A RAND NOTE

Tutoring Techniques in Algebra

David McArthur, Cathleen Stasz,
Mary Zmuidzinas

RAND

The research described in this report was supported by Grant MDR-8751104 from the National Science Foundation.

This Note contains an offprint of RAND research originally published in a journal or book. The text is reproduced here, with permission of the original publisher.

Published 1991 by RAND
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90407-2138

A RAND NOTE

N-3231-NSF

Tutoring Techniques in Algebra

David McArthur, Cathleen Stasz,
Mary Zmuidzinas

Supported by the
National Science Foundation

RAND

Tutoring Techniques in Algebra

David McArthur, Cathleen Stasz, and Mary Zmuidzinas
The RAND Corporation

Although one-to-one tutoring has been regarded as the most effective method of teaching (Bloom, 1984), surprisingly little is understood about tutoring expertise. Much educational research focuses on classroom teaching, whereas the few studies that focus on one-to-one tutoring do not offer a precise information-processing account of this skill. This article describes our initial attempts to study one-to-one tutoring. The goal of our research is to construct a detailed cognitive model of the reasoning and knowledge of an expert human tutor. The method we have employed is a variant of knowledge engineering. We videotaped tutoring sessions with expert teachers, subjecting them to a detailed analysis aimed at abstracting the tutor's knowledge structures. In this article, we describe some important tutoring techniques we have isolated using these methods. We discuss several dimensions along which tutors appear to be intelligent planners and problem solvers. Finally, we note several implications of our research, including its potential impact on the construction of intelligent computer-based tutoring systems.

The search for effective teaching techniques has had a long history in education. A vast body of educational research has focused on the question: What teacher behaviors relate to student outcomes? Much of this research examines the classroom and the relationship between teacher's classroom behavior and students' learning as a whole (e.g., Brophy & Good, 1986). Although classroom teachers must often adjust their instruction to the needs and skills of individual students, relatively little research has examined individualized instruction or one-to-one tutoring (e.g., Collins & Stevens, 1982). We lack a basic theoretical understanding of tutoring, despite the fact that research comparing one-to-one human tutoring with other techniques consistently finds tutoring to be the standard against which to measure all other methods (Bloom, 1984; Cohen, J. Kulik, & C.

Reprinted from *Cognition and Instruction*, Vol. 7, No. 3, pp. 197–244, 1990, © 1990 Lawrence Erlbaum Associates. Reprinted by permission.

C. Kulik, 1982). Similarly, meta-analyses of studies examining the effectiveness of individual tutoring, in the context of computer-aided instruction, generally indicate that computer tutors aid learning (Bangert-Drowns, J. A. Kulik, & C. C. Kulik, 1985; J. A. Kulik, C. C. Kulik, & Bangert-Drowns, 1985).

Recent advances in cognitive science and artificial intelligence have enabled the development of a variety of intelligent tutoring systems (e.g., Anderson, Boyle, & Reiser, 1985; Brown, Burton, & DeKleer, 1982; Clancey, 1979, 1983b). This work has sparked new interest in one-to-one tutoring as an effective instructional strategy. To date, however, these efforts have been most successful at developing methods for student diagnosis (to model the student's knowledge of the subject matter) and representing subject-matter knowledge. The pedagogical rules embodied in many computer tutors, which encode knowledge of one-to-one tutoring expertise, are often ad hoc and, with few exceptions, only loosely linked to theories of teaching or learning. In a recent review of intelligent tutoring systems, Ohlsson (1986) concluded that research on teaching strategies is central to the construction of such tutors, yet little has been done at the level of analysis that would inform their development.

As developers of an intelligent tutor for basic algebra (McArthur & Stasz, 1989; McArthur, Stasz, & Hotta, 1987), we began research into the cognitive skills involved in one-to-one tutoring to inform the design of our system. Our goal is to develop a detailed information-processing model of the skills and heuristics used by expert human algebra tutors to guide students through a one-to-one tutoring session. Our computer-based algebra tutor successfully incorporates knowledge of algebra and of student misconceptions into its program, but we have constantly found ourselves forced to implement tutorial decisions with little justification, for example: Should students be interrupted after every error? Which errors should be corrected and which should be mentioned but not fixed for the student? When should students be given hints or coached? How detailed should hints be? In order to implement such decisions on an informed basis, much more must be learned about the cognitive skills that expert human tutors possess. Specifically, if we wish to automate tutoring expertise in intelligent computer systems, we require a description of the specific *cognitive processes* and *structures* involved in making tutorial decisions. The research reported here represents a first step toward understanding how expert tutors promote effective learning in the domain of algebra.

In the next section, we review the current literature on the cognitive skills of tutoring. Subsequent sections describe initial findings from our own research. In the final sections, we discuss implications of our findings, comparisons with other research, and our directions for future research.

RESEARCH ON TUTORING

What constitutes effective tutoring? Our review of research on human and computer-based tutoring reveals some important research that is beginning to answer this question. However, the work that has been done does not constitute a comprehensive theory, but rather investigates specific aspects of the tutoring process.

Research on Human Tutoring

Various studies of human tutoring attempt to decompose the tutoring process into elements that are expected to relate to student learning. These include the frequency (Colker, 1982) and detail (Putnam, 1985, 1987) of teacher's thoughts about the student and the techniques the tutor uses to facilitate learning (Collins & Stevens, 1983; Putnam, 1987) or motivation (Lepper & Chabay, 1985).

Research on tutor's thoughts has examined the extent to which an expert tutor thinks about the student, as opposed to other concerns, such as instructional objectives or subject-matter content. Classroom research (Colker, 1982; Connors, 1978; Marland, 1977; Marx & Peterson, 1981; McNair, 1978; Semmel, 1977) indicates that teachers' thoughts about their students account for the largest percentage of reported thoughts (39% to 50% in five studies). Furthermore, Colker (1982) found that the frequency with which teachers focused on the cognitive states of their students was not related to group size, even though groups ranged from 1 to 12 students.

However, the level of detail that teachers seek when trying to understand the cognitive state of their students remains unclear. Putnam (1985, 1987), for example, found that tutors rarely determine the exact nature or extent of their student's errors. Tutors ask questions or allow the student to continue working incorrectly to reveal more about their knowledge, but this occurs infrequently—following only 7% of student errors. Rather, their approach was to get the student to understand how to solve the problem correctly—to determine the student's missing skills, then teach them. This approach is in direct contrast to the diagnostic/remedial model implicit in much research on diagnosis (Ashlock, 1982; Brown & Burton, 1978). This model suggests that the teacher tries to determine the underlying cause of the student's error and then to remediate it by correcting the student's faulty understanding.

Collins and Stevens (1983) developed a taxonomy of strategies that teachers may use not only to correct misunderstandings, but also to teach concepts. These strategies include generating hypothetical cases to extend student reasoning, considering an alternative prediction, and entrapping students in their misconceptions. According to Collins and Stevens' theory, teachers choose strategies to pursue a particular teaching goal and maintain an agenda that allows them to allocate their time efficiently among various goals. Which strategies promote which changes in students' cognitive understanding, however, remains unknown.

Lepper and Chabay (1985, 1988) studied motivational aspects of human tutoring in addition to cognitive ones. They noted that, whereas tutors usually agree on motivational goals, they often differ in how they accomplish these goals. Their research classifies tutoring decisions along four dimensions: control (Who should initiate tutoring interventions, the student or tutor?), timing (When does a tutor intervene?), content (What should the tutor say when intervening), and style (How should the tutor say things?). In studying human tutors, they catalogued many ways in which tutors empathize with students and have also speculated on how computers can be more empathetic tutors. With good reason, they believed that motivational considerations may often be as important as cognitive ones in determining the success of tutoring.

Research on Intelligent Tutoring Systems

Another view of tutoring can be found in the work to develop intelligent computer tutors (e.g., Anderson & Skwarecki, 1986; Brown & Burton, 1987; Clancey, 1983b). These systems, as well as our own, aim to adapt instructional content and form to the cognitive needs of the student on a moment-by-moment basis. In a recent review of intelligent tutoring systems, Ohlsson (1986) argued that four principles guide the construction of tutors with this purpose: cognitive diagnosis, subject-matter analysis, teaching tactics, and teaching strategies. Of these four, teaching tactics and strategies are the least well-developed components of most intelligent tutors.

According to Ohlsson (1986), even state-of-the-art tutoring systems have a number of shortcomings. Anderson's geometry tutor (Anderson, Boyler, & Reiser, 1985), for example, operates by comparing the student's problem-solving steps to a large knowledge base consisting of both correct and incorrect problem-solving rules. The tutor intervenes when the student applies an incorrect rule—that is, makes useless or illegal moves in the space of possible inferences. This intelligent monitoring of student activity is very fine grained (at selection or application of a theorem) and thereby provides moment-by-moment flexibility and adaptability. The tutor responds to each action of the learner by mapping his or her action onto a set of problem-solving rules and then retrieving the appropriate tutorial action connected with that rule. However, because each incorrect rule is paired with a particular tutorial action (typically a stored message), every student who takes a given incorrect step gets the same message, regardless of how many times the same error has been made or how many other errors have been made. Thus, at the level of the single tutorial action, "there is no adaptation to the current cognitive state of the learner other than the classification of his last step as an instance of a particular type of error" (Ohlsson, 1986, p. 318).

Anderson's tutor is tactical, driven by local student errors, whereas several other tutoring systems appear to offer more strategic rules for tutoring. For example, Clancey's GUIDON system provides tutoring principles that deal with global aspects of tutoring (e.g., "examine the student's understanding and introduce new information whenever there is an opportunity to do so"; Clancey, 1983b, p. 13). However, as GUIDON now stands, it appears the behavior of its general principles cannot be modified to suit the needs of individual students. Thus, GUIDON and other tutors that embed strategic principles for tutoring may not be inherently more adaptable than Anderson's tutor or others using just local tutoring rules. In Ohlsson's words, "each student gets a unique lesson by drawing out a unique sequence of locally determined responses from the tutor" (Ohlsson, 1986, p. 319). The literature on effective tutoring indicates, at least, that tutoring is more than either global plans or a circumscribed response at a single moment in time. Neither notion alone helps us understand how a teaching goal (e.g., explain the distributive rule) becomes translated into a sequence of teaching techniques to satisfy that goal.

Overall, research on human tutoring is of limited value in developing a cognitive model of tutoring skills, because it often fails to offer a precise description of the tutor's mental processes and expertise. On the other hand, intelligent tutoring systems offer precise models of tutoring, but only for a few components of teaching expertise. Most significantly, although many systems embed impressive student diagnosis or modeling components, they typically fall short in their representation of pedagogical or didactic skills. Thus, an important goal for both human- and computer-based research in teaching is to develop a more thorough understanding of successful tutoring strategies and techniques. In this article, we attempt to formalize some important one-on-one tutoring skills, although our current research by no means addresses all the problems that must be solved to achieve this goal.

METHODOLOGY

To achieve this level of description of tutoring activities in algebra, we have adopted a methodology related to knowledge engineering. Researchers in artificial intelligence who seek to build expert systems use knowledge engineering techniques to extract detailed information about a target subject from a "domain expert" and then formalize the information so it can be implemented as "rules" in an expert system.

Subjects

In the context of this study, our knowledge engineering involved videotaping three ½-hr sessions with three expert tutors and high school students.

All students came from Grades 9 and 10 of a local high school. The teachers we taped each had at least 5 years experience teaching high school algebra and had won awards for teaching excellence. As Berliner (1986), among others, noted, such awards do not ensure that teachers are true expert tutors. Indeed, through our analysis, we see many behaviors we might regard as questionable teaching practice. However, even after intensive examination, we still believe the teachers are competent tutors, if not exceptional ones. Moreover, given the limited knowledge we have of the cognitive skills of tutoring, they provide ample expertise to attempt to formalize, far more than we believe will be automated in intelligent tutoring systems in the next decade.

Two of the three tutors had not previously met the students they tutored; one tutored a student who was then in her introductory algebra course. Perhaps surprisingly, we noticed no difference between tutors that we could confidently attribute to this difference in familiarity. Consequently, for purposes of our analysis, this distinction was disregarded. However, as we note later, additional studies that systematically vary the tutor's background knowledge of the student may be important in arriving at a general understanding of tutoring competence.

Procedure

We examined typical *remedial* tutoring sessions. The tutors were provided with problems that the student had failed on a recent in-class examination. Tutors were not limited to these questions, and all tutors generated their own problems for the student at appropriate points in the sessions. However, although the sessions did not focus exclusively on failed questions, they contrasted with *inquiry* tutoring in which students and tutors engage more in the discovery of new concepts and less in the repair of old ones (see, e.g., Collins, 1988).

Problems were relatively simple equations and inequalities (e.g., $x/3 = b/2 + c/6$). Tutor and student sat at a table and worked the problems together on paper. The videocamera focused on the work, rather than on the tutor and student. Following Putnam (1985, 1987), we viewed the videotape with each teacher after each session, stopping the tape to ask questions and discuss various aspects of the tutoring. Tutors were encouraged to comment on any aspect of the session that they wished. These "stimulated recall" sessions were loosely structured, because our main gaol was to gain insight into the tutor's thought processes and to clarify here goals and actions. This technique served to supplement the videotape data, rather than to provide a complete protocol of the tutor's thoughts while tutoring. Both tutoring and recall sessions were transcribed for further analysis. The results presented here are based on an analysis of three tapes. Although the tapes total no more than 75 min of tutoring, we estimate their analysis has taken about 150 person-hours.

Analysis

To analyze our data, we began by adopting a technique employed by anthropologists studying a variety of interactive behavior.[1] We repeatedly watched the videotapes in groups of (usually) three project members. In each session, we simply observed and discussed the tapes, audiotaping our own reactions for later playback. Our goal was to characterize what we observed in the interactions, without defining categories or hypothesizing underlying intentions and motivations on the part of the speakers.

The patterns that we noted in this analysis eventually suggested useful classifications of tutor behavior. At this point, we supplemented our methods with more traditional protocol analysis techniques and developed a coding scheme for observed patterns at several levels. These included the utterance (single statement by either tutor or student), the exchange (sequence of tutor and student utterances completing some local function), the visible step (sequence of exchanges resulting in an observed algebraic transformation in the solution of the current problem), and the problem. In addition, we also posited a more abstract classification focusing exclusively on the actions of the tutor, which we refer to as *activities*. Tutor activities typically describe an intention imputed to an utterance, exchange, or step. The three protocols were independently coded in terms of activities and their associated utterances, exchanges, and problems. Any discrepancies between coders were discussed and resolved. The coding scheme was refined and revised throughout this iterative process.

Having classified patterns of behavior at several levels, our next goal was to begin to piece together parts of a process model of a tutor's reasoning. Our assumption is that regularities at all levels of tutoring behavior can be explained in terms of underlying knowledge structures. In this article, we attempt to explain only a subset of the patterns we observed, in terms of the underlying tutoring knowledge they imply. For example, the knowledge of conversational conventions and natural language that we assume accounts for many of the regularities at the level of utterances and exchanges will not be addressed here. Our concern is mainly to describe the kinds of pedagogical knowledge that account for patterns within and among exchanges, steps, problems, and especially activities. In this analysis, we have relied on existing information-processing models of cognition from cognitive psychology (e.g., Anderson, 1983) and artificial intelligence (e.g., B. Hayes-

[1]We thank John Seely Brown for suggesting this approach and Gitte Jordan and Lucy Suchman for advising us how to apply this method to our data.

Roth, 1984) to guide the construction of our formative process model of tutoring.

The analysis and conclusions we present later thus represent a mix of analytic methods. We rely heavily on the use of excerpts from protocols to illustrate interesting patterns we observed. In addition, we offer descriptive statistics to provide some indication of the relative frequency and importance of the interesting patterns. Finally, we also present interpretations of patterns and classifications in terms of a process model of tutoring.

RESULTS

Theoretical Framework

We begin the discussion of our results with an overview of the framework of our process model. In some sense we are beginning at the end, because a process model for tutoring is our final goal. However, the framework we discuss here is only the skeletal outline that does not commit to a specific theory of tutoring. Presenting the framework first provides a convenient way to organize subsequent discussion of local observations and conclusions that fill out the framework into a more complete model.

The process model framework we describe is limited to one-to-one tutoring situations in which the tutor is attempting to communicate knowledge to the student through working a series of problems. More specifically, as we discuss later, the model is currently limited to remedial tutoring, in which the student's previous errors on problems largely drive the session. Within this context, the model is generic in the sense that it tries to describe the reasoning and knowledge structures common to all competent tutors. Discussions of the specific tutoring techniques and overall style that distinguish different tutors—which comprise much of what is interesting in an account of tutoring—are deferred until later sections.

As Figure 1 shows, our model includes three main kinds of components: *active memory, knowledge bases*, and a *tutorial planner*. Active memory encodes a history of specific events (representations of tutor and student actions) and conditions (representations of inferences by the tutor). Active memory can include a variety of different kinds of specific information, some of which may be transient and some much more enduring. In this article, the active memory conditions we discuss include:

1. *Current student model*. The student model is a database that stores inferences made by the tutor about the knowledge or lack of knowledge, which attempt to account for the student's performance. Typically, the student model is the product of a diagnostic process.

2. *Current K goals*. K goals, or knowledge goals, refer to the specific topics or concepts that the tutor has decided will be the focus of learning for the student at a given time. K goals may arise as a function of diagnos-

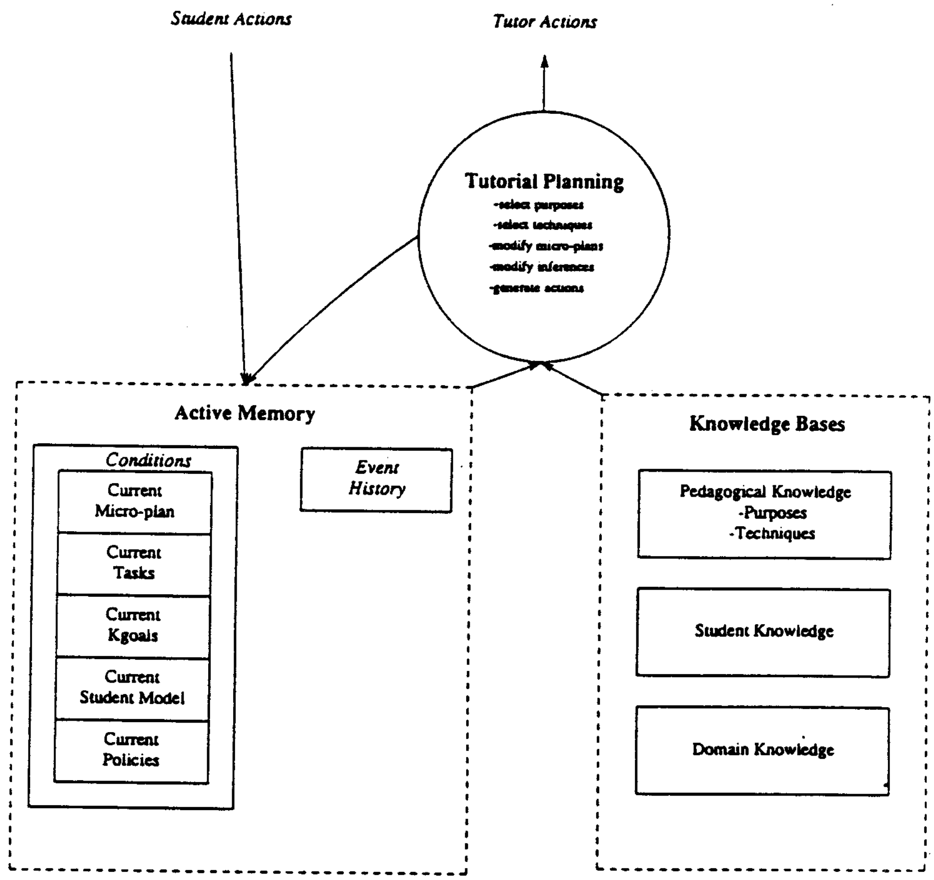

FIGURE 1 Schematic diagram of a tutoring process model framework showing conditions and events in active memory, knowledge bases, and tutorial planning that uses both knowledge and active memory. (Planning results not only in tutorial actions but also in changes to active memory.)

tic inferences that determine the student's strengths and weaknesses and can endure across many problems in a session or even across sessions.

3. *Current microplans*. Microplans refer to brief scripts of activities that tutors appear to use to organize tutoring sessions. Generally, we found that most tutoring is opportunistic, driven by the current data or needs of the student and not controlled by highly structured lesson plans. However, we did notice brief, script-like, repeated patterns of activities. We refer to these as *microplans* to emphasize their brevity.

4. *Current tasks*. The current tasks that the student and tutor are engag-

ing in include not only the problem they are working, but also, within the problem, the specific parts or steps that are now the focus of attention. One of the main findings we discuss later is that tutors decompose large problems into hierarchical sets of tasks and organize problem solving for the student, converting a large and difficult problem into a series of "bite-size" pieces for the student to accomplish.

5. *Current policy*. A tutorial policy refers to general constraints or principles that tutors appear to adhere to in decision making. In contrast with microplans that appear to influence tutoring on a relatively local basis, policies control the pattern of tutoring on a much broader scale, perhaps imparting a style.

This list does not exhaust the kinds of conditions and events that a tutor can attend to in making tutoring decisions. We have enumerated here only the main conditions that we reference later.

In contrast to active memory that encodes specific information about the tutoring session, knowledge bases represent generic information about students, teaching, and subject matter. Although much of active memory changes over time, knowledge bases remain stable; we assume the tutor's teaching expertise is relatively fixed. Just as we do not include all possible components of active memory, we also do not discuss all knowledge bases that contribute to tutoring. For example, we largely ignore tutors' knowledge of students that permits them to develop diagnostic models. As we noted earlier, many researchers have investigated cognitive aspects of student modeling (e.g., Anderson, 1983; Sleeman, 1982). We concentrate here only on the expertise that contributes to tutors' pedagogical knowledge.

We posit that decision making involving knowledge bases is a two-tiered *tutorial planning* process. In the first tier, the tutor consults various conditions and events recorded in active memory to select a pedagogical *purpose* or some small set of complementary purposes. A purpose is a general intent on the part of the tutor to accomplish some particular activity, such as remediating a student error or making a diagnostic inference about such an error. In the second tier of decision making, tutors implement a purpose(s) they have chosen by selecting from a knowledge base of one or several *techniques* consistent with the purpose(s). A technique is a piece of tutoring knowledge whose execution results in an actual tutoring activity. In choosing the tutoring techniques to accomplish a purpose, we assume that tutors can consult additional events and conditions in active memory that permit them to choose the way of accomplishing a general purpose or intent that is tailored to features of the particular situation.

The selection of a specific technique is assumed to have one of two possible effects on the tutoring environment. First, a technique may result in a visible tutoring action (event). For example, when the intention to remediate a student error is refined into a specific remedial technique, the technique actually generates an utterance on the part of the tutor. Second, a technique may result in changes to inferences and conditions in active memory. For example, the intention to perform student diagnosis will result in changes to the student model. A particularly important change we wish to distinguish is that involving the current tutorial microplan. We posit that task management and remedial purposes, discussed later, may not only result in the generation of overt tutorial actions but may also alter or replace the current tutorial microplan.

We postulate that the sequence of activities just discussed executes in a cyclic process. The tutor observes the student's response to the current task and uses this event, together with other conditions and events represented in active memory, to initiate a tutorial planning process. Once the tutor has executed an activity, we assume that he or she will wait for the next student response and then repeat the entire process.

In the following sections, we put some content into our framework. The next section discusses various aspects of the purposes we observed in the tutorial sessions. Subsequent sections document a variety of the techniques that were observed to implement purposes. Separate sections describe techniques that are largely local in scope and those of a more planful nature.

Tutoring Purposes

According to our model, when tutors encounter new events in the tutoring environment (usually a new student response), they first reason about general tutorial purposes that this event and other conditions retained in active memory might suggest. Generally, our observations suggest that tutors pursue only one purpose at a time, although we note some exceptions. Table 1 summarizes all the significant tutorial purposes we observed.

In addition to the name of each purpose, we include a brief description, a note on the kinds of conditions that might trigger the tutor to consider that purpose, and a count of the number of times we observed that purpose, across all tutoring sessions. Some purposes were inferred on little visible data, because they do not normally generate overt tutoring actions. For example, problem-solving purposes in which the tutor silently solves the posed problem are assumed to happen when each new problem is generated or selected. Our assumption is based on existing evidence that tutors compute solution structures in advance (e.g., Clark, Snow, & Shavelson, 1976).

In the following sections, several of the purposes receive little discussion. Problem generation, local clarifications, and performance assessment (even though it was a very frequent purpose) were not the source of interesting patterns of behavior. Motivational purposes are of interest, but these too are mentioned only briefly. As Table 1 shows, we found few tutor utterances that fulfilled an exclusively motivational purpose. However, many tutorial actions appeared to fulfill a motivational function while also accomplishing some

TABLE 1
Summary of Pedagogical Purposes

Purpose Name	*Purpose Description*	*Conditions*	*n*
Problem generation	Generating a new problem for the student and tutor to solve.	End problem	27
Problem solving	Generating reasoning structures that define the tutor's "ideal" reasoning for the current problem.	New problem	27
K- goal modification	Updating the set of concepts that the tutor regards as the main ones that the student should learn.	Changes in student model: inferences of student's strengths or weakness	8
Task management	(a) Deciding which specific task for the student during the working of a problem, (b) introducing the task, (c) monitoring the student's progress at the task, and (d) wrapping up (recapping) the task when completed.	Student response, beginning or end of a task	121
Performance assessment	Evaluating the student's performance at a task. Giving the student performance feedback.	Student response	135
Knowledge assessment	Attempting to understand the student's general level of competence and specific skill deficits. Includes making inferences about underlying causes and misconceptions behind student's visible performance.	Student response, beginning or end of a new topic (K goal)	10
Remediation	Assisting the student when he or she has failed.	Student error	40
Local clarification	Dealing with student comments and questions that are irrelevant to the main flow of the tutoring session.	Student query, unclear student response	4
Motivation	Making a deliberate attempt to maintain or enhance student motivation.	Student error, beginning or end of a task or problem, student inferences: attributions of errors	4

Note. Names of main pedagogical purposes observed in tutoring sessions, along with descriptions of the purpose, general conditions under which the purpose was invoked, and a gross count of the number of times we observed the purpose being executed (*n*).

other purpose. For example, a remedial purpose is often accomplished by choosing a specific remedial technique that both provides useful information and keeps the student's affect high. In general, motivational purposes appear to combine frequently with other informational purposes. In this article, we limit ourselves to examination of informational purposes and techniques.

Remedial Purposes

As previous research and models of tutoring suggest (e.g., Putnam, 1987), remedial purposes were relatively frequent occurrences (40 of all 376 occurrences of purposes; 10.6%). In fact, for every student error we recorded, there was a remedial response. At least the tutors we observed were apparently not willing to let students explore on their own and perhaps discover their own errors. We conjecture that this may be because our tutors were also teachers who place a premium on pacing students rapidly through material, given the time constraints of a classroom situation. Alternatively, teachers may believe that such explorations too frequently lead to unprofitable confusion for the student (A. Collins, personal communication, May 1989). Overall, the pattern of invocation of remedial purposes was straightforward and requires little additional commentary. On the other hand, the rich diversity of remedial techniques used to implement a remedial purpose receives much more discussion.

Knowledge Assessment Purposes

Knowledge assessment is another purpose that models of tutoring might suggest would be relatively frequent (Ohlsson, 1986). Knowledge assessment, also known as student modeling or student diagnosis, is a cornerstone of several models of tutoring that Putnam (1987) has characterized as *diagnostic-remedial*. Such models suggest that the tutor's selection of remedial interventions is largely governed by his or her inferences about the students' misconceptions underlying their overt errors. One would, therefore, predict that many tutor actions could be interpreted as intending to gather data with which to make such inferences, for example, posing questions that would clarify the student's reasoning. As Table 1 indicates, we found little compelling evidence for knowledge assessment. In only 10 cases (2.6% of 376 classified occurrences of purposes) did we observe tutors clearly making such inferences.

Table 2 shows an excerpt that begins with a simple case of knowledge assessment (Lines 1 to 4). We present the excerpt in detail because it is later used to make several points. The transcript of the interaction is shown on the left, and our coding of it is on the right. Parts of the coding scheme are interpreted as necessary. Appendix A discusses the coding scheme in more detail. The knowledge assessment shown in the excerpt is typical of most that we observed. The tutor implements the assessment purpose with a simple query (Lines 1 to 3). The apparently low importance of knowledge assessment is indicated by the fact that the student's indecisive answer to the query (Line 4) does not trigger any further probing by the tutor. The tutor proceeds despite the uncertain information. In 3 of the 10 clear cases of knowledge assessment, it appeared that the tutor did not follow up when receiving answers that did not permit clear inferences.

We also examined the kind of knowledge assessment inferences the tutor was attempting to make by looking at the nature of the question posed. In 7 of the 10 cases, tutors appeared to be interested in only very general diag-

TABLE 2
Excerpt Beginning With Knowledge Assessment

T: Okay, do you want to start on number 17. I don't know, have you done these before ever? [The equation is $x/3 = b/2 + c/6$].	task (do-problem) KNOWLEDGE ASSESS [query]
S: I think we have, I don't remember.	C+
T: Now usually the thing that bothers most students is fractions. They could do without them. So let us get rid of the fractions.	task (do-next-step) task-intro [state mid goal]
S: Okay.	C+
T: Let us multiply every single term by what we refer to as the least common multiple, something all that can divide evenly into. What do you think that would be?	task-intro [state low goal]
S: 6.	R+
T: 6. So I'm going to divide . . . And then it will turn into something you'll feel more comfortable with. [S writes a new equation 3: $x/3 = b/2 + c/6$.] Okay. Now when I say, "multiply each term by 6" I'm going to write in parentheses next to the object 6, okay? [T writes (6) above and to the left of each of the three division terms in the equation.] I'm multiplying each thing by 6. Three into 6 goes how many times? [T points to the 3 under x then] [the parenthesized 6 above it.]	task (do-next-step) task-intro [show work, model reasoning]
S: 2.	R+
T: So what are you going to have left?	task (do-next-step) task-intro [prompt]
S: [No response.]	Ro
T: $2x$. [S writes $2x$ below the $x/3$.]	REMEDIAL [give answer]
S: $2x$.	Re
T: 2 equals . . . 2 goes into 6 how many times? [T points to the 2 under the b, then the parenthesized6 above it.]	task (do-next-step) task-intro [model reasoning]
S: 3.	R+
T: So you'll have	task (do-next-step) task-intro [prompt]
S: $3b$. [S writes $3b$ below the $b/2$.]	R+
T: That's right.	task-pa [correct]
T: 6 into 6 goes . . . . [T points to the 6 in $6/c$, then the parenthesizes 6 above it.]	task (do-next-step) task-intro [model reasoning]
S: Once.	R+
T: So, you'll just have a c. [S writes c below the $c/6$ F now has	task (do-next-step) task-intro [give answer]

(Continued)

TABLE 2 *(Continued)*

48 constructed a new line equation: $2x = 3b + c$.]	
	———
51 T: See now you'll feel much more comfortable with that; you've just been doing that all along. Right?	task-wrapup [confidence assess query]
54 S: Right.	C +
T: So now once you eliminate the fractions, everything else is like what	task-wrapup [problem similarity]
57 we've been doing before. The only other thing remaining to do is to do what to both sides?	task (do-next-step) task-intro [prompt]
60 S: Divide?	R +
T: By	task (do-next-step)
63 S: 2. [S writes $x = (3b + c)/2$.]	R +
	———
T: Good.	task-pa [correct]

Note. An example of an excerpt beginning with a knowledge assessment purpose. The transcript of the interaction is on the left, and a simplified coding of the excerpt is on the right. *Task* indicates the commencement of a task, usually doing either a whole problem or a step within a problem. The scope of the task is indicated inside parentheses, following the task marker. Indentation of tasks indicates conceptual nesting. Upper case markers denote various main purposes like remediation and knowledge assessment. The bracketed items following different purpose markers indicate the specific technique used to accomplish a subpurpose. Dashed lines demarcate separate visible steps and the exchanges that comprise them. Finally, markers such as $R+$ code student responses, as detailed in Appendix A.

nostic information (e.g., "Do you understand?" "Have you ever heard of something called the additive inverse?" "Have you solved inequalities before?" "Have you done these before ever?"). Only three cases show evidence that the tutor was attempting to diagnose a specific misconception in a localized part of an equation (e.g., "Do you know why not $c/3x$?"). Overall, we have found very little evidence for tutors making inferences about the specific student "bugs" that are the focus of many automated student modeling programs (Anderson, Boyle, & Yost, 1985; Brown & Burton, 1978; Sleeman & Smith, 1981). In this regard, our results are consistent with Putnam's (1987), who found that tutors did not perform detailed evaluations of students' knowledge.

Of course it is possible that tutors did perform more extensive knowledge assessment and specific bug detection without any overt evidence. However, there are other reasons to doubt the validity of a strictly diagnostic-remedial model of tutoring. The interactions we observed rarely had the "wait-and-pounce" character one might associate with tutors whose policy was to swing into action only when students made errors. For example, even a brief glance at our coding of Table 2 and the following excerpt in Table 3 shows that, although the tutoring interaction was dense, very little was classified as remedial. Thus, in addition to finding little evidence for diagnosis, we found clear evidence for purposes and tutorial activities other than diagnostic and remedial ones.

Task Management Purposes

Along with remediation, the main purpose responsible for the density of tutorial interactions was task management, accounting for 32% (135 of 376) of all occurrences of classified purposes. Task management is a tutoring activity that has received relatively little attention elsewhere, and we examine it here extensively. Task management refers to the tutor's reasoning about the current task in which the student should be engaged. Rarely did our tutors just present a problem to the student and wait for him or her to solve it. At the very least, the problem is framed by some commentary or prompt. In most cases, task management is much more extensive.

Table 3 shows how task management typically breaks down the whole problem into a hierarchical set of smaller subgoals or tasks that jointly solve the problem. The top-level task is "do-problem." This decomposed to several "do-next-step" tasks. Notice that a step does not necessarily correspond to a visible inference, but rather to a reasoning step of variable size. The scope of a reasoning step may be a visible step (e.g., Lines 15 to 18), but may also be one of the several smaller decisions that implement a whole visible step (e.g., in Lines 20 to 24 the task of arriving at a single reasoning decision is completed, and Lines 26 to 30 use this decision to finish a visible step). Table 5 shows an even more extreme case, where nine exchanges (Lines 1 through 51) preceded the first written step. Overall, the number of exchanges per visible step ranged from one through nine, with an average of three.

Although some tasks may be smaller than a visible step, other tasks done by the student can also be much larger in scale than a typical single algebraic transformation. For example, the tutor may constrain the student to telescope several steps into one:

[Question is $(6x - 8 = -8)$]	
T: Okay. Now I'm going	
to put a problem like this,	
basically the same problem.	task (do-problem)
And this time let's try to do	task-intro [problem similarity]
it without putting these	task (do-next-step)
intermediate steps in. See	task-intro [do in head, prompt]
what you can do on that one.	
S: just directly	I
give you the answer?	
T: Yeah, Can you do that?	
	KNOWLEDGE-ASSESS [query]
s: Yes.	C+

TABLE 3
Simple Task Management Excerpt

T: Try 13. I'll read it: "$8 - 2x = kx$." [S writes $8 - 2x = kx$.] Again you have a problem where your xs are on both sides of the equation and you can make the choice which side you want your *xs* on.	task (do-problem)
	  task (do-next-step)
	    task-intro [problem similarity, state mid goal]

S: This side. [S points to right side.]	R +
T: That would be easier. [S writes $+ 2x$ below both sides.] Correct.	task-pa [correct]
S: $8 = kx + 2x$. [S writes new line $8 = kx + 2x$.]	task (do-next-step) R +
	———
T: That's good. Okay, what do you think you should do next?	task-pa [correct] task (do-next-step) task-intro [prompt]
S: Factor. [S writes $8 = x(k + 2)$.]	R +
	———
T: That's correct. And to get *xs* to totally stand alone what would you do?	task-pa [correct] task (do-next-step) task-intro [state high goal, prompt]
S: Divide.	R +
T: Correct. By what?	task-pa [correct] task (do-next-step) task-intro [prompt]
S: $k + 2$. [S writes $/(k + 2)$ under both sides.]	R +
	———
T: Uh huh.	task-pa [correct]
S: These cancel out. [S cancels out the $k + 2$ terms on right side.]	task (do-next-step) R +
	———
T: Uh huh.	task-pa [correct]
S: $x = 8/(k + 2)$ [S writes $x = 8/(k + 2)$.]	task (do-next-step) R +
	———
T: That's correct. Now do you feel you want more practice because . . . or do you feel comfortable with it?	task-wrapup [correct, confidence assess query]

Note. An example of an excerpt showing task management purposes. Indentation of tasks indicates conceptual nesting. For example, several do-next-step tasks are inside a do-problem task. The activities *task-intro*, *task-pa* (performance assessment), and *task-wrapup* are parts of the "script" of activities of managing a task and are nested under the task they comprise. The bracketed items following task-intro, task-pa, and task-wrapup markers indicate the specific technique used to accomplish this subpurpose. Dashed lines separate visible steps and the exchanges that comprise them. See Appendix A for a more detailed discussion of the coding of student responses.

	T: Okay.	
15	S: $6X$ equals	R+
	zero, so X equals zero	———
	(student writes $6x = 0$) ($x = 0$)	
18	T: Good	task-pa [correct]

Overall, when managing tasks for the student, the tutors were able to vary the bite size of the task from single reasoning decisions within an algebraic transformation to a decision that itself comprised many such transformations.

In many cases the desired scope of the task is explicitly communicated by the student by the technique chosen to introduce the task (task-intro). For example, at Line 3 in Table 3, the tutor's introduction loosely relates the problem to previous ones (problem similarity technique) and tells the student that the next task should just be to get all the variables on the same side of the equation (state mid goal). Different introductory techniques can have more or less constraining information, as we discuss next. At Line 16 in Table 3, for example, the tutor just uses a prompt technique, which implies little constraint.

The excerpt also indicates that task management has a loose, script-like quality. The idealized script for managing tasks of both wide scope (do-problem) and narrower scope (do-next-step), appears to have the following parts: task introductions (task-intro); task monitoring, in which the tutor checks progress and assesses performance (task-pa); and task wrapup or summarization, in which the tutor concludes one task and prepares for the next (task-wrapup). As Table 3 shows, different parts of the script can be deleted (e.g., do-next-step tasks are often not wrapped up), and the specific technique chosen to implement a script part can radically change how that part appears. Later sections that discuss specific techniques document this flexibility in greater detail.

Viewed more globally, the most salient property of the excerpts in Tables 2 and 3 is the remarkable intensity of the interaction. The discussion does not focus directly on obtaining the right answer or even on the correct next visible step in the solution (i.e., going from $x/3 = b/2 + c/6$ to $2x = 3b + c$. Instead, the student and tutor join in a discussion of the cognitive reasoning processes required to make the next algebraic transformation. This contrasts significantly with classroom procedures. The classroom teacher usually has far too little time to engage in microscopic discussions of the ideal reasoning for a problem. Rather, the teacher typically lectures the class on general techniques used to solve a particular class of problems. Only in a tutorial session does a teacher have the luxury of actually modeling ideal reasoning and coaching the student at the level of the cognitive processes that the tutor would like the student to learn.

Simple Tutoring Techniques

This section discusses many of the specific techniques we observed tutors using to accomplish different purposes outlined previously. As in the discussion of purposes, not all techniques are mentioned in detail. We first discuss simple techniques of relatively brief scope; later sections discuss more complex techniques.

Appendix B gives a complete list of all the important techniques that we observed. Each of the 44 entries names a technique, gives a brief description, notes the purposes the technique could be used for (some were multi-purpose), gives a count of the number of times the technique was observed, and mentions the topics or topics or cognitive objects that could be discussed using the technique.

Although performance assessment accounted for a high percentage of occurrences of techniques (38%; 135 of 354), tutors used relatively few techniques in giving students feedback about their performance. Only four distinct performance assessment techniques were noticed, accounting for 11% of all techniques (4 of 44). Typically, tutors either told the student his or her performance on the current task was acceptable or not acceptable, or they employed a "grain of truth" technique in which the student was given some credit but warned that some aspect of his or her task performance was at least strategically dubious, if not mathematically invalid, for example:

[Question is: Solve for X in $x/b - c = a$]	
T: What do you think I'm	REMEDIAL [state low goal]
3 suggesting you eliminate then	
on the left hand side?	
[S begins to write $a/1$ under the	R –
c.] That's right. What's the opp...	task-pa [grain of truth]
7 Well thats good... $c/1$ is nice...	REMEDIAL [explain bug]
See these are two separate terms,	
right? [T points to the $-c$ and x/b.]	

Here the student had already made an error in executing the next step, and the tutor was attempting to remediate (Line 2). The first remedial attempt appears to have failed, because the student is now trying to divide by 1 (Line 4). In response to this action, the tutor's performance assessment implies the student is incorrect but hedges (grain of truth), possibly to maintain the student's affect after several errors.

The preponderance of techniques we catalogued implemented task management or remedial purposes, and many were used for both purposes. The overlap is accounted for by the fact that many techniques can serve not only to remediate a student error after it has happened, but also to coach a student before the student makes an error (task-intro subpurposes of task management), or even to wrap up a task after it has been accomplished (task-wrapup). Forty-five percent (20 of 44) of all techniques could be used for task management purposes; we observed 24 distinct techniques that tutors used to remediate student errors. Remedial techniques accounted for only 15% (53 of 354) of all occurrences of techniques, limited by the total number of errors made by all students (40). Unlike performance assessment, however, one or two techniques did not dominate—Tutors used a diverse collection of remedial techniques.[2]

Techniques Discussing Problem Conditions, Procedures, and Concepts

As the table in Appendix B indicates, remedial and task management techniques can be distinguished in terms of the kinds of problem features and cognitive objects they discuss. *Problem conditions* refer to properties of the current algebraic expression that should be attended to in deciding on the next thing to do. Several different techniques made use of problem conditions. Using a problem description technique, a tutor simply draws the student's attention to some salient part of the question and away from irrelevant parts. The problem similarity and problem difference techniques extend problem description. In addition to, or sometimes in place of, pointing to the relevant features of a problem, the tutor referenced similar (or contrasting) past problems. Perhaps the motivation behind this technique is to help students learn a metacognitive skill (when you are stuck, look for similar problems you have solved in the past to help you decide what to do now), in addition to giving them information that might be useful in solving the current problem.

Tutors also frequently gave task management hints or remedial help by discussing or naming the procedures relevant to the current question. For example, in the following brief excerpt, the student makes several false starts (Lines 7, 10), and at Line 16 the tutor suggests the factoring procedure. The student immediately succeeds.

[S writes: $mx - px = -t$]	
T: Okay. Now you've got two	
terms with x in both terms.	task (do-next-step)
Right. We still need x all by	task-intro
itself. Do you have any idea	[state hi goal, prompt]
how we can get x all by itself?	
S: By dividing.	R?

[2]The number of instances of remedial techniques (53) is greater than the number of student errors (40), because tutors sometime apply several techniques to one error. The total number of technique applications (354) is different from the total count of purpose instances (376), because some purposes (e.g., problem generation) do not result in a technique being used.

T: By dividing. What do you mean by dividing?	KNOWLEDGE-ASSESS [clarify]
S: If I divide this side by x and this side by x, you see? And then both sides equal out.	R −
T: Okay.	
S: So 1 x would be left.	
T: Okay. You could—how about factoring, do you know how to factor?	task-pa [grain of truth] REMEDIAL [suggest right procedure]
S: Yes. I would. . . .	R +
[S writes: $x(m - p) = -t$]	———

Mathematical concepts were also discussed, using several techniques, about as frequently as procedures ($n = 3$ for concepts, $n = 4$ for procedures). For example, in Table 4 at Line 21, the student and tutor first mention the concept of the additive inverse, then attempt to use it in relative isolation (Lines 23 to 25), and finally they attempt to apply it to the current problem (Lines 27 to 28).

Techniques Discussing Goal Structures

Perhaps the least familiar kinds of topics in reasoning discussions were *goal structures*. They were also the most frequently referenced cognitive objects ($n = 37$) in techniques that introduced tasks and that provided remedial assistance. Goal structures refer to the hierarchical "tree" of decisions that is generated when one solves an algebra problem by "problem decomposition": decomposing the main goal of solving a problem into successively smaller goals until each goal can be executed directly. To understand the many reasoning discussions that involve goal structures, we have explicitly encoded the goal structures needed to solve first semester algebra problems. This process was relatively simple, because the goal structures that suffice to analyze tutorial discussions are virtually identical to the goal structures used by the algebra "expert system" embedded in our algebra tutor (McArthur et al., 1987). Figure 2 gives an abbreviated representation of the goal structures for the excerpt in Table 2. Bundy and Welham (1980) also provided a similar vocabulary for describing such algebra symbol manipulation goals.

The excerpt in Table 2 exemplifies several important points concerning how goal structures are used by tutors in many interactions to guide reasoning discussions. First, the goal discussion begins in a top-down fashion. In discussing the goals for the next step, the tutor does not begin with a low-level description of the mathematical transformation to effect. Rather, she begins (in Lines 6 to 8) by mentioning a middle-level goal, ***eliminate fractions***, that should be achieved. Moreover, several intermediate levels in the

TABLE 4
An Excerpt With Discussion of Additive Inverse Concept

[T writes down equation "*mx* − 4 = 2*a*".]	task (do-next-problem)
T: Okay, what we have here is "*mx* − 4 = 2*a*. And the reason I asked you prior to this is you had any difficulty solving equations because the procedure we're going to use here is exactly like everything you've done before. Before you came with two variables. The first thing you want to know . . . it says solve for *x*. This is the object, or you might say the *variable* you want to isolate. [T boxes the *x*.] You want to get it all by itself. Therefore, your objective is to get rid of everything about it. You'll notice the 4 here. [T points to the 4.] Do you know how to eliminate the 4?	task (do-next-step) task-intro [problem similarity, state high goal, state mid goal, state low goal, prompt how]
S: Yes, subtract it from both sides.	R −
T: Okay, when you say "subtract it," have you ever heard of something called the *additive inverse*?	REMEDIAL [apply-concept] KNOWLEDGE-ASSESS [query]
S: Yes.	C +
T: What is the opposite of −4?	task (local-concept-use)
S: +4, positive	R +
T: Okay let's add the additive inverse,	task-pa [correct]
add +4 to both sides. Why don't you do that?	task (map-concept) task-intro [prompt]
[S puts +4 below both sides of the equation.]	R +

Note. Lines 20 to 30 show the tutor and student applying the concept of the additive inverse. In this complex remedial technique, the concept is first introduced through a query (Line 21), then exercised in isolation (Line 25), and finally mapped onto the current problem (Line 27).

goal structure may be mentioned before the tutor begins to discuss the actual algebraic manipulations. Here, for example, a lower level subgoal, "multiply by least common multiple," is also discussed as part of the task-intro phase for this step.

Second, because of the richness of the goal structures, tutors can choose among a variety of levels of support for the student. The beginning monologue of Table 4, for example, shows an extremely detailed top-down discussion of goal structures. Sandwiched between a description of problem similarity and a prompt to the student is the tutor's description of the full range of high-, middle-, and low-level goals that should be considered in the problem. At the other extreme, tutors often introduce a task with just a prompt:

[T writes $5x - 7 = 7$] T: How would you tackle a problem like that?

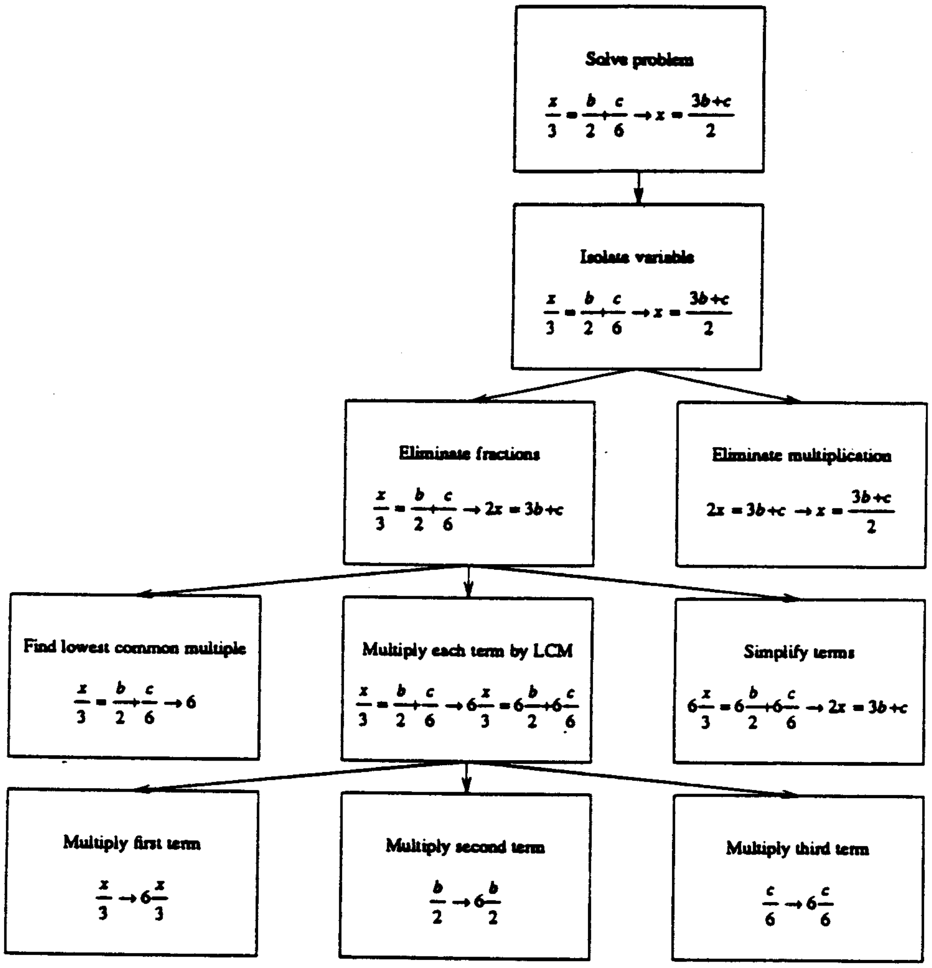

FIGURE 2 Hierarchical goal structures for tutoring excerpt. (The figure shows a simplified version of the goal structure needed to generate the problem-solving step found in the transcript excerpt in the text. Higher level goals are connected to its subgoals by arrows. When a goal has several subgoals, they are interpreted conjunctively; that is, they all must be achieved to satisfy the higher level goal. Below the name of each goal is a pair of algebraic expressions describing the goal's effect. The input expression to the goal is on the left. The goal's output follows an arrow (→), on the right. Several lower level goals, which are not discussed in the text, are omitted from the figure.)

The goal structures permit tutors to vary the relative contributions of the tutor and student in deciding and satisfying each goal. For each task introduction, tutors must decide how far down the hierarchical goal structure they will cut off their descriptions. The further down the tutor goes, the less the student must do, and the more support they give. In Table 2, the goal to *eliminate fractions* is simply given by the tutor, as is the subgoal "Multiply by least common multiple" (Lines 10 to 12). On the other hand, "Find lowest common multiple" is prompted for by the tutor but is satisfied by the student (Lines 13 to 15). Similarly the goal *simplify terms* is shared by the student and tutor, with the tutor mentioning the terms to divide and the student simply carrying out the divisions and writing the results (Lines 24 to 45). In addition, tutors often decide to abbreviate their reasoning descriptions by leaving out some goals in the hierarchy. Unfortunately, in some cases high-level goals are eliminated, and students and tutors focus on low-level subgoals directly involved in computing an algebraic transformation. This policy promotes a "cookbook" problem-solving approach in which the student is drilled on what to do without mentioning why it is being done.

Third, once the tutor and student have collectively discussed the goals and implemented them, arriving at a new step in the solution (Line 48 in Table 2), the tutor does not simply go on to the next step but reminds the student of the reason for the low-level multiplications and simplications, namely, to get rid of the fractions (Lines 55 to 57). In other words, once the student and tutor have proceeded down the hierarchical goal structure for this step, the tutor again focuses the discussion on higher-level goals. This time the intent behind discussing the high-level goals is not to help the student determine which lower level actions to do next, but rather to provide a justification for the reasoning that has just been accomplished. Thus, goal structures prove useful in task-wrapup phases of task management as well as in task-intro phases.

Finally, as a general point, it is interesting to note that in reasoning discussions the tutor and student talk about problem-solving goals for which no formally specified vocabulary exists. Algebra texts provide a formal vocabulary for things like the axioms of mathematics. Yet these concepts comprise remarkably little of the tutorial interactions. Much more commonly mentioned are concepts such as isolating the variable, attracting occurrences of the variable closer together, and eliminating terms. The vocabulary students and tutors share to talk about such problem-solving goals appears to arise tacitly and informally. Unfortunately, we have observed that this can result in terminological misunderstandings that foil the tutorial exchange. For example, one tutor systematically used the phrase "isolate the variable" to denote what we mean when we say "collect like terms" (e.g., transforming $2x + 3x = 5$ to $5x = 5$).

Complex Tutoring Techniques

The aforementioned techniques are of limited scope because their effects on the tutorial process are temporally brief. For example, a simple remedial technique is usually completed in one exchange, as is the introduction of a task or query that assesses the student's current knowledge. However, our analysis of the tutoring sessions also indicates the existence of decisions that have a scope longer than a few exchanges, possibly as long as several problems. In this section, we describe some of these patterns and discuss the kinds of cognitive structures that might account for them.

Simple remedial techniques typically interrupt the flow of task management for one or two exchanges, then relinquish control (see, for example, Lines 31 to 32 of Table 2). By contrast, other remedial techniques can generate a series of actions, not just one. We refer to such techniques as *complex*. Appendix B lists the complex remedial techniques we observed. Complex remedial techniques were used frequently. Fifty-three percent (8 of 15) of all remedial techniques were complex, whereas 30% (16 of 53) of all occurrences of remedial techniques were complex. As the examples in the table suggest, we regard such complex techniques as implemented by scripts similar to those characterizing task management. The tutor implements a complex remedial technique by interpolating a sequence of tasks into the ongoing flow of task management.

Table 5 gives an extended example of complex remedial techniques in action. After a couple of false starts by the student (Lines 3 to 7), the tutor invokes a complex remedial technique referred to as goal-given-feature. It is complex in the sense that it decomposes into several subtasks: first, orienting the student to the variable in the problem (find-variable); second, finding the relevant feature of the problem involving the pattern of variable occurrences (find-feature); and finally, choosing the correct action, given the feature (step-given-feature). In this case, the first two subtasks are introduced and accomplished successfully (Lines 12 to 20). However, the student fails to accomplish the final subtask after being prompted (Line 21).

Having failed with this remediation, the tutor initiates another complex remedial technique, map-similarity (Line 22). The script for this technique involves three subtasks: finding a similar problem, finding the specific operation in that problem that is relevant, and mapping the operation into the current problem. The tutor begins this remedial interaction by initially skipping the step of finding a similar problem (Line 23) and introducing the task of finding the operation from a past similar problem. However, when this fails (Line 26), the tutor reintroduces map-similarity, beginning at the start of the script (Line 28). The subtask of finding a similar problem is done by the tutor herself (Lines 30 to 32). Now, with the similar problem visible, the student's first attempt to find the operation is wrong (Line 33), but with the interpolation of a simple remedial technique (try-again), the student is successful (Lines 37 to 38). Having completed two of the three subtasks of map-similarity, the student's first attempt at the final subtask (map-operation) is immediately correct (Line 39). Finally, the tutor assesses the student's performance on both the nested remedial task (Line 43) and on the completion of the step as a whole (Line 47).

This example illustrates a number of important points about complex

TABLE 5
Example of Complex Remedial Techniques in Use

[Writes: $ax = -b - x - c$]	———
	task (do-next-step)
S: So, now I want to get rid of the a.	R –
T: What . . . . Umm. Let's see	
S: Divide.	R –
T: If you divide right now by a . . . check your terms Look at these terms	REMEDIAL [goal-given-feature]
again. What do we have here.	task (find-variable)
What are we solving for?	task-intro [prompt]
S: X.	R +
T: Where are the xs in	task (find-feature)
the problem?	task-intro [prompt]
S: Here and here. [Points to xs on each side of the equation.]	R +
T: Okay.	task-pa [correct]
S: So you want to get	task [step-given-feature]
[Short pause.]	Ro
T: So what did we do in	REMEDIAL [map-similarity]
the other problems when we	task (find-similar-operation)
had xs on both sides of our equal sign?	task-intro [prompt]
S: [Not clear. Maybe "I'm lost here."]	R ¯
T: Okay. Let's look at one of the problems we did	task (find-similar-problem)
before. Here was a problem	task (find-similar-operation)
with xs. [Shows paper with previous question.]	task-intro [prompt
S: We factored.	R –
T: Right. But before we	task-pa [grain of truth]
factored, what did we do with our xs?	REMEDIAL [try again]
S: We brought both of them on the same side.	R +
T: Right. Okay.	task-pa [correct]
	task (map-operation)
S: Okay. So I put x here.	R +
T: Right.	task-pa [correct]
[S Writes left-hand side of equation: $-ax + x >$.]	R +
T: Yes.	task-pa [correct]

(Continued)

TABLE 5 *(Continued)*

48		
	S: Plus x . . . minus b minus c.	R +
51	[S Finishes writing equation: $-ax + x \triangleright b - c$.]	----

techniques and the script-like nature of tutoring in general. First, because complex remedial techniques often generate several tasks, not only are remedial purposes interpolated within task management (when errors occur), but they also generate task management purposes. For example, using the complex technique map-easier-problem, we have observed tutors generate several simpler problems, complete them, and successfully resume a suspended question. As a consequence of such techniques, the nesting of purposes in tutorial reasoning can be very deep, indicated by the different levels of indentation in Table 5. We believe this reciprocal nesting of task management and remedial purposes and techniques accounts for much of the richness and diversity observed in tutors' behavior.

Second, tutors' implementation of complex remedial techniques shows much of the flexibility that we observed in their execution of task management scripts. In Table 5, the tutor exemplifies several kinds of deviations from the prototype scripts listed in the table in Appendix B. Tutors may decide to complete some subtasks of a remedial script themselves (Line 28, Table 4). Similarly, they may omit subtasks (Line 23). If a subtask of a remedial script is not completed successfully, tutors may either exit the whole script and try another technique (Lines 20 to 22), or they may persist with the subtask and interpolate yet another remedial technique (Line 35), or they may back up and insert a subtask that was initially omitted (Line 28). Finally, as with task management scripts, much of the flexibility in implementing a script derives from the tutor's wide choice of techniques for introducing a task. As discussed earlier, tutors can supply either much or very little information when introducing a task.

Process implications of complex techniques. Although simple tutoring techniques are executed and then forgotten, the task management and remedial scripts analyzed earlier clearly indicate that many tutorial decisions persist for extended periods. We suggest that when tutors select a technique or purpose to execute from their knowledge bases, in cases where that technique implies a script of activities, the script is executed by creating a *microplan*. In cases where a technique involves a single action, a plan is not necessary, because the action can be done immediately. A microplan is an instantiation of the script in active memory that permits the tutor to remember a series of actions that must be executed in the future. For example, when executing the technique map-easier-problem (see Appendix B), the tutor will instantiate the script as a microplan that says first do-easier-problem (create and do an easier problem with the student), then map-problem (return and apply the lesson learned with the interpolated simpler plan to the original problem). Assuming the tutor has no micro-plans in active memory preceding this instantiation, the first element of the new micro-plan becomes the next action and generates the current task for the student and tutor to accomplish, and the second element is retained in active memory until the first one is completed.

This kind of planning is relatively straightforward; however, the preceding analysis of task management and complex remedial techniques suggests that additional sophistication is common. The execution of a given micro-plan may be modified by deleting or adding various elements. Alternatively, one micro-plan may be dropped, and its place taken by another. Finally, in the course of executing one plan, if difficulties arise, the tutor may suspend the plan and insert several others, returning to the original plan only when the interpolated ones are complete.

Constraints on the Selection of Techniques

The flexible way tutors implement scripts and the many techniques available to tutors to introduce a task, or to remediate a mistake, raise the issue of how tutors choose among the techniques at their disposal. Previous sections have simply described scripts and catalogued different purposes and techniques. In this section, we attempt to examine several of the factors that may play a role in tutors' choices among their options.

Many of the events and conditions that determine the invocation of tutorial purposes appear clear cut. For example, students' mistakes always trigger remediation, and the completion of one task always triggers a new one. But what conditions are attended to in deciding the specific remedial technique or in determining the technique for introducing the task? We first discuss some of the factors that appear to constrain the selection of task introduction techniques, then look at some factors determining the selection of remedial techniques. In neither case is our analysis exhaustive. Given our relatively limited data, we have examined the roles of only the most salient conditions.

Constraints on Task Management

Figure 3 graphically shows the level of support given by the three tutors across time when introducing tasks. The levels of support (y-axis) have been arranged into an ordinal scale, based on the principle that more detailed information about the student's goal gives greater support. Prompts (s) give essentially no information. A tutor mentioning only high level (h) goals of structures, as in Figure 2, gives relatively little support. Mention of lower level (l) goals gives greater support, because they more explicitly describe

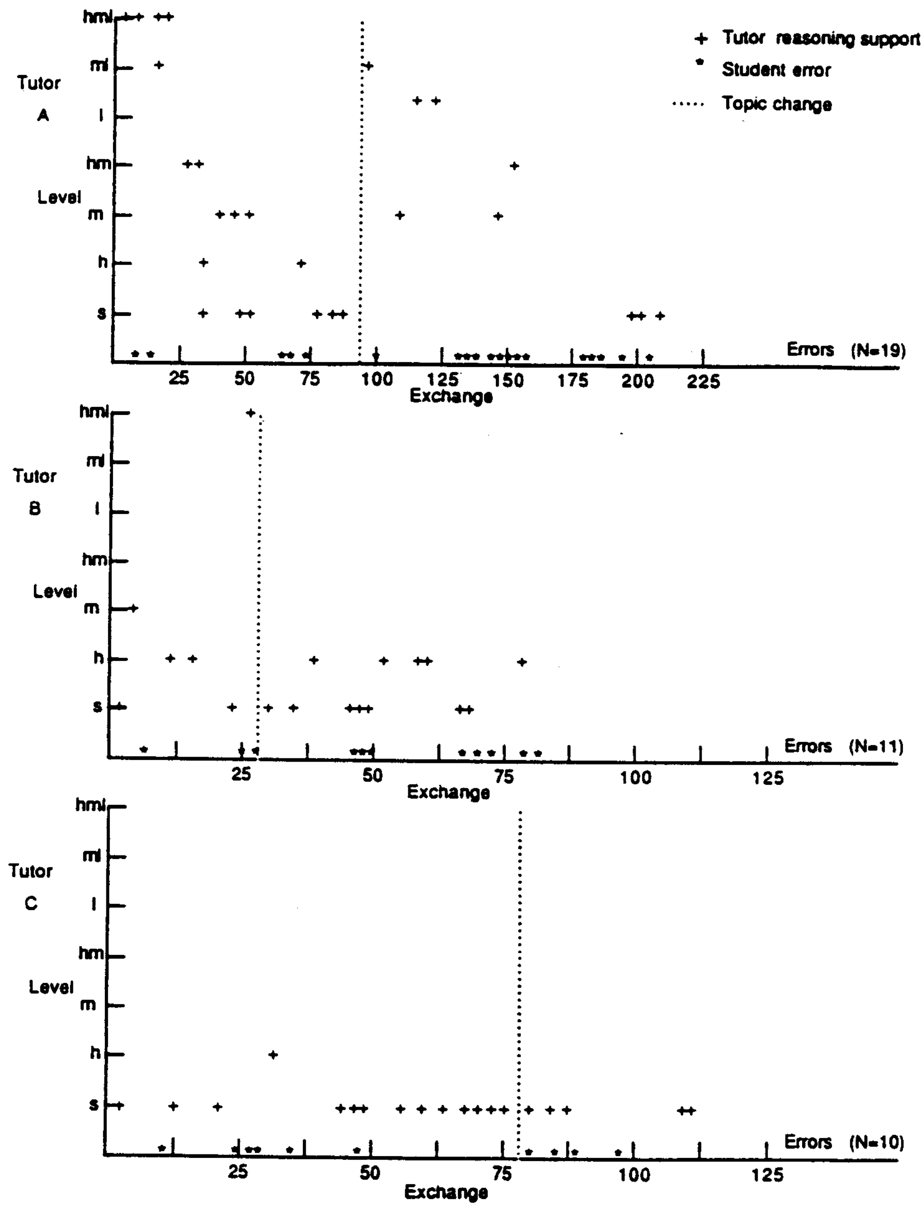

FIGURE 3 Reasoning support as a function of time in tutoring sessions. (Tutor A shows declining levels of support through both topics; Tutors B and C show a more constant low level of support across exchanges. Overall, student errors (*) appear unrelated to level of support.)

the kinds of manipulations required of the student. Finally, mention of goals at several levels (e.g., *hml*) gives the greatest support.

The patterns in the graphs suggest that the tutors' level of support is not random. The first graph in Figure 3, for example, shows Tutor A beginning with a very high level of support. The excerpt in Table 4 is an example of the high level of support initially given by this tutor (*hml* or *ml*). By approximately Exchange 50 in the transcript, the level of support has dropped to an asymptote where in most cases either the student does the task after prompting by the tutor, or the tutor describes only high-level goals (*m*, *h*, or *s*). Near 100, the tutor's level of support again jumps to high and declines to a low asymptote by the end of the session. In contrast, Graphs 2 and 3 in Figure 3 show that Tutors B and C have a very different pattern of support. With only one exception, these tutors supply little support when introducing tasks, often simply prompting the student.

Hypotheses concerning the reasons for these different patterns of support cannot be firmly substantiated, given our limited data. Nevertheless, we can offer some compelling conjectures, buttressed by existing views of tutoring expertise. Our first conjecture is that Tutor A is following a policy of "modeling-scaffolding-fading." She begins by supplying extensive support, often doing (modeling) the correct reasoning herself. She continues to provide considerable direction (scaffolding the student's problem solving) and progressively withdraws more and more support (fading). On this interpretation, the tutor's behavior appears intuitively reasonable. Indeed, Collins, Brown, and Newman (1989) extensively documented this style of tutoring, discussing in much more detail its merits as part of a general model of cognitive apprenticeship. Based on our data and this general view of tutoring, we posit that modeling-scaffolding-fading is a tutorial policy that tutors can deliberately adopt. In contrast, Tutors B and C appear to be following a policy we refer to as *constant-coaching*. Their behavior during the tutoring session, together with comments in the postsession interviews, confirms that Tutors B and C deliberately supply little information when introducing a task, instead preferring to deal with student difficulties when errors arise.

The graphs of Figure 3 suggest other factors may also modulate tutors' introduction and management of tasks. In particular, the graph for Tutor A indicates that the cycle of modeling-scaffolding-fading does not span the whole tutoring session, but rather begins and ends twice. For Tutor A, the beginning of such a cycle is perfectly correlated with the commencement of a new topic in the tutoring session (shown by the vertical dotted line in the graphs of Figure 3). For Exchanges 1 through 88, the questions posed to the student concern simple factoring concepts. At Exchange 89 the concept of solving linear equations with fractions is introduced. When the first question involving this concept begins, the tutor assesses the student's knowledge of the concept, indicating that the tutor is aware of a topic change. Subsequently, the tutor's support returns to a high level and then declines over the next several questions.

A final interesting observation concerning Figure 3 is the apparent lack of sensitivity of tutors' introduction and management of tasks to student errors. Errors are shown in Figure 3 as asterisks at the exchange where the error occurred. An obvious prediction might be that the tutor's level of support would increase following student errors and decrease only after students succeeded in accomplishing tasks without errors. Although Tutor A shows one case of fading after a long period without errors (Exchanges 10 to 50), we also see cases of fading support during a period of many errors (Tutor A, Exchanges 125 to 200) and several cases of relative indifference to student errors (Tutors B and C).

Process implications of task management constraints. In general, tutors appear to attend to several kinds of factors in deciding how to introduce and manage reasoning tasks for the student. In our process model, these constraints take the form of conditions represented in the tutor's active memory. At least two kinds of active memory conditions appear necessary to account for the data in Figure 3. First, we posit that tutors remember a current tutorial policy, which is consulted by tutors to constrain their selection of techniques for introducing the managing tasks. Second, we suggest that important mathematical topics or concepts become K goals. K goals are representations of particular topics in the tutor's active memory that record the knowledge or concepts the tutor is currently trying to communicate to the student. We speculate that K goals play several roles in tutoring. In addition to modulating the choice of techniques for introducing and managing tasks, they also play a role in remediation, which we discuss later. Several other roles are possible. For example, K goals may be used to generate new questions for the student. Tutors often generate questions, which, if answered correctly, should elicit the concepts that are current tutorial topics (e.g., McArthur, Stasz, Hotta, Peter, & Burdorf, 1988).

Constraints on Remediation

Our analysis of the constraints on selection of remedial techniques was guided by initial observations that the nature of remediation chosen by our tutors appeared to depend on several properties of the students' errors. These properties included *causality* (the imputed reasons for the student's error), *topic* (whether the error pertained to a tutorial K goal or not), and *context* (whether the error was the first mistake on a task or not). Accordingly, we classified student errors into several categories:

1. *Slip.* On the basis of student's past and subsequent performance, the error appeared to be a simple slip.
2. *Knowledge error, lesson topic.* On the basis of the student's perform-

ance, the error appeared to be due to a lack of knowledge or flaw in knowledge. We limited this category to errors in knowledge that pertained to the current topics (K goals) that the teacher was attempting to communicate to the student.

3. *Knowledge error, not lesson topic*. The error appeared to be due to a limitation in the student's knowledge of topics that were not the current focus of attention.
4. *Nested error*. The error was not the first one the student made in attempting to complete a task, or was nested in a remedial task that was itself triggered by an error (e.g., Table 5, Line 35). This category dominates the previous three. Regardless of cause, any nested error was classified here.

These error features did not appear to determine the selection of specific remedial techniques, but rather limited the choice to a general kind of technique. One useful division was to differentiate *performance-correcting techniques* and *knowledge-correcting techniques*. Performance-correcting techniques aim mainly to help the student accomplish a particular task on which the student has previously failed. Putnam's (1987) finding is that the preponderance of teacher remediations are of this type. On the other hand, knowledge-correcting techniques appear to have two goals. In addition to helping the student accomplish the current task, they take the error as an opportunity to teach the student a piece of knowledge relevant to accomplishing the current task. Using this basic division, we developed four categories into which all the remedial techniques from Appendix B were classified:

1. *Simple, noninformational*. Simple techniques that tell students they have made an error but give no guidance about correcting it (e.g., try-again, continue).
2. *Simple, task terminating*. Simple techniques that tell the student what the right reasoning is for the current task (e.g., suggest-right-procedure, give-answer) or that make it unnecessary for the student to have to accomplish the task (e.g., change-problem).
3. *Simple, informational*. Simple techniques that discuss aspects of the correct reasoning or conditions to attend to in order to reason correctly, but which do not complete the task for the student.
4. *Complex*. All complex techniques are informational in the just-mentioned sense.

Generally we regard the first two categories as mainly performance-correcting techniques, because they do not use students' errors as a basis for improving their abstract understanding of concepts, goals, procedures, or

TABLE 6
Remedial Techniques Classified by Error Type

Error Category	*Technique Category*			
	Simple Non Informational	*Simple Task Terminating*	*Simple Informational*	*Complex*
Slip	9	1	1	0
Nested	1	9	1	2
Knowledge error, nontopic	0	4	0	0
Knowledge error, topic	0	2	9	14

Note. Columns represent a classification of remedial techniques. Each remedial technique was classified into one category. Rows represent a classification of student errors. Each cell contains a count of the number of remedial techniques of a particular type that followed a particular error type. The total in all cells (53) is greater than the total number of errors observed, because some errors were followed by more than one technique.

conditions of their application. The latter two categories indicate that the tutor intends to impart such knowledge.

Our analysis of the constraints on selection of remedial techniques was limited to relating the previously discussed classification of error types to the class of remedial technique that followed them. Table 6 shows, perhaps not surprisingly, that responses to slips were almost always noninformational. Tutors appear to assume that slips require only redirecting the student's attention. Nested errors were usually followed by task-terminating techniques. That is, if students made a second error on task, tutors usually brought that task to an end immediately. Tutors apparently were not willing to let students' errors get "out of hand." Perhaps more interesting, tutor's treatment of knowledge errors was radically different, depending on whether the error involved a current topic or K goal. Errors pertaining to current K goals were always dealt with using information techniques; errors not related to the current K goals were never dealt with this way. In other words, our tutors largely reserved knowledge correcting techniques for non-nested errors on currently important topics.

Process implications of remedial constraints. As with task introductions, tutors appear to attend to several factors in deciding how to remediate student errors. K goals that tutors represent in active memory seem to play an important role here, as well as in constraining ask management. In addition, some rudimentary student modeling or student diagnosis also appears important. Our tutors at least seem to attempt to distinguish accidental slips from mistakes that imply knowledge errors, although they do not appear to infer any specific bug that might underly the mistake. Finally, in addition to conditions, the history of tutor/student events (see Figure 1) plays a role in selecting remedial techniques. Tutors can determine if an error is nested only by reference to the local history of student responses.

DISCUSSION

In this section we discuss our findings along several dimensions. First, we discuss the limitations of our results. Next, we note several implications of our process model of tutoring and compare our findings and process model to other views of teaching and tutoring. Finally, we discuss some possible implications of our results for computer-based tutors and for teacher training.

Limitations

There are several features of our study that both suggest limitations on the generality of our current findings and suggest future directions for research in the study of human tutoring skills. First, the kind of one-on-one tutoring we examined is not representative of all tutoring interactions. Remedial tutoring, centered on working through problems that students have previously failed, is certainly very common. Indeed, almost by definition, tutoring connotes a remedial image; someone being tutored in a subject is often presumed to be doing poorly. However, in inquiry tutoring, students are generally learning new concepts rather than fixing old ones. The fundamentally different goals of remedial tutoring and inquiry tutoring make it likely that we will find significantly different kinds of activities in each (see, e.g., Collins, 1988, on different goals of inquiry teaching). Clearly, any model of tutoring that attempts to span both remedial and inquiry tutoring skills will have be to be based on intensive studies of experts in each area.

In addition to the kind of tutoring, we believe that several other variables need to be systematically studied to arrive at a more general model of tutoring. Among the more important variables are student–tutor familiarity and tutor practice effects. Two of our tutors had never met the student they tutored, and although we did not notice systematic differences attributable to familiarity, in general, prior knowledge of the student may play a role. For example, R. T. Putnam (personal communication, May 1989) suggested that tutors who are familiar with their students may be more likely to use curriculum scripts. Similarly, Clark et al. (1976) observed that tutors' performance in physics tutoring appeared to improve across several sessions with different students. Thus, it is likely that the tutor's familiarity with the topic, with the bottlenecks in learning it, and with the specific tricks for teaching it may be as important as the tutor's history with the student. Both variables require further examination before their implications for our current model of tutoring can be assessed. We are currently engaged in a study that focuses on inquiry tutoring in which tutors teach several different students.

Planning and Problem Solving in Tutoring

As a study of remedial tutoring, our findings and model suggest that tutoring is largely, but not exclusively, data driven. That is, it is primarily triggered by events that arise in the tutoring session (e.g., a specific student error) and is less governed by enduring constraints (e.g., a lesson plan). For example, remedial purposes are invoked only (and always) when students make errors. A second and related point is that the scope of most tutorial decisions is brief or local, where the scope of a decision is defined as how long (in terms of utterances and exchanges) a decision continues to have an effect on tutorial performance. We define tutorial decisions of local scope as *tactical* ones; thus, our model suggests most tutoring is both data driven and tactical. The tactical character of tutoring is well illustrated by simple remedial techniques. They are triggered by student mistakes, generate a simple intervention, and then terminate.

Several aspects of our data suggest tutoring is goal driven as well as data driven. As we have described when analyzing task management and complex remedial techniques, we posit that events may not only trigger immediate tutor responses but may also trigger scripts and microplans or changes to microplans. These microplans may endure for several exchanges and may even influence tutorial performance across problems. K goals and policies that the tutor represents in active memory impart additional continuity across time in tutoring. Although scripts dictate relatively well-defined sequences of activities over several exchanges, K goals and policies dictate less specific constraints on the specific actions taken by the tutor but generally act over longer periods of time. Thus, we believe that tutorial decisions can be *strategic* as well as tactical, where a strategic decision is one whose scope extends across several exchanges or event problems.

Perhaps the most important conclusion we can draw from our analysis is that the reasoning involved in tutoring is subtle and sophisticated. In the past, teaching/tutoring has not been regarded as a skillful or knowledge-intensive profession as, for example, physics (Berliner, 1986). We believe this illusion has persisted largely because the expertise good teachers possess has not been systematically formalized. Our preliminary results have begun to expose several dimensions in which teachers and tutors can be regarded as skilled practitioners.

First, judging from the table in Appendix B, competent tutors possess extensive knowledge bases of techniques for defining and introducing tasks and remediating misconceptions. Viewing a tutor as an expert system, the number of pedagogical rules our tutors possess may approach that of expert systems that capture human expertise in fields such as medical diagnosis (Shortliffe, 1976) and computer configuration design (McDermott, 1982). Admittedly, those systems may not account for the full range of human skills in their respective domains; however, our list of tutoring techniques is no doubt equally limited.

Second, we have seen that tutors are capable of taking a variety of different events and conditions into account when selecting from their diverse array of techniques. Continuing to view tutoring techniques as expert-system rules in the tutor's knowledge base, we can say that the antecedent conditions of those "if-then" rules are often nontrivial.[3] Rather than associating some relatively simple event with a fixed response, factors such as K goals for the student, inferences about the student's knowledge, overall pedagogical policy, and local history of events appear to modulate the selection of techniques in ways we have only begun to clarify.

Finally, perhaps the most important dimension of expertise we have observed in tutoring involves planning. Not only do tutors appear to formulate and execute microplans, but also their execution of a given plan may be modified and pieces deleted or added, depending on changing events and conditions. These activities are instances of an ability to dynamically replan in response to changing circumstances (B. Hayes-Roth, 1984; B. Hayes-Roth & F. Hayes-Roth, 1979). Overall, the kinds of planning in skilled tutoring appear to be as complex as in many cognitive domains that have been subjected to detailed analysis (e.g., Larkin, McDermott, Simon, & Simon, 1980). Indeed, because tutors plan in an uncertain environment with several agents (i.e., the student as well as the tutor), dynamic planning in tutoring may be inherently more demanding than in many domains of perfect information. Thus, in accord with Ohlsson (1986), we see teaching and tutoring as true problem-solving activities.

Comparison With Other Models of Tutoring and Teaching

Our findings both support and contradict previous research. Like Putnam (1987), we find that a diagnostic/remedial approach to tutoring cannot account for many of our observations. Although it is clear that tutors do make general inferences about a student's level of skill, we found almost no evidence that specific misconceptions or bugs were diagnosed. Consistent with this finding, we also observed that much of the tutor's decision making appears, perhaps surprisingly, insensitive to inferences about the underlying causes of student errors. For example, the level of coaching or scaffolding of the tutor was not related in any obvious way to students' errors (see Figure 3).

Although our data underscore limitations of the diagnostic/remedial model of tutoring, we also wish to point out that in such a model diagnosis has often been too narrowly construed as equivalent to simply detecting errors or bugs in students' knowledge. However, there are other senses in which students' knowledge can be incomplete. In particular, it is often as important for tutors to detect "gaps" in knowledge as to find bugs. Still,

[3]Alternatively, we might say that the *conflict resolution* algorithm (Langley & Neches, 1981) for tutoring is quite sophisticated. Conflict resolution is the process by which an expert system chooses among several different rules, each of whose antecedents match current conditions. In tutoring, for example, conflict resolution might decide which remedial rule (technique) to fire from the repertoire of rules enabled when a student makes an error.

our data indicate that much tutoring is spent in nondiagnostic activities, either broadly or narrowly construed.

Contrasted with a diagnostic-remedial model of tutoring, our analysis of task management purposes suggests a much more "coaching" view (Burton & Brown, 1982). Tutors appear to spend as much effort structuring the task of problem solving for the student as they do critiquing students' weak performance. At least our tutors appear to place a premium on setting up tasks so that they are at the right level—neither too difficult for the student nor too trivial. This approach in many cases appears actually to minimize the production of errors, which, according to the diagnostic-remedial view, would reduce opportunities for important tutoring.

More generally, a simple diagnostic-remedial view of tutoring appears too one-dimensional to account for much tutoring behavior. It views the diagnosis of student errors and their remediation as the primary activities of tutoring. To the contrary, we see tutoring as comprising many additional activities, including task management and tutorial planning. Tutors' overall skill does not appear dominated by diagnostic abilities. Rather, it depends at least as much on diverse pedagogical expertise, including a repertoire of tutoring techniques and rules for using them.

As an alternative to a diagnostic-remedial model of tutoring, Putnam (1987) proposed a "curriculum script" view, which argues that tutors' choices of actions are largely predetermined by sequence of problem types, rather than by students' local errors. Putnam acknowledged that tutors deal with student errors but suggested that their diagnosis is often cursory. In the curriculum script model, the overall macrostructure of the lesson is determined by the script, whereas only the microstructure is constrained by the student model and student performance cues.

Like the curriculum script model, our view sees tutoring as a more planful process than does the diagnostic-remedial model. However, our microplans and scripts differ in several respects from Putnam's (1987) curriculum scripts. First, the planning that Putnam referred to is relatively rigid compared with our microplans. The plans implied by a curriculum script appear relatively impervious to changing conditions, whereas microplans are constantly changing in response to new situations. More important, the microplans we describe are closer to the level Putnam referred to as microstructure. The planning we have discussed occurs largely within a single problem, not across many problems. Little of our analysis focused on Putnam's macrostructural level of tutoring, possibly because our tutors were teaching students in a remedial context and, thus, may not have brought to the lesson a predefined set of problem types with which to structure the session.

Our view of tutoring may be seen as a midground between a simple diagnostic-remedial model and a curriculum script view. We believe tutoring is both opportunistic (driven by current conditions and events) and also influenced by more enduring decisions such as policies and microplans. More generally, we have attempted to provide a model that explains how a competent tutor can interweave data-driven and tactical constraints with planful strategic ones, to arrive at intelligent tutorial decisions.

Implications for Intelligent Computer Tutors.

Although our findings have led to a tentative model of tutoring, it is still not specified precisely or completely enough to yield a computer-based tutor embodying the model. Nevertheless, we can draw several implications for the development of such systems. First, many intelligent tutoring systems appear to adhere to the diagnostic-remedial view, because most of their expertise is used for student modeling. Our results suggest that this approach might not yield the most effective automated tutors or the most human-like. However, this conclusion must be hedged. Although there is little evidence that human tutors perform extensive student diagnosis, we cannot conclude that it is impossible to produce high quality, computer-based tutors built around diagnosis. Just as computers play excellent chess using approaches unlike those used by humans, it is conceivable that computers could tutor well using approaches not exemplified by human teachers.

Although our results suggest that developers of intelligent tutoring systems might focus less on perfecting student diagnosis expertise, they also suggest that more research should be conducted to bolster the pedagogical expertise of such systems. Very little of the tutorial knowledge we have discussed can be found in current intelligent computer-based tutors. As Ohlsson (1986) pointed out, most of their expertise is at a tactical level. They exhibit little ability to develop strategic tutorial plans. Rather than calling them tutors, it might be more accurate to call most of them intelligent monitors of practice that use their knowledge of tactics and the student to generate detailed feedback for each local student error. Even at a tactical level, intelligent tutoring systems are limited. Their repertoire of tutoring techniques is small compared with those we have catalogued for human tutors. For example, their remedial responses are usually limited to a description of a bug that matches the student's response (if any) and a hint about the correct approach to the problem. The table in Appendix B shows a much wider range of human techniques for responding to errors, and even this list is not complete.

Currently, we are using the results of our analysis of human tutors to enrich the pedagogical component of our computer-based algebra tutor. Our plan is to begin modestly by adding to its repertoire of remedial techniques. Then we extend the repertoire to other phases of tutoring, including the task-management and performance assessments. Finally, we begin to impose a strategic organization on the deployment of techniques. A first attempt to impose strategy-level control on our algebra tutor is discussed in McArthur, et al. (1988).

Implications for Teacher Training

Finally, as we begin to understand what tutors do, and when and how they do it, it becomes natural to ask why. It may seem obvious why tutors provide hints or reasoning supports. But the reasons for choosing particular techniques under specific conditions are often difficult to understand. Interestingly, in our experience, asking the tutors why they choose certain techniques rarely helps. We speculate that they often have little knowledge of why they take certain actions. In this regard, their knowledge resembles that of many expert systems, which can frequently perform well using rules that associate specific conditions with specific actions. However, they embed no underlying theory of their subject matter that they could use to justify their actions (e.g., Clancey, 1983a).

If tutors cannot justify their choice of techniques, should we be concerned with such justifications? We believe so, for several reasons. First, we think many techniques do have deep justifications in terms of how they support the cognitive needs of learners. Even though tutors may not know such justifications, we believe the techniques they use have evolved and gone through a selection process. Generally, only those that are justified survive. Second, if we can discern justifications for tutoring techniques, we may take the first steps toward a normative science of tutoring. Ultimately, we would like to say not only what actions tutors do take, but also what actions they *should* take, under certain circumstances. To do so, we need some objective standards against which to measure a prospective technique. It is our belief that the compilation of the techniques that skilled tutors use, along with a rational reconstruction of justification for those techniques, holds the potential for defining objective tutoring standards.

ACKNOWLEDGMENTS

This research is funded by the National Science Foundation's Teaching and Learning Program (Grant MDR-8751104) and by The RAND Corporation. Views or conclusions expressed herein do not necessarily represent the policies or opinions of the sponsors.

We thank Lauren Resnick, Allan Collins, and Ralph Putnam for valuable comments on an early draft of this article.

REFERENCES

Anderson, J. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Anderson, J. R., Boyle, D. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science, 228*, 456–462.

Anderson, J., Boyle, D. F., & Yost, G. (1985). The geometry tutor. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 1*, 1–14.

Anderson, J. R., & Skwarecki, E. (1986). The automated tutoring of introductory computer programming. *Communications of the ACM, 29*(9), 842–849.

Ashlock, R. B. (1982). *Error patterns in computation: A semi-programmed approach* (3rd ed.). Columbus, OH: Merrill.

Bangert-Drowns, R. L., Kulik, J. A., & Kulik, C. C. (1985). Effectiveness of computer-based education in secondary schools. *Journal of Computer-Based Instruction, 12*(3), 59–68.

Berliner, D. (1986). In pursuit of the expert pedagogue. *Educational Researcher, 15*(7) 5–13.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of groups instruction as effective as one-to-one tutoring. *Educational Researcher, 13*(6), 4–16.

Brophy, J. E., & Good, T. L. (1986). Teacher behavior and student achievement. In M. Wittrock (Ed.), *Handbook of research on teaching* (pp. 328–375). New York: MacMillan.

Brown, J. S., & Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science, 2*, 155–192.

Brown, J. S., & Burton, R. R. (1987). Reactive learning environments for teaching electronic troubleshooting. *Advances in Man-Machine Systems Research, 3*, 65–98.

Brown, J. S., Burton, R. R., & DeKleer, J. (1982). Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II, and III. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 227–282). New York: Academic.

Bundy, A., & Welham, B. (1980). Using meta-level inference for selective application of multiple rewrite rule sets in algebraic manipulation. *Artificial Intelligence, 16*, 189–211.

Burton, R. R., & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 79–98). New York: Academic.

Clancy, W. J. (1979). Tutoring rules for guiding a case method dialogue. *International Journal of Man-Machine Studies, 11*, 25–49.

Clancey, W. J. (1983a). The epistemology of a rule-based expert system: A framework for explanation. *Artificial Intelligence, 20*, 215–251.

Clancey, W. J. (1983b). GUIDON. *Journal of Computer-Based Instruction, 10*(1 & 2), 8–15.

Clark, C. M., Snow R., & Shavelson R. J. (1976). Three experiments on learning to teach. *Journal of Teacher Education, 20*, 174–180.

Cohen, P. A., Kulik, J., & Kulik, C. C. (1982). Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal, 19*, 237–248.

Colker, L. (1982). *Teachers' interactive thoughts about pupil cognition*. Unpublished doctoral dissertation, University of Illinois at Champaign-Urbana.

Collins, A. (1988). Different goals of inquiry teaching. *Questioning Exchange, 2*(1), 39–45.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453–494). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Collins, A., & Stevens, A. L. (1982). Goals and strategies of inquiry teachers. In R. Glaser (Ed.), *Advances in instructional psychology* (Vol. 2, pp. 65–120). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Collins, A., & Stevens, A. L. (1983). A cognitive theory of interactive teaching. In C. M. Reigeluth (Ed.), *Instructional design theories and models: An overview.* New York: Academic.

Connors, R. D. (1978). *An analysis of teacher thought processes, beliefs, and principles during instruction*. Unpublished doctoral dissertation, The University of Alberta, Canada.

Hayes-Roth, B. (1984). BB1: An architecture for blackboard systems that control, explain, and learn about their own behavior (Report No. 84-16). Palo Alto, CA: Stanford University, Heuristic Programming Project.

Hayes-Roth, B., & Hayes-Roth, F. (1979). A cognitive model of planning. *Cognitive Science, 3*, 275–310.

Kulik, J. A., Kulik, C. C., & Bangert-Drowns, R. L. (1985). Effectiveness of computer-based education in elementary schools. *Computers in Human Behavior, 1*, 59–74.

Larkin, J., McDermott, D., Simon, D., & Simon, H. (1980). Expert and novice performance in solving physics problems. *Science, 208*, 1335–1342.

Langley, P., & Neches, R. (1981). *PRISM user's manual*. Technical report, Department of Psychology, Carnegie-Mellon University, Pittsburgh.

Lepper, M. R., & Chabay, R. W. (1985). Intrinsic motivation and instruction: Conflicting views on the role of motivational processes in computer-based education. *Educational Psychologist, 20*, 217–230.

Lepper, M. R., & Chabay, R. W. (1988). Socializing intelligent tutors: Bringing empathy to the computer tutor. In H. Mandl & A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems* (pp. 242-257). New York: Springer-Verlag.

Marland, P. W. (1977). *A study of teachers' interactive thoughts*. Unpublished doctoral dissertation, The University of Alberta, Canada.

Marx, R. W., & Peterson, P. L. (1981). The nature of teacher decision making. In B. R. Joyce, D. D. Brown, & L. Peck (Eds.), *Flexibility in teaching: An excursion into the nature of teaching and training* (pp. 236-255). New York: Longman.

McArthur, D., & Stasz, C. (1989). *An intelligent tutor for basic algebra* (Report No. R-3811-NSF). Santa Monica, CA: RAND.

McArthur, D., Stasz, C., & Hotta, J. (1987). Learning problem-solving skills in algebra. *Journal of Educational Technology Systems, 15*(3), 303-324.

McArthur, D., Stasz, C., Hotta, J., Peter, O., & Burdorf, C. (1988). Skill-oriented task sequencing in an intelligent tutor for basic algebra. *Instructional Science, 17*, 281-307.

McDermott, D. (1982). R1: A rule-based configurer of computer-systems. *Artificial Intelligence, 19*, 39-88.

McNair, K. (1978). Capturing inflight decisions. *Educational Research Quarterly, 3*(4), 26-42.

Ohlsson, S. (1986). Some principles of intelligent tutoring. *Instructional Science, 14*, 293-326.

Putnam, R. T. (1985). *Teachers' thoughts and actions in live and simulated tutoring of addition*. Unpublished doctoral dissertation, Stanford University.

Putnam, R. T. (1987). Structuring and adjusting content for students: A study of live and simulated tutoring of addition. *American Educational Research Journal, 24*, 13-48.

Semmel, D. S. (1977, April). *The effects of training on teacher decision making*. Paper presented at the annual meeting of the American Educational Research Association, New York. (ERIC Document Reproduction Service No. ED 138 538)

Shortliffe, E. (1976). *Computer-based medical consultants: MYCIN*. New York: American Elsevier.

Sleeman, D. (1982). Assessing aspects of competence in basic algebra. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 185-199). New York: Academic.

Sleeman, D. H., & Smith M. J. (1981). Modeling student's problem solving. *Artificial Intelligence, 16*, 171-188.

APPENDIX A

The following table interprets some of the symbols coding student responses in tutoring sessions.

Symbol	*Interpretation*
R +	Response to a previous request or prompt that is acceptable.
R −	Response to a previous request or prompt that is not acceptable.
Ro	Null response (hesitation) to a previous request or prompt.
R ˉ	Unclear response to a previous request or prompt.
R?	Response to a previous request or prompt is clear but incomplete or incomprehensible in the current context.
Rc	Response continues or elaborates on previous response to a request or prompt.
Re	Response echos a previous tutor utterance.
C +	Assenting response to previous yes-no query.
C −	Dissenting response to previous yes-no query.
Q	Request the tutor to confirm the acceptability of the student's previous response.
I	Request the tutor for more information about a previous prompt or request of the tutor.

APPENDIX B

Table of Tutoring Techniques

Technique Name	*Technique Description*	*Example*	*Purposes*	*Topics*	*n*[a]
Multipurpose					
Problem description	Tutor describes the problem. No new information is given. The description orients the student to the important features of the problem.	And you have *x*s on both sides and you have *b* on both sides.	Task management introduction and recapping, remedial	Problem conditions	5
Problem similarity	Tutor compares the current problem to a previously solved problem. Comparison is for similarities the problems share.	Again [as in the last problem] you have a problem where your *x*s are on both sides of the equation.	Task management introduction and recapping, remedial	Problem conditions, procedures	10
Problem difference	Tutor compares the current problem with a previously solved problem. Comparison is for differences between problems.	I'd like to do the next one, because you got this one correct, but there is a little different twist on 15.	Task management introduction and recapping, remedial.	Problem conditions, procedures	3
State high goal	Tutor provides the student the high-level goal for the reasoning step.	Now your objective is to get *x* by itself.	Task management introduction, remedial	Goal structures	11
State mid goal	Tutor provides the student the mid-level goal for the reasoning step.	I think you probably want to put all *x*s on one side and all *b*s on the other.	Task management introduction, remedial	Goal structures	11
State low goal	Tutor provides the student the low-level goal for the reasoning step.	You'll notice the 4 here. Do you know how to eliminate the 4?	Task management introduction, remedial	Goal structures	14
Model reasoning	Tutor models the "ideal" reasoning to the extent that all the student has to do is to provide an answer.	If you have 4 of something and I have 8 of something and we have exactly the same amount, the only thing we can have is absolutely . . .	Task management introduction, remedial	Goal structures, problem conditions, procedures	10

(Continued)

APPENDIX B *(Continued)*

Give answer	Tutor gives the student the answer or completes the reasoning step herself.	And that would only happen if you didn't have anything and I didn't have anything either. It would only happen if you put a zero in here.	Task management introduction, remedial	Goal structures, problem conditions, procedures, concepts	13
Compare methods	Tutor compares alternative approaches to solving a problem giving detailed explanation (modeling) the correct reasoning of each approach.	What you could have done is to multiply both sides by negative 1. It doesn't have to be written down but $(-1)(-x) = (-1)(21)$ that's what you could have done. Because if you did that, the -1 and the $-x$ gives you a $+x$ and the -1 times the 21 gives you -21.	Task management recapping, remedial	Goal structures, problem conditions, procedures, concepts	3
Task management					
Do in head	Tutor constrains the method of problem solving so that the student must do the work in his/her head.	On this particular one, if you just look at it without doing the math . . .	Task management introduction	Procedures	3
Alternate representation	Tutor constrains the method of problem solving so that the student must either rework the problem using an alternate method or represent the solution to the problem in a different manner.	Show me a number line . . .	Task management introduction	Procedures, concepts	4
Show work	Tutor constrains the method of problem solving so that the student has to write down his or her "work."	Now when I teach this I like to tell the youngsters to put that number up there.	Task management introduction	Goal structures, procedures, concepts	2
Prompt	Tutor asks what should be next. Prompts student to do or complete a reasoning step.	Where do you want to begin?	Task management introduction	n/a	34
Prompt how	The tutor asks the student to give more detail about his or her reasoning step.	How do you do that?	Task management introduction	n/a	6

Transition prompt	With this technique, the tutor indicates a shift in topic.	Let's get off from the zeroes now. Okay. How about some inequalities?	Task management introduction	n/a	1
Warning prompt	A prompt warns the student to be careful on the next step.	Now don't get caught on that one.	Task management introduction	n/a	1
Continue prompt	Tutor signals student to continue.	Umm . . . All right. Let's start and see . . .	Task management recapping	n/a	2
Problem point	The "lesson to be learned" is restated by the tutor. Tutor summarizes, justifies, explains, emphasizes how or why the correct reasoning step is done.	That's right. So once you eliminate fractions in any problem, you see how smoothly things begin to go after that.	Task management recapping	Goal structures, problem conditions, procedures, concepts	20
Cleanup	Part of recapping where tutor gives the student the task of putting the "finishing touches" to the reasoning step being completed.	Why don't you write your mx first. That might help you.	Task management recapping	n/a	3
Confidence assess query	Tutor questions how "comfortable" or "good" a student feels about his or her performance.	Do you feel good on this?	Task management recapping, knowledge assessment	n/a	11
Performance assessment					
Correct	Technique tutors use to let the student know how he or she has correctly completed a reasoning step.	Right.	Performance assessment	n/a	122
Grain of truth	Warns students that their decisions may be inappropriate, if not invalid.	Well, okay, if you want to use division, that's fine.	Performance assessment	n/a	8
Deny	Technique used to let the student know he or she has incorrectly completed a reasoning step.	Don't think so.	Performance assessment	n/a	4
Do more assess	Technique used to relate assessment that student needs more practice at a skill to "master" that skill.	I think if you do a lot of these I think you just get real relaxed with it.	Performance assessment	n/a	1

(Continued)

APPENDIX B *(Continued)*

Knowledge assessment					
Query	Tutor asks the student if he or she has the knowledge to complete a task.	Can you do that?	Knowledge assessment	n/a	6
Clarify	Tutor requests the student to further explain or elaborate his or her reasoning.	By dividing. What do you mean by dividing?	Knowledge assessment	n/a	1
Justify	Tutor requests the student to give reasons "justify" his or her reasoning step or answer.	How did you know to use division there?	Knowledge assessment	n/a	3
Remedial					
Simple techniques					
Continue	Tutor allows the student to continue with his or her reasoning.	Go ahead, divide both sides by *m*.	Remedial	n/a	1
Focus error	Tutor draws attention to the "location" of the error.	Not *ac*.	Remedial	Problem conditions, procedures, concepts	3
Try again	Tutor tells student that his or her answer is wrong and to try again.	Better look at it one more time.	Remedial	Goal structures, procedures	6
Suggest right procedure	Tutor suggests to the student the procedure that should be used to complete the reasoning step.	How about factoring, do you know how to factor?	Remedial	Procedures	3
Explain but	Tutor provides student with explanation or source of his or her error.	See, these two are separate terms, right?	Remedial	Goal structures, problem conditions, procedures, concepts	1
Change original problem	The tutor changes the problem that the student is solving so that the incorrect answer becomes a correct answer.	The way you filled that in, I'm going to have to change the original problem.	Remedial	Procedures	1
Consequences incorrect reasoning	The tutor carries out the student's incorrect reasoning to demonstrate how it would not lead to a logical or correct solution.	If you put b/x it would turn into a 1; I want x to stand alone . . .	Remedial	Goal structures, procedures, concepts	3

Complex techniques[b]		Script			
Map easier problem	Tutor interpolates an easier problem with the feature or skill that the student finds difficult. First, the easier problem is solved; then the tutor assists the student in applying the similar features or skills to solve the current problem.	Task (do-easier-problem), task (map-problem)	Remedial	Goal structures, problem conditions, procedures, concepts	3
Redo	Tutor has the student redo the step or problem they have just completed.	Task (redo)	Remedial	Goal structures, problem conditions, procedures, concepts	1
Model and check	Tutor models the correct reasoning and then questions the student to check for understanding.	Task (model), knowledge assess	Remedial	Goal structures, problem conditions, procedures, concepts	2
Drill	Tutor interpolates several easier problems with the skills that the student finds difficult. These problems are presented in succession, a "drill" of the skill, concept, and so on.	Task (do-easier-problem), repeat as necessary	Remedial	Goal structures, problem conditions, procedures, concepts	1
Map difference	Tutor refers the student to a previously solved problem. A comparison of differences is made, and using these differences, the tutor and student arrive at the correct reasoning step.	Task (find-different-problem), task (find-difference), task (step given feature)	Remedial	Goal structures, problem conditions, procedures	2
Map similarity	Same as map difference; however, in this case, a comparison of similarities leads to the correct reasoning step.	Task (find-similar-problem), task (find-similar-operation), task (map-operation)	Remedial	Goal structures, problem conditions, procedures	2

(Continued)

APPENDIX B *(Continued)*

Goal given feature	Tutor draws student's attention to the salient features of the problem. The tutor and student then arrive at the goal that these features suggest.	Task (find-variable), task (find-feature), task (step-given-feature)	Remedial	Goal structures, problem conditions, procedures	1
Apply concept	The tutor and student discuss the relevant concept that must be understood for the reasoning step to be completed. The concept is then applied to complete the reasoning step.	Knowledge assess, task (local-concept-use), task (map-concept)	Remedial	Goal structures, procedures, concepts	4
Local clarification					
Reply	Tutor replies positively to the student's question.	That's a negative *p*, yea.	Local clarification	n/a	4
Motivation					
Able	Tutor expresses her confidence in student's ability to complete a task.	I think you can do that real well.	Motivation	n/a	4

[a]Names of pedagogical tutoring techniques, along with brief descriptions, examples, purposes for which techniques can be used, topics that techniques cover, and a gross number of times we observed techniques being used. [b]Examples for complex remedial techniques describe the script of activities that make up the techniques. Specific examples are given in the text.

RAND/N-3231-NSF