceptual fluency hampers complex thinking, a functional context is established and the child realizes the need for practice. Thus, practice is self-motivated. This contrasts with the two-tiered approach, in which the child is trained to some threshold skill level before being let loose to solve problems. Drill and practice software for fluency in "basic mathematics" seems to work better as a fallback rather than as a startup activity.

2. *Mathematical Exploration Tools*. Mathematics education has long emphasized discovery learning, particularly in the primary school with the use of manipulatives such as Dienes blocks, Cuisenaire rods, and pattern blocks. Much more complex conceptions and mathematical relationships, such as recursion and variables, can also be approached at a more intuitive level of understanding without abstract symbolic equations.

The computational discovery learning environment provides a rich context that helps students broaden their intuition. Logo programming is a paradigm case. The design of Logo environments is based on the assumption that one can recognize patterns and make novel discoveries about properties of mathematical systems through self-initiated search in a well-implemented domain of mathematical primitives (Abelson & Di-Sessa, 1981; Papert, 1980). Nonetheless, recent findings indicate that students encounter conceptual difficulties—for example, with recursion, procedures, and variables—in Logo and find it hard to understand how Logo dictates flow of control for command execution (Hillel & Samurcay, 1985; Kurland & Pea, 1985; Kurland, Clement, Mawby, & Pea, in press; Kurland, Pea, Clement, & Mawby, in press; Kuspa & Sleeman, 1985; Mawby, in press; Pea, Soloway, & Spohrer, in press; Perkins, Hancock, Hobbs, Martin, & Simmons, in press; Perkins & Martin, 1986). There is consequently much debate about the extent and kinds of structure necessary for successful discovery environments. The question of how well lit the paths of discovery need to be remains open.

Many software programs now offer structured exploration environments to help beginning students over some of these difficulties. Programs such as Delta Draw (Spinnaker) and Turtle Steps (Holt, Rinehart & Winston) are recommended as preliminary activities to off-the-shelf Logo. There are in fact several dozen programs that allow students to use a command language to create designs and explore concepts in plane geometry, such as angle and variable, in systematic ways.

The Geometric Supposers (Sunburst Corporation: for Triangles, Quadrilaterals, and Lines; see Kaput, 1985; Schwartz & Yerushalmy, in press) are striking examples of a new kind of discovery environment. Using these programs, students make conjectures about different mathematical objects in plane geometrical constructions—medians, angles, bisectors. Intended for users from grade 6 and up, it is designed so that students can explore the characteristics of triangles and such concepts as bisector and angle. In this way, students can discover theorems on their own. The program is an electronic straight-edge and compass. It comes with "building" tools for defining and labeling construction parts (like the side of a triangle or an angle) and measurement tools for assessing length of lines, degrees of angles, and areas. Most significant, it will remember a geometric construction the student makes on a specific object (such as an obtuse triangle) as a procedure (as in Logo) and allow the student to "replay" it on new, differently shaped objects (e.g., equilateral triangles). The exciting feature of the environment is that interesting properties that emerge in the course of a construction cry out for testing on other kinds of triangles, and students can follow up. Their task is simplified by the labeling and measurement tools provided for mathematical objects in the construction, and experimentation is encouraged. In fact, several students have discovered previously unknown theorems with the discovery tool. Students find this program an exciting entry into empirical geometry (induction), and it can be used to complement classroom work on proofs (deduction). Kaput (1985) has highlighted the major representational breakthrough in the Supposers: they allow a particular construction to represent a general *type* of construction rather than just itself.

3. *Representational Tools*. These tools help students develop the languages of mathematical thought by linking different representations of mathematical concepts, relationships, and processes. Their goal is to help students understand the precise relationships between different ways of representing mathematical problems and the way in which changes in one representation entail changes in others. The languages of mathematical thought, which become apparent in these different representations, include:

- Natural language description of mathematical relations (e.g., linear equations).
- Equations composed of mathematical symbols (e.g., linear equations).
- Visual Cartesian coordinate graphs of functions in two and three dimensions.
- Graphic representations of objects (e.g., in place-value subtraction, the use of "bins" of objects representing different types of place units).

Mathematics educators have begun to use cognitive technologies in this way as a result of empirical studies demonstrating how competency in mathematical problem solving depends partly on one's ability to think in terms of different representational systems during the problem-solving process. Experts can exploit particular strengths of different representations according to the demands of the problem at hand. For example, many relationships that are unclear in textual descriptions, mathematical equations, or other tables of data values can become obvious in well-designed graphs (Tufte, 1983). One can often gain insight into mathematical relationships, like algebraic functions, by seeing them depicted graphically rather than as symbolic equations (Kaput, 1985, in press).

Interactive technologies provide a means of intertwining *multiple representations* (Dickson, 1985) of mathematical concepts and relationships—like graphs and equations or numbers and pictorial representations of the objects the numbers represent. These representations enhance the symbolic tools available to the student and the flexibility of their use during problem solving. They also have the effect of shortening the time required for mathematical experience by allowing, for example, many more graphs to be plotted per unit of time (Dugdale, 1982), and they make possible new kinds of classroom activities involving data collection, display, and analysis (e.g., Kelman et al., 1983: point and function plotting; histograms).

Manipulable, dynamically linked, and simultaneously displayed representations from different symbol systems (Kaput, 1985) are likely to be of value for learning translation skills between different representational systems, although they are as yet untested in research. For example, students can change the value of a variable in an equation to a new value and observe the consequences of this change on the *shape* of the graph. These experiments can be carried out for algebraic equations and graphs in the motivating context of games like Green Globs (in Graphing Equations by Conduit; cf. Dugdale, 1982) and Algebra Arcade (Brooks-Cole). In these games for grades K–6, the player is given Cartesian x-y coordinate axes with 13 green globs randomly distributed on the graph. The players have to type in equations, which the computer graphs; when a graphed equation hits a glob, it explodes and the player's score increases. Students become skilled at knowing how changes in the values of equations, like adding constants or changing factors (x to 3x, for example), correspond to changes in the shape of the graph (Kaput, personal communication). They discover equation forms for families of graphs, such as ellipses, lines, hyperbolas, and parabolas.

Other examples of dynamically linked representational tools are programs that give visual meaning to operations on algebraic equations (such as adding constants to conditionally equivalent operations). Operations on equations are simultaneously presented with coordinate graphs so that any action on an equation is immediately reflected in the graph shape (Kaput, 1985; Lesh, in press). The student can literally see that doing arithmetic, that is, acting on expressions rather than equations, does not change graphs.

Another example is provided by software programs in which different representations are exploited in relation to one another for learning place-value subtraction. We can point to Arithmekit (Xerox PARC: Sybalsky, Burton, & Brown, 1984), Summit (Bolt, Beranek & Newman: Feurzeig & White, 1984), and Place-Value Place (Interlearn). In each of these cases, number symbols and pictorial representations of objects (and in the case of Summit, synthesized voice) are used in tandem to help students understand how the symbols and the operations on them relate to the corresponding pictorial representations. Only Place-Value Place is commercially available. In this program, a calculator displays the addition and subtraction process using three different representations of number values: a standard number symbol, a position on a number line,, and a set of proportionally sized objects (apples, bushels of apples, crates of bushels, truckloads of crates). As students add or subtract numbers, all three displays change simultaneously to illustrate the operation. Kaput (1985) describes efforts underway at the Harvard Educational Technology Center to develop dynamically linked representational tools for ratio and proportion reasoning activities (e.g., m.p.g. and m.p.h. problems: Schwartz, 1984).

Programs incorporating this function are excellent examples of how the rapid interactivity and representational tools the computer provides create a new kind of learning experience. Students can test out hypotheses, immediately see their effects, and shape their next hypothesis accordingly through many cycles, perhaps through many more cycles than they would with noncomputer technologies.

Computers are also frequently used in displaying graphs and functions in algebra, transformations in geometry, and descriptive statistics. Their use is dynamic and allows student interaction with mathematics in ways that would not be possible in noncomputer environments. In the recent *NCTM Computers and Mathematics Report* (Fey, 1984), the importance of multiple representations in mathematics education is highlighted. In particular, the authors note the rich possibilities for the dynamic study of visual concepts, such as symmetry, projection, transformation, vectors, and for developing an intuitive sense of shape and relationship to number and more formal concepts.

4. *Tools for learning how to learn.* This category refers to software programs that promote reflective learning by doing. They start with the details of specific problem-solving experiences and allow students to consolidate what they have learned in episodes of mathematical thinking. They focus on what both Dewey (1933) and Pólya (1957) describe as the final step in problem solving, reflection that evaluates the work accomplished and assesses the potential for generalizing methods and results (Brown, 1984b). These programs also make possible, in ways to be described, new activities for learning how to learn.

The programs leave traces of the student's problem-solving steps. Tools based on this function provide a more powerful way of learning from experience, because they help students relive their experience. The problem-solving tracks that students leave behind can serve as explicit materials for *studying, monitoring,* and *assessing* partial solutions to a problem as they emerge. They can help students learn to *control* their strategic knowledge and activities during problem-solving episodes (Schoenfeld, 1985a, 1985b).

The crucial feature of such systems is that students have access to trace records of their problem solving *processes* (e.g., the network of steps in geometry proofs: Geometry Tutor [Boyle & Anderson, 1984]; the sequence of operations in algebra equation-solving: AlgebraLand [Xerox PARC; Brown, 1984b]). These records externalize thought processes,

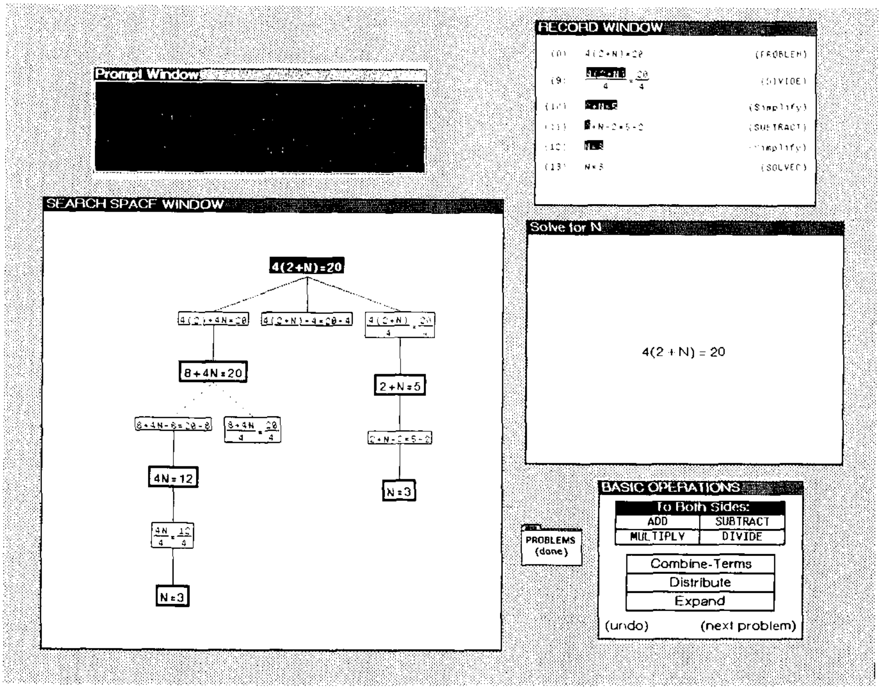

FIG. 4.1.

thus making them accessible for inspection and reflection. Let us consider AlgebraLand, in particular its approach to linear equations.

Search is not a central concept in algebra instruction today, but a central insight of cognitive science is that learning problem-solving skills in mathematics requires well-developed search procedures, that is, knowledge about when to select what subgoals and in what sequence. In most classroom instruction of algebra equation solving, the teacher selects the operator to be applied to an equation (e.g., add-to-both-sides), and the student carries out the arithmetic. The pedagogical flaw in this method is that the student does not learn *when* to select the various subgoals (Simon, 1980), but only *how* to execute them (e.g., to do the arithmetic once the divide operation has been selected).

Originally created several years ago at Xerox PARC by J. S. Brown, K. Roach, and K. VanLehn, and currently being revised by C. Foss for work with middle school students, AlgebraLand is an experimental system for helping students learn algebra by doing problems (Brown, 1984b). Figure 4.1 illustrates some of the features of the system to be discussed.

The task in Fig. 4.1 for the student is to solve the equation for N (shown in the *Solve for N* window on the figure's right side). Algebraic operators listed in the *Basic Operations window* on the bottom right-hand side (such as Combine-Terms, Add-to-Both-Sides, Distribute) can be selected to apply to the whole equation or to one of its subexpressions. After selecting the operation and where to apply it, the student can execute it. This creates a second algebraic expression.

The *Record window,* upper right, records the steps taken by the student towards a solution. Its left-hand column lists all the intermediate expressions; its right-hand column shows each operation used to transform the recorded expression. The *Search Space window* represents all the student's steps as a search tree; it displays solution paths depicting all the student's moves, including backtracking. In the solution attempt depicted here, the student took three different approaches to solving the equation. These approaches are reflected in the three branches that issue from the original equation. Each intermediate expression that resulted from applying the do-arithmetic operator appears in boldface for clarity. AlgebraLand performs all tactical, algebraic operations and arithmetical calculations, effectively eliminating errors in arithmetic or in the application of operators. The student, whose work is limited to selecting the operator and the scope of its operation, is free for the real mental work of search and operator evaluation.

Operators are also provided for exploring solution paths. There is an *Undo* operator that returns the equation to the state immediately preceding the current one and a *Goto* operator, which is not on the menu, that returns the equation to any prior state. The student can also back up a solution path by applying the inverse of a forward operator (e.g., selecting divide just after applying multiply).

Because the windows show every operator used and every state into which the equation was transformed, students have valuable opportunities to learn from their tracks and to *play* with possibilities. They can explore the search paths of their solution space, examining branch points where, on one stem, an operation was used that led down an unsuccessful path, and on another stem, the operation chosen led down a path towards the solution. Then they can decide which features of the equation at the branch point could have led to the best choice. Transforming that decision into an hypothesis, they can test out that hypothesis in future equation solving. These learning activities are *not* possible with traditional methods for learning to solve equations. Indeed, the cognitive technology offered by AlgebraLand affords opportunities for new and different types of learning through problem solving than were available in static, non-computer-based symbolic technologies.

In summary, the computer environment AlgebraLand emphasizes a procedure diametrically opposed to the traditional instructional method. With AlgebraLand, the student decides *when* to apply operators, and the computer carries out the mechanical procedures that transform the equation. Students do face the problem of searching for and discovering a path of operations leading from the problem state to the solution. The graphic representation provided by the search map allows students to reflect on the means they used to solve the problem, and problem solving is no longer an ephemeral process.

AlgebraLand provides a very rich learning and research environment. But we still know far too little about the potential effectiveness of these types of educational instruments, especially since they introduce new learning problems: How do students learn to "read" and use their traces? How do such learning-how-to-learn skills develop, and what ancillary features of software programs such as games will help students understand how to make effective searches in the space of their "thinking tracks"?

The technology is seldom used in this way, and few developers are working along these lines. Nonetheless, when it is used this way, it provides a qualitatively new kind of tool for students to learn more effectively from their problem-solving experience. It can also provide a rich data source for analyzing student understanding of problem-solving processes and methods.

Providing explicit traces of a student's problem solving activities during an episode reveals possible entry points for tutorials in problem-solving skills and for intelligent coaching. In a variant on AlgebraLand, students could be prompted to check the operation (such as factoring) that they have been using during equation-solving activities, for example, if they have used that operator in a way that has led to a more complex equation rather than a simpler one.

Along the same lines, the information available through such traces provides the data for modeling what a student understands; these models could be based on student interactions with a computer in a specific mathematics problem-solving domain. As experience with current prototypes such as the arithmetic game West indicates (Burton & Brown, 1979), such systems allow for coaching dialogues that are sensitive to a student's developmental level and quite subtle in their coaching methods. Unlike intelligent tutoring systems such as the Geometry Tutor (Boyle & Anderson, 1984) and the Lisp Tutor (Anderson & Reiser, 1985), they don't correct the student after every suboptimal move.

There are two problems with using this type of tool for learning how to learn. First, it is not yet apparent how broadly modeling and coaching can be applied to student misconceptions. Second, all the programs that exemplify this category of function run on minicomputers and have not been developed commercially for schools.

5. *Tools for learning problem-solving methods*. This category of tools

encourages *reasoning strategies* for mathematical problem solving. Recent work on the development of mathematical thinking highlights the importance of reasoning strategies. People who have expertise in approaching problem-solving activities in mathematics utilize, in addition to knowledge of mathematical facts and algorithmic procedures, strategies to guide their work on difficult problems that they cannot immediately solve. Such heuristics, well known from the work of Pólya (1957) and modern studies of mathematical problem solving (e.g., Silver, 1985), include drawing diagrams, annotating these diagrams, and exploiting related problems (Schoenfeld, 1985a). Segal, Chipman, and Glaser (1985) review related instructional programs to teach thinking skills.

Few of the existing examples of educational technologies aim to help students develop problem-solving heuristics of this kind. However, one prominent example should be briefly mentioned. Wumpus (Yob, 1975) is a fantasy computer game in which the player must hunt and slay the vicious Wumpus to avoid deadly pitfalls. Goldstein & colleagues (e.g., Goldstein, 1979) created artificial intelligence programs to help students acquire the reasoning strategies in logic, probability, decision analysis, and geometry that are needed for skillful Wumpus hunting. The Wumpus coaches are minicomputer programs that have not been evaluated in educational settings, and the issue of transfer of the skills acquired by students to settings other than this game has not been studied.

Sunburst Corporation (1985) has developed a problem-solving matrix that is indexed to the software it sells for schools, so that purportedly the teacher can know what problem-solving skills and strategies (e.g., binary search) the student will learn by using the program. However, these strategies are not explicitly taught by the software. Furthermore, the scheme erroneously presupposes that one can identify a priori the problem-solving skills that all students will use at all times in working with a specific program. But the problem-solving processes or component thinking skills a student will use in working with a software program change with cognitive development. The skills that are used also vary across individuals because of cognitive style and other variables.

These reservations notwithstanding, if sufficient attention were devoted to the effort, current tools in artificial intelligence could be used to create learning environments in which the application of a mathematics problem-solving heuristic, or set of heuristics, would be exemplified for many problems. Students could explore these applications and then be offered transfer problems that assess whether they have induced how the heuristic works sufficiently to carry on independently (Wittgenstein, 1956). If they have not, the system could offer various levels of coaching, or different layers of hints, that would lead up to a modeling example for that specific problem.

SYNTHESIS OF FUNCTIONS: THE NEED FOR INTERDISCIPLINARY RESEARCH ON COGNITIVE TECHNOLOGIES FOR MATHEMATICS EDUCATION

I would like to close this discussion of functions for cognitive technologies in mathematics education by making a plea for interdisciplinary, classroom-based research, involving mathematics educators, cognitive scientists, software makers, and mathematicians. We know very little about the educational impact, actual or possible, of different technology applications in mathematics education. We cannot predict how the role of the human teacher may change as the use of such technologies increases or in what new ways teachers need to be trained to help students use the technologies to gain control of their own mathematical thinking and learning. As these new technologies reduce the focus on teaching routine computation algorithms, teachers' jobs will become much more intellectually challenging; their activities, more like those of mathematicians. Teacher-training institutions will need to change in as yet unspecified ways. We face a plethora of unknowns. Yet, mathematics educators will have to understand these unknowns in order to use cognitive technologies effectively. Empirical testing of exploratory new instructional curricula that embody the various functions I have described is necessary, for without it, we will have little idea whether the functions actually promote mathematical thinking.

Coda

These are very exciting times for learning mathematics and for using new technologies to shape the futures of mathematical thinking. Mathematics was a dreary subject for many of us in the past, but mathematical thinking is now often learned through problem-solving activities that bury the mechanical aspects of mathematics in interesting ideas. And the puritanical attitude that the mind is a muscle to be exercised through mechanical repetition is giving way to a richer view of the creative, exploring mind, which can be nurtured and guided to discover and learn through meaningful problem-solving activities. By infusing life into the learning tools for mathematics, by integrating supports for the personal side of mathematical thinking with supports for knowledge, we can perhaps help each child realize how the powerful abstractions of mathematics confer personal power. In such a utopia, learning mathematics is but one more way of learning how to think and how to define one's personal voice in the world.

It is hoped that the transcendent functions proposed for cognitive technologies in mathematical education will be useful for crafting new generations of cognitively supportive and personally meaningful learning and teaching tools for mathematical thinking. They may also provide practitioners with new ways of evaluating these educational tools. Other researchers are bound to offer a different set of transcendent functions than those I have proposed; debate should clarify the issues under discussion and contribute to the fundamental goal of using cognitive technologies in mathematics education to prepare students effectively for the complexities of mathematical thinking.

Although we cannot predict what shape the cognitive technologies for mathematics education will take, we can certainly monitor whether they are congruent with emerging concepts of mathematical thinking and the nature of the learner, and assure that the science and tools of education are never far apart.

ACKNOWLEDGMENTS

I have had opportunity to discuss some of these issues with Wally Feurzeig, Ron Mawby, Andrea Pettito, Al Schoenfeld, Judah Schwartz, Kurt VanLehn, Ron Wenger, Jim Wilson, and members of the Sloan workgroup on Cognitive Science and Math Education. Jim Kaput, in particular, has greatly influenced this account of how drastically information technologies are changing and will continue to change our curriculum topics and educational goals in mathematics. None of those mentioned necessarily subscribes to the taxonomy and arguments presented here.

REFERENCES

Abelson, H., & diSessa, A. A. (1981). *Turtle geometry: The computer as medium for exploring mathematics*. Cambridge, MA: MIT Press.

Anderson, J. R., & Reiser, B. (1985). The LISP tutor. *Byte, 10*, 159–175.

Bereiter, C., & Scardamalia, M. (1986). *The psychology of written composition*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Boyle, C. F., & Anderson, J. R. (1984). *Acquisition and automated instruction of geometry proof skills*. Paper presented at the meeting of the American Educational Research Association, New Orleans, LA.

Brown, J. S. (1984a). *Idea amplifiers: New kinds of electronic learning environments*. Palo Alto, CA: Xerox Palo Alto Research Center, Intelligent Systems Laboratory.

Brown, J. S. (1984b). Process versus product: A perspective on tools for communal and informal electronic learning. In S. Newman & E. Poor (Eds.), *Report from the learning lab: Education in the electronic age* (pp. 41–57). New York: Educational Broadcasting Corporation, WNET.

Bruner, J. S. (1966). On cognitive growth II. In J. S. Bruner, R. R. Olver, & P. M. Greenfield (Eds.), *Studies in cognitive growth* (pp. 30–67). New York: Wiley.

Burton, R. R., & Brown, J. S. (1979). An investigation of computer coaching for informal learning activities. *International Journal of Man-Machine Studies, 11*, 5–24.

Cajori, F. (1929a). *A history of mathematical notations, Vol. 1: Notations in elementary mathematics*. La Salle, IL: Open Court.

Cajori, F. (1929b). *A history of mathematical notations, Vol. 2: Notations mainly in higher mathematics*. La Salle, IL: Open Court.

Campione, J. C., & Brown, A. L. (1978). Training general metacognitive skills in retarded children. In M. Grunberg, P. E. Morris, & R. N. Sykes (Eds.), *Practical aspects of memory* (pp. 410–417). New York: Academic Press.

Cassirer, E. (1910). *Substance and function and Einstein's theory of relativity*. Mineola, NY: Dover.

Cassirer, E. (1944). *An essay on man: An introduction to a philosophy of human culture*. New Haven, CT: Yale University Press.

Cassirer, E. (1957). *The philosophy of symbolic forms. Vol. 3: The phenomenology of knowledge*. New Haven, CT: Yale University Press.

Champagne, A. B., & Chaiklin, S. (1985). *Interdisciplinary research in science learning: Great prospects and dim future*. Paper presented at the meeting of the American Educational Research Association, Chicago, IL.

Char, C., Hawkins, J., Wootten, J., Sheingold, K., & Roberts, T. (1983). *"Voyage of the Mimi": Classroom case studies of software, video, and print materials*. New York: Bank Street College of Education, Center for Children and Technology.

Cole, M. (1985). *Non-cognitive factors in education: Subcommittee report*. Washington, DC: National Research Council Commission on Behavioral and Social Sciences and Education.

Cole, M., & Griffin, P. (1980). Cultural amplifiers reconsidered. In D. R. Olson (Ed.), *The social foundations of language and thought: Essays in honor of Jerome S. Bruner* (pp. 343-364). New York: Norton.

Conference Board of the Mathematical Sciences (1983). *The mathematical sciences curriculum K–12: What is still fundamental and what is not*. Washington, DC: CBMS.

Dewey, J. (1933). *How we think*. Chicago: Henry Regnery.

Dewey, J. (1938). *LOGIC: The theory of inquiry*. New York: Henry Holt & Co.

Dickson, P. (1985). Thought-provoking software: Juxtaposing symbol systems. *Educational Researcher, 14*, 30–38.

Dugdale, S. (1982, March). Green Globs: A microcomputer application for the graphing of equations. *Mathematics Teacher*, 208–214.

Dweck, C. S., & Elliot, E. S. (1983). Achievement motivation. In P. H. Mussen (Ed.), *Handbook of child psychology, Vol. IV* (pp. 643–691). New York: Wiley.

Elgarten, G. H., Posamentier, A. S., & Moresh, S. E. (1983). *Using computers in mathematics*. Menlo Park, CA: Addison-Wesley.

Feldman, D. H. (1980). *Beyond universals in cognitive development*. Norwood, NJ: Ablex.

Feurzeig, W., & White, B. (1984). *An articulate instructional system for teaching arithmetic problems*. Cambridge, MA: Bolt, Beranek, & Newman.

Fey, J. T. (1984). (Ed.). *Computing and mathematics: The impact on secondary school curricula*. College Park, MD: National Council of Teachers of Mathematics, The University of Maryland.

Goldstein, I. (1979). The genetic epistemology of rule systems. *International Journal of Man-Machine Studies, 11*, 51–57.

Goodman, N. (1976). *Languages of art*. (Rev. ed.) Amherst, MA: University of Massachusetts Press.

Goody, J. (1977). *The domestication of the savage mind*. New York: Cambridge University Press.

Gould, S. J. (1980). Shade of Lamarck. In *The panda's thumb: More reflections in natural history* (pp. 76–84). New York: Norton.

Grabiner, J. V. (1974). Is mathematical truth time-dependent? *American Mathematical Monthly, 81*, 354–365.

Graves, D. H., & Stuart, V. (1985). *Write from the start: Tapping your child's natural writing ability*. New York: Dutton.

Greenfield, P. (1972). Oral and written language: The consequences for cognitive development in Africa, the United States, and England. *Language and Speech, 15*, 169–178.

Greenfield, P. M., & Bruner, J. S. (1969). Culture and cognitive growth (revised version). In D. A. Goslin (Ed.), *Handbook of socialization theory and research* (pp. 633–657). Chicago: Rand McNally.

Harter, S. (1985). Competence as a dimension of self-evaluation: Toward a comprehensive model of self-worth. In R. Leahy (Ed.), *The development of the self*. New York: Academic Press.

Hawkins, J., Hamolsky, M., & Heide, P. (1983). Paired problem solving in a computer context (*Tech. Rep. No. 33*), New York: Bank Street College, Center for Children and Technology.

Hawkins, J., Sheingold, K., Gearhart, M., & Berger, C. (1982). Microcomputers in schools: Impact on the social life of elementary classrooms. *Journal of Applied Developmental Psychology, 3*, 361–373.

Hillel, J., & Samurcay, R. (1985). *Analysis of a Logo environment for learning the concept of procedures with variable*. (Technical Report). Montreal: Concordia University, Mathematics Department.

Kaput, J. J. (1985). *Information technology and mathematics: Opening new representational windows*. Paper presented at National Conference on Using Computers to Teach Complex Thinking, National Academy of Sciences, Washington, DC.

Kaput, J. J. (in press). Towards a theory of symbol use in mathematics. In C. Janvier (Ed.), *Problems of representation and translation in mathematics*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kelman, P., Bardige, A., Choate, J., Hanify, G., Richards, J., Roberts, N., Walters, J., & Tornrose, M. K. (1983). *Computers in teaching mathematics*. Reading, MA: Addison-Wesley.

Kessel, F. S., & Siegel, A. W. (Eds.). (1983). *The child and other cultural inventions*. New York: Praeger.

Kessen, W. (1979). The American child and other cultural inventions. *American Psychologist, 34*, 815–820.

Kline, M. (1972). *Mathematical thought from ancient to modern times*. Oxford: Oxford University Press.

Kunkle, D., & Burch, C. I. Jr. (1984). Symbolic computer algebra: The classroom computer takes a quantum leap. *Mathematics Teacher*, 209–214.

Kurland, D. M., Clement, C., Mawby, R., & Pea, R. D. (in press). Mapping the cognitive demands of learning to program. In J. Bishop, J. Lochhead, & D. Perkins (Eds.), *Thinking*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kurland, D. M., & Pea, R. D. (1985). Children's mental models of recursive Logo programs. *Journal of Educational Computing Research, 1*, 235–243.

Kurland, D. M., Pea, R. D., Clement, C., & Mawby, R. (in press). A study of programming ability and thinking skills in high school students. *Journal of Educational Computing Research*.

Kuspa, L., & Sleeman, D. (1985). *Novice Logo errors*. Unpublished manuscript, Stanford University, School of Education and Department of Computer Science.

Lepper, M. R. (1985). Microcomputers in education: Motivational and social issues. *American Psychologist, 40*, 1–18.

Lesh, R. (1981). Applied mathematical problem solving. *Educational Studies in Mathematics, 12(2)*, 235–265.

Lesh, R. (in press). The evolution of problem representations in the presence of powerful

cultural amplifiers. In C. Janvier (Ed.), *Problems of representation in mathematics learning and problem solving*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Levin, J. (1984). *Place-value place*. Cardiff-by-the-sea, CA: InterLearn.
Luria, A. R. (1976). *Cognitive development: Its cultural and social foundations*. Cambridge, MA: Harvard University Press.
Luria, A. R. (1979). *The making of mind: A personal account of Soviet psychology*. Cambridge, MA: Harvard University Press.
Mach, E. (1960). *The science of mechanics: A critical and historical account of its development*. 6th ed. LaSalle, IL: Open Court. (Originally published, 1893).
Malone, T. W. (1981). Toward a theory of intrinsically motivating instruction. *Cognitive Science, 4*, 333–369.
Maurer, S. B. (1984a). College entrance mathematics in the year 2000. *Mathematics Teacher, 77*, 422–428.
Maurer, S. B. (1984b). The effects of a new college mathematics curriculum on high school mathematics. In A. Ralston & G. S. Young (Eds.), *The future of college mathematics* (pp. 153–175). New York: Springer-Verlag.
Mawby, R. (in press). Proficiency conditions for the development of thinking skills through programming. In J. Bishop, J. Lochhead, & D. N. Perkins (Eds.), *Thinking*. Hillsdale, NJ: Lawrence Erlbaum Associates.
McLellan, J. A., & Dewey, J. (1895). *The psychology of number: And its applications to methods of teaching arithmetic*. New York: Appleton.
McLeod, D. B. (1985). Affective issues in research on teaching mathematical problem solving. In E. A. Silver (Ed.), *Teaching and learning mathematical problem solving: Multiple research perspectives* (pp. 267–279). Hillsdale, NJ: Lawrence Erlbaum Associates, 267–279.
Mehan, H., Moll, L., & Riel, M. (1985). *A quasi-experiment in guided change*. (#NIE 6-83-0027). Washington, DC: National Institute of Education.
Menninger, K. (1969). *Number words and number symbols: A cultural history of numbers*. Cambridge, MA: MIT Press.
National Science Board Commission on Precollege Education in Mathematics, Science, and Technology. (1983). *Educating Americans for the 21st century*. Washington, DC: National Science Foundation.
Noddings, N. (1985). Small groups as a setting for research on mathematical problem solving. In E. A. Silver (Ed.), *Teaching and learning mathematical problem solving: Multiple research perspectives* (pp. 345–359). Hillsdale, NJ: Lawrence Erlbaum Associates.
Olson, D. R. (1976). Culture, technology and intellect. In L. B. Resnick (Ed.), *The nature of intelligence*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Olson, D. R. (1977). From utterance to text: the bias of language in speech and writing. *Harvard Educational Review, 47*, 257–281.
Olson, D. R. (1985). Computers as tools of the intellect. *Educational Researcher, 14*, 5–8.
Olson, D. R., & Bruner, J. S. (1974). Learning through experience and learning through media. In D. R. Olson (Ed.), *Media and symbols: The forms of expression, communication, and education* (pp. 125–150). Chicago: University of Chicago Press.
Ong, W. J. (1982). *Orality and literacy: The technologizing of the word*. New York: Methuen.
Papert, S. (1972). Teaching children to be mathematicians versus teaching about mathematics. *International Journal for Mathematical Education, Science, and Technology, 3*, 249–262.
Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
Papert, S., Watt, D., diSessa, A., & Weir, S. (1979). *An assessment and documentation of a*

children's computer laboratory. Final Report of the Brookline Logo Project. Cambridge, MA: MIT, Department of Artificial Intelligence.

Patterson, J. H., & Smith, M. S. (1985, September). *The role of computers in higher order thinking*. (#NIE-84-0008). Washington, DC: National Institute of Education.

Pavelle, R., Rothstein, M., & Fitch, J. (1981). Computer algebra. *Scientific American, 245*, 136–152.

Pea, R. D. (1985a). Integrating human and computer intelligence. In E. Klein (Ed.), *New directions for child development: Children and Computers, No. 28* (pp. 75–96). San Francisco: Jossey-Bass.

Pea, R. D. (1985b). Beyond amplification: Using the computer to reorganize mental functioning. *Educational Psychologist, 20*, 167–182.

Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology, 2*, 137–168.

Pea, R. D., Soloway, E., & Spohrer, J. (in press). The buggy path to the development of programming expertise. *Focus on Learning Problems in Mathematics*.

Perkins, D. N., Hancock, C., Hobbs, R., Martin, F., & Simmons, R. (in press). Conditions of learning in novice programmers. *Journal of Educational Computing Research*.

Perkins, D. N., & Martin, F. (1986). Fragile knowledge and neglected strategies in novice programmers. In E. Soloway & S. Iyengar (Eds.), *Empirical studies of programming* (pp. 213–229). Norwood, NJ: Ablex.

Pettito, A. L. (1984a). *Beyond drill and practice: Estimation and collaboration in a microcomputer environment*. Paper presented at the meeting of the American Educational Research Association, New Orleans, LA.

Pettito, A. L. (1984b). *Discussant's comments on misconceptions and comments on important issues for research in science and mathematics education*. Paper presented at the meeting of the American Educational Research Association, New Orleans, LA.

Pollak, H. 1983). *The mathematics curriculum: What is still fundamental and what is not*. Conference Board of the Mathematical Sciences (H. Pollak, Chair). Washington, DC.

Pólya, G. (1957). *How to solve it*. New York: Doubleday-Anchor.

Resnick, D. P., & Resnick, L. B. (1977). The nature of literacy: An historical explanation. *Harvard Educational Review, 47*, 370–385.

Resnick, L. B. (1985). *Education and learning to think: Subcommittee report*. Washington, DC: National Research Council Commission on Behavioral and Social Sciences and Education.

Rosch, E., & Mervis, C. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology, 7*, 573–605.

Schoenfeld, A. (1985a). *Mathematical problem solving*. New York: Academic Press.

Schoenfeld, A. (1985b). Metacognitive and epistemological issues in mathematical understanding. In E. A. Silver (Ed.), *Teaching and learning mathematical problem solving* (pp. 361–379). Hillsdale, NJ: Lawrence Erlbaum Associates.

Schwartz, J. (1984). *An empirical determination of children's word problem difficulties: Two studies and a prospectus for further research* (Tech. Rep.). Cambridge, MA: Harvard Graduate School of Education, Educational Technology Center.

Schwartz, J., & Yerushalmy, M. (1985). *The geometric supposers*. Pleasantville, NY: Sunburst Communications.

Schwartz, J., & Yerushalmy, M. (in press). The geometric supposer: The computer as an intellectual prosthetic for the making of conjectures. *The College Mathematics Journal*.

Scribner, S., & Cole, M. (1981). *The psychology of literacy*. Cambridge, MA: Harvard University Press.

Segal, J. W., Chipman, S. F., & Glaser, R. (Eds.). (1985). *Thinking and learning skills: Relating instruction to basic research (Vol. 1)*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Sheingold, K., Hawkins, J., & Char, C. (1984). "I'm the thinkist, you're the typist": The interaction of technology and the social life of classrooms. *Journal of Social Issues, 40,* 49–61.

Silver, E. A. (Ed.). (1985). *Teaching and learning mathematical problem solving: Multiple research perspectives.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Simmons, W. (in press). Beyond basic literacy skills. In R. D. Pea & K. Sheingold (Eds.), *Mirrors of Minds.* Norwood, NJ: Ablex.

Simon, H. A. (1980). Problem solving and education. In D. Tuma & F. Reif (Eds.), *Problem solving and education* (pp. 81–96). Hillsdale, NJ: Lawrence Erlbaum Associates.

Slavin, R. (1983). *Cooperative learning.* New York: Longman.

Slavin, R., Sharan, S., Kagan, S., Hertz-Lazarowitz, R., Webb, C., & Schmuck, R. (1985). *Learning to cooperate, cooperating to learn.* New York: Plenum.

Sleeman, D., & Brown, J. S. (Eds.). (1982). *Intelligent tutoring systems.* New York: Academic Press.

Snyder, T. (1982). *Search Series.* New York: McGraw-Hill.

Stodolsky, S. (1984). Frameworks for studying instructional processes in peer work-groups. In P. L. Peterson, L. C. Wilkinson, & M. Hallinan (Eds.), *The social context of instruction.* Orlando, FL: Academic Press.

Suppes, P. (Ed.). (1978). *Impact of research on education: Some case studies.* Washington, DC: National Academy of Education.

Sybalsky, R., Burton, R., & Brown, J. S. (1984). *Arithmekit.* Palo Alto, CA: Xerox Palo Alto Research Center.

Tufte, E. R. (1983). *The visual display of quantitative information.* Cheshire, CT: Graphics Press.

Usiskin, Z. (1980). What should not be in the algebra and geometry curricula of average college-bound students? *Mathematics Teacher, 73,* 413–424.

Vygotsky, L. S. (1962). *Thought and language.* Cambridge, MA: MIT Press.

Vygotsky, L. S. (1978). *Mind in society: The development of the higher psychological processes.* Cambridge, MA: Harvard University Press.

Wartofsky, M. (1983). The child's construction of the world and the world's construction of the child: From historical epistemology to historical psychology. In F. S. Kessel & A. W. Sigel (Eds.), *The child and other cultural inventions* (pp. 188–215). New York: Praeger.

Webb, N. M. (1984). Microcomputer learning in small groups: Cognitive requirements and group processes. *J. Educational Psychology, 76,* 1076–1088.

White, S. (1983). Psychology as a moral science. In F. S. Kessel & A. W. Sigel (Eds.), *The child and other cultural inventions* (pp. 1–25). New York: Praeger.

Whitehead, A. N. (1929). *The aims of education.* New York: Macmillan.

Whitehead, A. N. (1948). *An introduction to mathematics.* Oxford: Oxford University Press.

Wilder, R. L. (1981). *Mathematics as a cultural system.* New York: Pergamon Press.

Wilf, H. S. (1982). The disk with the college education. *American Mathematical Monthly, 89,* 4–8.

Williams, G. (1982, October). Software Arts' TK!Solver. *Byte,* 360–376.

Wittgenstein, L. (1956). *Remarks on the foundations of mathematics.* Oxford: Blackwell.

Yob, G. (1975). Hunt the Wumpus. *Creative Computing, 1,* 51–54.