Practice Enables Successful Learning Under Minimal Guidance

Angela Brunstein, Shawn Betts, and John R. Anderson Carnegie Mellon University

Two experiments were conducted, contrasting a minimally guided discovery condition with a variety of instructional conditions. College students interacted with a computer-based tutor that presented algebralike problems in a novel graphical representation. Although the tutor provided no instruction in a discovery condition, it constrained the possible actions sufficiently that students could always discover the algebraic transformations they needed to learn. In Experiment 1, with ample practice for each new transformation, students performed better in the discovery condition than any instructional condition. In Experiment 2, with only a little practice for each transformation, students performed worst in the discovery condition. The authors suggest that the high levels of practice in the 1st experiment made students more efficient at discovering the algebraic transformations. When the cognitive demands were manageable, the discovery students may have more often encoded the algebraic transformations in mathematically correct ways.

Keywords: discovery learning, intelligent tutors, practice, cognitive load

There is a long history of advocacy of discovery learning that includes such intellectual giants as Rousseau, Dewey, and Piaget. Bruner (1961) is frequently credited as the source for the modern research on discovery learning in the last 50 years. Discovery learning is typically contrasted with direct instruction, and the contrast between the two is best conceived of as a continuum. At one end of the continuum, students are directly told what they are to learn; at the other end, students are left to find out what they are to learn through exploration. However, no learning experience is pure; students given direct instruction often find themselves struggling to discover what the teacher means, and all discovery situations involve some minimal amount of guidance, if only to tell the students to try to make sense of the situation. Moreover, the space of instructional strategies is hardly one-dimensional, and strategies that tend to the discovery end can vary substantially. Kirschner, Sweller, and Clark (2006) have used the term minimally guided instruction to refer to strategies that tend to this end of the spectrum.

Correspondence concerning this article should be addressed to John R. Anderson, Carnegie Mellon University, Pittsburgh, PA 15213. E-mail: ja@cmu.edu Although minimally guided instruction continues to have its advocates (e.g., Fuson et al., 1997; Hiebert et al., 1996; Kamii & Dominick, 1998; von Glasersfeld, 1995), evidence and argument have been accumulating against it (e.g., Kirschner et al., 2006; Klahr & Nigam, 2004; Mayer, 2004; Rittle-Johnson, 2006). Indeed, in two of the responses to the Kirschner et al. (2006) criticisms of minimally guided learning, the authors (Hmelo-Silver, Duncan, & Chinn, 2007; Schmidt, Loyens, van Gog, & Paas, 2007) did not question the claim that minimally guided learning was bad. Rather, they questioned whether Kirschner et al. had it right in classifying problem-based and inquiry learning as minimally guided.

A distinction that one frequently finds in the literature (e.g., Baroody, Lai, & Mix, 2006; Mayer, 2004; Shulman & Keisler, 1966) is between *pure discovery* and *guided discovery*. In typical guided discovery, the teacher provides "hints, direction, coaching, feedback and/or modeling" to keep the student on track, whereas in pure discovery the teacher provides "little or no guidance" (Mayer, 2004, p. 15). Mayer argued that pure discovery is almost always worse than direct instruction because students often fail to come in contact with the material to be learned. On the other hand, he argued that guided discovery can be more successful than direct instruction because it leads to integration of the new information with existing information.

It is ambiguous whether what we call a discovery condition in this article should be called guided discovery. In this study, students interacted with a computer-based tutoring system. In the discovery condition, the tutor does not provide any hints, direction, coaching, or modeling, as described by Mayer (2004). However, the nature of the computer interface means that it does provide feedback, sometimes more immediate and sometimes more delayed, on whether students have performed correct or incorrect actions. This feedback can be viewed as providing "hotter" or "colder" evaluations of student actions at some delay from these actions. The computer interface also limits the search space that the students have to explore in trying to make their discoveries.

Angela Brunstein, Shawn Betts, and John R. Anderson, Psychology Department, Carnegie Mellon University.

Angela Brunstein is now at the Department of Social and Decision Sciences, Carnegie Mellon University.

This research was supported by National Science Foundation Award REC-0087396 and Grant AFOSR–FA9550-07-1-0359 from the Defense Advanced Research Projects Agency. Angela Brunstein was partly supported by a Feodor-Lynen Research Prize from the Alexander von Humboldt Foundation.

An extended version of this article reporting an ACT-R model and its fit to the data is available from the ACT-R website, http://act-r.psy.cmu.edu/ publications/index.php (Brunstein, Betts, & Anderson [2009]. *When minimal guidance does and does not work: Drill and kill makes discovery learning a success*. Unpublished manuscript)

Perhaps it would be most accurate to characterize our discovery¹ condition as minimally guided discovery, in line with the usage of Kirschner et al. (2006).

This research is part of an effort to understand the contribution of instructional content in cognitive tutors that are based on cognitive models of how students solve mathematics problems. Cognitive tutors have been shown to have some success in the teaching of mathematics (Anderson, Corbett, Koedinger, & Pelletier, 1995; Koedinger, Anderson, Hadley, & Mark, 1997). They are deployed in over 2,600 schools throughout the United States and interact with approximately 500,000 students each year (Koedinger & Corbett, 2006; Ritter, Anderson, Koedinger, & Corbett, 2007; Ritter, Haverty, Koedinger, Hadley, & Corbett, 2008). It has been noted that students working with the tutors can become too dependent on the help they receive, and there has been extensive research on how to make these tutors more effective (e.g., Aleven, McLaren, Roll, & Koedinger, 2006; Baker et al., 2008; Heffernan, Koedinger, & Razzaq, 2008). The current research investigates whether one can get better learning by removing some of the guidance that the tutor provides on how to solve problems. More specifically, this research looks at how degree of practice can influence the effectiveness of a minimally guided discovery condition. In this research, the instructional conditions we use mainly serve as reference points for understanding the effect of such manipulations on the discovery condition.

The mathematics topics taught by the tutors have a sufficiently rich combinatorial structure that it is not possible to provide students with direct instruction on all possible cases. Students must generalize what they learn on specific cases to new cases. For instance, in this research, after students learned to rewrite (4 + x) + 3 as 7 + x, they were given the new problem (5 + x) - 3. Although the majority of students correctly generalized and rewrote this as 2 + x, a significant minority displayed the error of rewriting it as 2 - x. Making the correct generalization to this case can be viewed as minidiscovery informed by knowledge of the constraints of algebra and arithmetic. Thus, even though students are taught to rewrite (4 + x) + 3 as 7 + x, and even though they know enough about subtraction and addition to extend that knowledge, they must still determine how to integrate that knowledge in the case of (5 + x) - 3. This is basically a minidiscovery. Students are better able to make such generalizations if they integrate what they are learning with their general knowledge of arithmetic. Mayer (2004) argued that this is more likely to happen with guided discovery, and we show that this can happen in the discovery environment we have created.

Figure 1 shows some screen images involving equation solving in the Carnegie Learning Cognitive Tutor (2007). In terms of the interface interactions, these are the simplest parts of the algebra curriculum, but they reflect the general model of interaction with the Cognitive Tutor. In Figure 1a, the student is presented with the equation 8y + (-6y) + 9 = 10, and the student selects an operation to perform from a pull-down menu—in this case, the student has erroneously selected *Distribute*. The student then will receive feedback and eventually choose the correct operation of *Add/Subtract Terms*. When this correct operation is chosen, the tutor presents a display like Figure 1b, where the student must indicate the result of adding like terms by filling in a series of boxes. The resulting equation is represented in Figure 1c, and the student must choose a correct operation again. Upon doing so, the tutor once again presents a series of boxes (Figure 1d) where the student must indicate the terms being subtracted. This illustrates the basic cycle in the tutor in which the student selects some operation to perform (Figures 1a and 1c) and then executes the result of that operation (Figures 1b and 1d) by filling in some boxes. By isolating the individual operations and executions, the tutor is able to identify specific difficulties that the student is having and to provide instruction on those aspects.

The research reported here involves some extreme experimental manipulations that might well fail to result in mastery of algebra. Therefore, we did not want to study children learning linear algebra, lest our experimental manipulation hinder their future ability to master algebra. Rather, we developed a data-flow isomorph of linear algebra equations suitable for teaching to college students who have already mastered linear algebra. Essentially, because of the novel format, college students go through the process of learning to solve equations anew. If students fail to learn, as they did in some of the conditions reported here, it is at no loss to their competence with normal algebra. Nevertheless, the semantics that underlie operations in the data-flow isomorph are the same as in standard algebra. Therefore, learning to interact with the tutor corresponds with grasping the semantics of algebra to exploit its combinatorial structure. Figure 2 shows two examples of data-flow graphs that correspond to particular linear equations. Figure 2a is the isomorph of the equation 5x + 4 = 39, and Figure 2b is the isomorph of the equation (2x - 5x) + 13 + 9x =67. In such a diagram, a number comes in the top box and flows through a set of arithmetic operations; the result is the number that appears in the bottom box. Students are taught a set of graph transformations isomorphic to the transformations on the linear equations that result in simplifying the diagram. In the case of problems like those in Figure 2, these simplifications result in determining the value that has to go into the top box to produce the value in the bottom box. This is the equivalent to solving for a variable (i.e., x). However, some diagrams are the equivalent of expressions to be simplified (not equations to be solved), and their simplification requires the equivalent of algebra's collection of like terms and distribution of multiplication over addition. Anderson (2007) reported a behavioral comparison of children working with linear equations and adults working with the data-flow tutor. Although children were a bit more error prone, they learned and behaved very similarly.

We used the tutoring system described in Anderson (2007), which has the same interaction style as the Cognitive Tutor (2007) for algebra. It involved selecting parts of the diagram with a mouse, selecting transformations of those diagrams, and typing in the contents of the changed portions of the diagram. As with the Cognitive Tutor, this experimental tutor provides some initial guiding instruction, with further instruction if the student requests help or makes errors. Almost all of this instruction and guidance was removed in the discovery condition. We specify more about the interaction style and the various instructional conditions below in the description of Experiment 1.

¹ *Discovery* is intended to denote the name for a condition rather than an assertion about the true nature of that condition.



Figure 1. A representation of four steps in the solution of an equation with the algebra tutor: (a) selection of a transformation; (b) filling in of the transformation; (c) selection of an evaluation; (d) filling in of the evaluation.

Experiment 1

Much of the instruction provided by cognitive tutors comes in response to help requests or errors. Typical of most instruction in mathematics, this instruction involves a mix of verbal directions and pieces of worked examples. The first experiment reported here was an attempt to assess separately the contributions of the instruction and worked example and whether we could get better learning if we removed one or both. There was a *verbal direction* condition in which participants received abstract verbal instruction without any specific directions about how to solve a problem and a *direct demonstration* condition in which participants were told what to do in a specific case without receiving any general characterization of the action. To complete a factorial design, we crossed the use of verbal direction with direct demonstration. This created a *both* condition that was similar to the original condition



Figure 2. The data flow equivalents of (a) 5x + 4 = 39 and (b) (2x - 5x) + 13 + 9x = 67.

of Anderson (2007), where students received both an abstract characterization and a demonstration of what to do. This also created the discovery condition, where there was no instruction accompanying the steps. Many experiments have compared examples, instructions, and a combination of the two (e.g., Charney, Reder, & Kusbit, 1990; Cheng, Holyoak, Nisbett, & Oliver, 1986; Fong, Krantz, & Nisbett, 1986; Reed & Bolstad, 1991), but these experiments have tended not to look at situations in which the participants receive no direction, as in our discovery condition. These experiments have produced somewhat different estimates of the relative contributions of examples and instruction, presumably reflecting differences in the material.

Figure 3 illustrates the simple interface of the tutor. There are three interactions that students can have with the tutor. They can select boxes in the data-flow graph to operate on, select operations from the buttons to the right, and type values of the expressions into dialog boxes like the one illustrated in Figure 3. The correct combination of these actions can succeed in simplifying the diagram. This is much like the Carnegie Learning algebra tutor (Cognitive Tutor, 2007) in Figure 1.

Figure 4 uses a problem concerned with collection of like terms to illustrate the basic cycle that occurs throughout the curriculum. The problem in Figure 4 is the data-flow equivalent of 3 + (2x + 7). The first row in Figure 4 shows steps in the transformation of the problem from its original form to the equivalent of (7 + 3) + 2x; the second row shows steps in transforming this to the equivalent of 10 + 2x. As the curriculum progresses, the problems become more complex and require more varied transformations, but their solutions always have the interaction cycle illustrated in Figure 4, as follows:



Figure 3. The tutor interface at the point where a result is to be typed.

- The diagram begins in some neutral display (panels a and d), and the student selects some boxes to operate on. Later problems can require selection of as many as five boxes, and there can be a number of alternative correct choices about which sets of boxes to operate on next.
- 2. The selected boxes are highlighted in red (panels b and e), and the student selects some operation by clicking a button to the right of the diagram.
- 3. If a correct set of boxes and operations have been chosen, the diagram is transformed with a number of green boxes (panels c and f) to be filled in. The student can click these

green boxes; an input dialog appears (see Figure 3), and the student can type information into the input dialog.

4. When the boxes are filled in with syntactically correct expressions (not necessarily the correct values), the diagram returns to a neutral state (panels d and g), ready for the next selection of some set of boxes.

The tutor's color conventions, as illustrated in Figure 4, are that red boxes indicate parts of the diagram selected for an operation and green boxes indicate information to be filled in. When the transformations are complete, the student clicks the *Next Problem* button. If the transformations have been correctly performed, the student can go onto the next problem. If there was an error, the student is informed that he or she can not go on to the next problem but has to correct the error. The *First Mistake* button takes the student to the state of the diagram before the first mistake. The arrow buttons allow students to move backward or forward by a single transformation.

The material used in this experiment comes from 12 sections over four chapters in *Algebra I* (Foerster, 1990), an algebra textbook that covers what is needed to solve linear equations. The first one or two problems in each section were used for instructing the material in that section. For these problems, instruction was volunteered whereas instruction was available on request for later problems (except for the discovery condition, where there was never any instruction).

The problem in Figure 4 was used for initial instruction in section 2.6 on combining constants. Table 1 shows the instruction that accompanied this problem. There is some general initial instruction and then instruction that accompanies each state of the problem. The instructional manipulations involved the state-by-



Figure 4. The steps in the solution of a combine problem, the data flow equivalent of (2x + 7) + 3, from section 2.6. Each picture is a different state of the diagram on its way to its simplest form. The two lines reflect the two transformations. The first line starts with the problem (a), then a part of the graph is selected and highlighted in red (b), then the combine operation is selected (resulting in c), and the parts are filled in (resulting in d). The second line starts with d from the previous line; then a part of the graph is selected for evaluation (e); the evaluation operation is selected (resulting in f); and the value is filled in (resulting in g).

 Table 1

 Instructions for Section 2.6 on Collecting Constant Terms

	One can collapse two boxes with $+$ or $-$ into a single
Initial general	box and preserve the value of the diagram. One can
instructions	do the same thing with two boxes with * or /.

	Verbal directions	Direct demonstration
State		
а	Find two boxes with addition or subtraction and click them	Click This arrow
b	Click the button labeled Combine.	Click This arrow
с	Click the little green box.	Click This arrow
	Enter the operator from the box above.	Type + arrow
	Click the green big box.	Click This arrow
	Enter the number from the box above, then the operator from the box below, and then the number from the box below.	<i>Type 7</i> + <i>3</i> arrow
d	Find a box with two numbers and an operator and click it.	Click This arrow
e	Click the button labeled Evaluate.	Click This arrow
f	Click the little green box.	Click This arrow
	Find the answer by evaluating the box above and enter it.	<i>Type 10</i> arrow
g	Your answer is correct. Type the Next Problem button.	Click This arrow

state instruction. In the verbal direction condition, participants would receive instructions such as *Find two boxes with addition or subtraction and click them*, which provided guidance on how to perform the operation on this and similar problems without saying exactly what to do. In the direct demonstration condition, participants were told what to do in this specific case without stating any general characterization of the action. For instance, arrows would point to the two boxes with the instruction *Click this*. In the *both* condition, participants saw both forms of instruction, whereas in the discovery condition they saw neither. For section 2.6 on combining constants, the most critical transformation is between states like Figures 4c and 4d, where the participant must specify the content of the boxes in a way that preserves the value of the graph structure.

Participants in the discovery condition received none of the guidance illustrated for states (a) through (g) in Table 1, although they did see the initial general instructions. They had to try various actions and learn from the consequences of their actions. The following is a list of the sorts of errors that could be made and the feedback that would occur—this feedback was also available in the other conditions. The cases are listed in order of increasing delay of feedback.

1. Interface errors: If the student tried some action that the interface was not prepared to process (such as typing a number when there is no dialog box, as in Figure 3), the tutor did not change state. This lack of response was an immediate indication that the action had been rejected.

2. Operator errors: After students selected some boxes and an operator, they either saw the screen transform into a state with green boxes to be completed, indicating success, or saw an error message saying the operator would not apply to the boxes selected. The error message provided no explanation of why. It just provided the student with information that there was something

wrong—either with the boxes they had selected or with the operator they had just selected.

3. Transformation errors: After selecting the operation, the student specified the transformation by typing material into the green boxes. Syntactically incorrect material was not accepted, but wellformed incorrect answers were accepted. Feedback on such semantic errors only occurred when the students got to the end of the problem and chose the Next Problem operator. If all the transformations had been correctly entered, the tutor went on to the next problem. Otherwise, it would give an error message: Your answer is incorrect. Use the buttons (or the left and right arrow keys) and the First Mistake button to review your work and correct the mistake. Delaying feedback on these errors to the end of the problem is analogous to what happens in algebra texts, where a student performs a series of transformations and can only check the final result against the answer in the back of the book. One exception to delay of feedback on transformation errors was on the first one or two instructional problems: If an incorrect result was typed into a green box, it was rejected with the error message Your answer is incorrect. The other possible exception is that students could choose to hit the First Mistake button to find out if they had made any transformation errors so far in their solution, but they seldom used this option.

These features had been built into the interface prior to designing our discovery condition. They were simply what remained after we removed the direct instruction to create the discovery condition. It might seem surprising that students could always discover how to solve the problems, but the interface limited the options enough that all students in the discovery condition eventually found solutions. Thus, the discovery students could be viewed as searching through a maze of interface actions, with the interface being responsive and restrictive enough that they eventually found their way out of the maze (i.e., solved the problem). The research tells us whether they actually learned anything about the domain from their search through this maze of actions.

Method

Materials

Participants solved 174 data-flow problems based on problems from 12 sections in the Foerster (1990) text that spanned the first four chapters. The first session took on average about 1 hr, whereas the second and third sessions took on average approximately 1.5 hr. Solving these problems required performing at least 674 operations. Below are the 12 sections and examples of the problems in their linear algebra equivalent form (the sections are labeled with chapter number first, followed by the section number within the chapter).

Section 1.1: Evaluating diagrams (14 problems). Teaches students how to evaluate the contents of boxes in the data-flow diagrams—for example, rewrite $(9 - 4) \times 2$ as 5×2 , and rewrite this as 10.

Section 1.2: Input boxes (nine problems). Teaches students to evaluate a diagram given a value for an input box—for example, rewrite (24/x) - 1 and x = 12 as 24/12 - 1, and this as 2 - 1, and this as 1.

Section 1.7: Finding input values (25 problems). Teaches students to find the input values given single operations—for example, rewrite x + 3 = 8 as x = 8 - 3, and this as x = 5.

Section 2.6: Combining operations (20 problems). Teaches students how to combine constant terms—for example, rewrite (5 + x) - 3 as (5 - 3) + x, and this as 2 + x.

Section 2.7: More on finding input values (16 problems). Teaches students to find the input values given two operations—for example, 2x + 3 = 19—and to deal with asymmetric operators—for example, rewrite 10 - x = 2 as x = 10 - 2, and this as x = 8.

Section 3.1: Reordering operations (six problems). Teaches students the graph equivalent of distribution—for example, rewrite $5 \times (x + 2) + 9$ as $[5x + (5 \times 2)] + 9$, and this as (5x + 10) + 9, and this as (10 + 9) + 5x, and finally as 19 + 5x.

Section 3.2: Reordering and subtraction (nine problems). Teaches students to use reordering with subtraction in problems such as $9 - 2 \times (x - 4)$.

Section 3.4: Combining multiple input boxes (13 problems). Teaches students the equivalent of collecting variable terms—for example, rewrite 7x + 5x as $(7 + 5) \times x$, then as 12x, and rewrite 5x + (6 - 2x) as $6 + (5 - 2) \times x$, then as 6 + 3x.

Section 3.5: More on combining input boxes (12 problems). Deals with special cases like 2x + x (no coefficient before the variable) and (6x + 3) - (6 - 2x) (combining both variables and constants).

Section 4.1: Finding input values in more complex problems (11 problems). Puts the operations together, building up to equations like [(3x + 4) + 5x] + 6x = 32.

Section 4.2: Finding input values in harder problems (21 problems). Builds up to equations like $3 \times (2x - 1) + 2 \times (x + 5) = 55$.

Section 4.3: Finding input values when two data-flow diagrams are equal (18 problems). Presents equations like 3x + 55 = 8x.

The other sections of the Foerster (1990) textbook did not involve material relevant to linear equations. For instance, sections 2.1 to 2.5 were a review of signed arithmetic.

Participants and Conditions

Forty Carnegie Mellon undergraduates (23 male and 17 female; M = 23 years, SD = 1.6 years) took part in this study. They reported relatively high grades in their high school algebra courses (24 As, 8 Bs, 4 Cs, 4 missing). Students participated in three single sessions, each lasting between 1 and 2.5 hr. In the first session, they went through the sections above from chapters 1 and 2; in the second session, they went through chapter 3; and in the third session, they went through chapter 4. They received performance-based feedback in the form of \$ 0.07 per correctly performed operation in the tutor or a guaranteed minimum of \$5 per half hour, whichever was greater. Fourteen students received performance-based pay in Session 1, none in Session 2, and 23 in Session 3.

Ten participants were randomly assigned to each of four conditions. The four conditions were defined by different combinations of instructions such as those in Table 1. The verbal direction condition received only the verbal directions; the direct demonstration condition received only the direct demonstration; the both condition received both; and the discovery condition received neither.

Measurements

The first problem in each section involved different combinations of the guided instruction in Table 1 (including none for the discovery condition). For sections 2.7, 3.4, and 3.5, the second problem in a section also involved guided instruction. Even in sections without guided instruction on the second problem, participants often floundered on the second problem and requested instruction. For these reasons, we treat the first two problems as the instructional problems and the remainder as the practice problems. We measured time to solve the whole problem, number of operations, time to perform single operators, number of operator errors, and number of transformation errors.

Results

Figure 5 shows the mean total time (time from initial presentation of the problem to successful clicking of Next Problem button to complete the current problem) to solve problems in the four conditions for the four chapters. The data are partitioned into performance on the first two instructional problems and performance on the remaining practice problems in each section. There are large differences in the times to solve problems for different chapters, reflecting the different number of transformations required to solve a problem in that chapter. We ignore the factor of chapter in our statistical analyses and simply use graphs like Figure 5 to show that the basic effects replicate over chapters. Therefore, our statistical analyses are 4×2 analyses of variance with the factors being the four instructional conditions and position in section (first two problems vs. later problems). In the case of total time, there are no significant effects of instructional condition, F(3, 36) = 1.29, p > .25, MSE = 2,598, or position, F(1, 36) = 1.29, p > .25, MSE = 2,598, or position, F(1, 36) = 1.29, p > .25, MSE = 2,598, or position, F(1, 36) = 1.29, p > .25, MSE = 2,598, or position, F(1, 36) = 1.29, p > .25, MSE = 2,598, or position, F(1, 36) = 1.29, p > .25, MSE = 2,598, or position, F(1, 36) = 1.29, P > .25, MSE = 2,598, or position, F(1, 36) = 1.29, P > .25, MSE = 2,598, or position, F(1, 36) = 1.29, P > .25, MSE = 2,598, P > .25, P > .25, MSE = 2,598, P > .25, P > .2536) = 0.30, MSE = 554, but there is a very strong interaction between the two, F(3, 36) = 17.99, p < .0001, MSE = 553. As is apparent from Figure 5, this interaction is driven by the fact that the discovery condition is worst on the initial two problems but best on the remaining problems. A contrast for this effect is highly significant, F(1, 36) = 53.17, p < .0001, whereas the residual effects in the interaction are not significant, F(2, 36) = 0.40. It is



Figure 5. Time to solve problems as a function of instructional condition, chapter, and whether the problems were the first instructional problems in a section or later practice problems (Experiment 1).

not surprising that participants have difficulty on the initial couple of problems in the discovery condition. What is interesting is their superior performance on the remaining problems. These remaining problems assess what the student has learned in a section. Individual *t* tests confirm that the discovery condition is statistically superior to the both condition and the verbal direction conditions on the rest of the problems in the section, t(18) = 2.78, p < .05, Cohen's d = 1.31, and t(18) = 3.35, p < .005, d = 1.58, but the difference between direct demonstration and discovery does not reach significance, t(18) = 1.40, p < .20, d = 0.66.

The total time to solve problems can be decomposed into the number of transformations that participants perform and the time per transformation (the product of these two numbers yields the total time on a problem). These two measures are shown in Figure 6. Figure 6a shows the number of transformations and, for reference, the minimum number of transformations required for perfect performance. The main reason participants perform more than the minimum number of transformations is that they make errors in filling in the values for the transformations and have to



Figure 6. Mean time per transformation (a) and mean number of transformations (b) as a function of instructional condition, chapter, and whether the problems were the first instructional problems in a section or later practice problems (Experiment 1).

redo them when they discover this-usually when they try to submit their answer at the end. There are main effects of condition, F(3, 36) = 3.62, p < .05, MSE = 0.915; and position, F(1, 36) =674.70, p < .0001, MSE = 0.474; and a strong interaction between the two, F(3, 36) = 7.17, p < .001, MSE = 0.474. The effect of position just reflects the fact that later problems in a section tend to involve more transformations. The interaction reflects the fact that there is almost no effect of condition on the first two problems, although the conditions separate on later problems in a section. Participants did not do much more than the minimum number of transformations on the first two problems because many transformation errors are immediately flagged for these problems. On the remaining problems in the rest of the section, where transformation errors are not flagged, the discovery condition shows the fewest transformations. A contrast for this effect is highly significant, F(1,36) = 19.51, p < .0001, whereas the residual effects in the interaction are not significant, F(2, 36) = 1.00. Individual t tests confirm that the discovery condition is statistically superior to all conditions on the rest of the problems in the sections: both, t(18) =4.05, p < .001, d = 1.91; verbal direction, t(18) = 3.39, p < .005, d = 1.60; direct demonstration, t(18) = 3.74, p < .005, d = 1.76.

Figure 6b shows the time per transformation.² The effect of condition is not significant, F(3, 36) = 1.69, p > .10, MSE =170.15, whereas the effect of position is, F(1, 36) = 136.71, p < 100.0001, MSE = 55.42. The effect of position in the section reflects a speed-up with practice. There is again a strong interaction between the two factors, F(3, 36) = 9.62, p < .0001, MSE =55.42, and again this reflects the fact that the discovery condition is worst on initial problems but best for the rest of the problems in a section. Again, a contrast for this effect is highly significant, F(1,36) = 27.34, p < .0001, whereas the residual effects are not, F(2, p) = 1000036) = 0.76. However, this time the effect mainly comes from the slower performance of discovery students on the initial transformations, where they must find out how to perform the new transformations. This effect is particularly pronounced for the first two chapters, where most of the operations in the first problems are new. Individual t tests on the rest of the problems find no significant differences between the discovery condition and other conditions: both, t(18) = 1.68; verbal direction, t(18) = 1.72; direct demonstration, t(18) = 0.60; all ps > .10. Thus, discovery students are faster on the rest of the problems in Figure 5 because of their advantage in number of transformations (Figure 6a), not time per transformation (Figure 6b).

One can better understand the source of the difference among the conditions by considering separately the operator and transformation errors described in the introduction to this experiment. The first, the operator error, involves selecting a wrong operator for the boxes chosen (the state transitions from Figure 4b to Figure 4c and from Figure 4e to Figure 4f). This can reflect either that boxes were selected for which no operator applies or that the wrong operator was chosen for an appropriate set of boxes. These errors are flagged after the operator is chosen. The second type of error, the transformation error, involves entering the wrong values for

² There is a sharp drop-off in time per operation for the first two problems in chapter 4 because this chapter mainly involves putting together operations already taught to solve complex equations. Thus, with one exception, the operations in the first problems are not new.

these boxes. The tutor will accept these wrong values and transition to the next state (e.g., wrong versions of states in Figures 4d and 4g). Thus, in contrast to operator errors, transformation errors are not flagged, and students tend to go on making further operations that will eventually have to be undone. Operator errors just lengthen the duration of a transformation as the students try again for a different box–operator combination, and so they should impact the performance measure in Figure 6b. Transformation errors will increase the number of transformations in Figure 6a. Roughly stated, operator errors reflect not knowing what to do next, and transformation errors reflect not knowing how to do it. These two categories of errors are presented in Figure 7.

Figure 7a shows the mean number of operator errors per problem. The effect of condition is significant, F(3, 36) = 6.25, p < .005, MSE = 2.67, whereas the effect of position is not, F(1, 36) = 1.04, MSE = 1.98. There is again a strong interaction between the two, F(3, 36) = 20.42, p < .0001, MSE = 1.98, and this time it reflects how poorly the discovery participants were doing on the first problems where they had to discover box–operator combina-



Figure 7. Mean number of operator errors (a) and transformation errors (b) as a function of instructional condition, chapter, and whether the problems were the first instructional problems in a section or later practice problems (Experiment 1).

tions. The main effect of condition also reflects this effect on the first problems. Again, a contrast for this effect (discovery worse than the rest) on the first problems is highly significant, F(1, 36) = 61.02, p < .0001, whereas the residual effects are not, F(2, 36) = 0.12. Individual *t* tests on the rest of the problems find no significant differences between the discovery condition and other conditions: both, t(18) = -0.16; verbal direction, t(18) = 0.33; direct demonstration, t(18) = 1.18; all ps > .10.

Figure 7b shows the number of transformation errors. The effect of condition is nonsignificant, F(3, 36) = 2.54, p < .10, MSE =0.189, whereas the effect of position is quite significant, F(1, 1)36) = 38.16, p < .0001, MSE = 0.152, reflecting the strong guidance provided for initial problems. The interaction of these two factors is again significant, F(3, 36) = 4.05, p < .05, MSE =0.152. The interaction reflects the fact that there is almost no effect of condition in the first two problems, whereas the discovery condition is better on later problems where there is more opportunity for wrong transformations. Again, a contrast for this effect is significant, F(1, 36) = 6.88, p < .05, whereas the residual effects in the interaction are nonsignificant, F(2, 36) = 2.64, p <.10. Individual t tests confirm that the discovery condition is statistically superior to all conditions on the rest of the problems in the sections: both, t(18) = 2.88, p < .01, d = 1.36; verbal direction, t(18) = 2.48, p < .05, d = 1.17; direct demonstration, t(18) = 2.56, p < .05, d = 1.21.

In summary, after the first couple of learning problems the discovery condition enjoys an advantage over the other conditions on the remaining practice problems. Even if we add in the first two problems in each section, the discovery condition is at an advantage: It takes an average of 193 min to go through all 174 problems, whereas the average in the other conditions is 226 min—an advantage of over half an hour that is quite significant, t(38) = 3.40, p < .005, d = 1.10. Although the number of students in the conditions is not large, the effect size is very large. The advantage of the discovery condition can be traced to the fewer transformations that participants have to perform. This in turn can be traced to the fewer mistaken transformations that students make, leading to fewer repairs and less confusion.

Experiment 2

In the previous experiment, students in the discovery condition seemed to have completed their learning after the first two problems. The later problems in a section gave us evidence about what participants had learned but did not seem to be important to learning. Averaged across all sections, discovery students took 26.85 s per transformation on the first two problems, 10.89 s on the next two, and 11.87 s on the last two. Thus, there seems to be no speed-up after the first two problems in a section. The critical transformation errors were a low 2.1% per opportunity on the second two problems (it is hard to make transformation errors on the first two because of the interface) and 1.7% on the last problems. It appears that the students could have obtained the benefit of the discovery condition with far fewer problems. However, we suspected that the extra practice gave participants a familiarity with the overall system and the semantics of the diagrams that enabled them to learn so effectively in the discovery condition. To investigate this, we greatly reduced the number of problems in the second experiment, from 174 to 44. We kept the

same first two problems for each of the 12 sections but used only 20 of the remaining 152 for an average of about two extra problems per section. We tried to keep the number of extra problems approximately in proportion to the original frequency in the full set of 152. The remaining problems per section were one for section 1.1, one for section 1.2, four for section 1.7, three for section 2.6, two for section 2.7, none for section 3.1, one for section 3.2, one for section 3.4, two for section 3.5, one for section 4.1, two for section 4.2, and two for section 4.3.

The experiment was also performed to investigate a second issue about the first experiment. As Table 1 indicates, even though discovery participants did not receive any instruction about how to perform the transformations, they were given general instructions about the general purpose of the transformations—for instance, that the combine operator served to collapse boxes with two + or - operators or two \times or / operators. We wanted to determine the contribution of these general instructions to learning.

There were no dramatic differences between the three instruction conditions in the first experiment. Therefore, this experiment used just one of the conditions, the direct demonstration condition, to contrast with the discovery condition. Thus, the design of the experiment crossed whether participants were given direct demonstrations or not and whether there were global instructions or not.

Method

Participants

Forty Carnegie Mellon undergraduates (27 male and 13 female; M = 23 years, SD = 2.1 years) took part in this study. Although they received the same performance-based feedback in terms of a financial score, the low practice in this experiment meant that students did not earn performance-based pay greater than the guaranteed minimum of \$5 per half hour. Ten participants were randomly assigned to each of the four conditions produced by crossing the presence of global versus no global instructions with the factor of demonstration versus discovery. They reported relatively high algebra grades (20 As, 11 Bs, 2 Cs, 7 missing data). These participants came from the same undergraduate pool as the first experiment, and there is no significant difference in the distribution of prior grades, $\chi^2(3, N = 80) = 2.32$, p = .50. In both cases the grade point average of the reported grades is 3.55.

Procedure

Except for fewer problems and the removal of the general instructions for half of the participants, the tutor and procedures were the same in this experiment as the previous experiment.

Results and Discussion

Qualitatively, results for the discovery conditions in this experiment differed greatly from the previous experiment. Six participants quit in the discovery condition with global instructions and four participants quit in the discovery condition without global instructions. They reached a point where they felt totally lost and did not want to continue. No participants quit in the direct demonstration conditions of this experiment, and none had quit in any conditions of the previous experiment. In addition, 3 further participants did not have enough time to complete all the problems in the discovery condition with global instructions, and 2 did not have time to complete all the problems in the discovery conditions without global directions. Thus, 50% of the discovery students quit, and another 25% went so slowly that they could not complete the experiment. In the direct demonstration condition, only one participant (without global instructions) did not complete the problems in the allotted time. The difference in number of participants completing the experiment is quite significant between the discovery and direct demonstration conditions, $\chi^2(1, N =$ 40) = 19.06, p < .0001. Although there was a slightly greater tendency for greater participant loss in the discovery condition with global instructions, this was not significant, $\chi^2(1, N =$ 20) = 2.40, p = .12.

Figure 8 presents the time per problem for those participants who did offer observations to a chapter (number of participants contributing is noted on the figure). Even though the poorest performing participants were eliminated on later chapters, the discovery participants were significantly worse than the direct demonstration participants at the .05 level or greater, with only one exception (the difference on the remaining problems for chapter 1). None of the differences between the two direct demonstration conditions were significant, and only one of the differences between the two discovery conditions was significant; in the rest of chapter 3, global instructions were worse than no global instructions, t(13) = 2.23, p < .05.

Interpreting the results for chapters 3 and 4 is problematical for another reason besides the loss of over half the participants in the discovery condition. Participants in the direct demonstration condition asked for a great many hints as they solved the rest-of-thesection problems in these chapters. In chapters 1 and 2, they averaged 0.04 hint requests per problem, whereas they averaged 3.76 for chapters 3 and 4. For comparison, instructed participants in Experiment 1 averaged 0.02 requests on the same problems for chapters 1 and 2, whereas they averaged 1.25 for chapters 3 and 4. The difference between experiments is not significant for chapters 1 and 2, t(48) = 1.41, d = 0.40, whereas it is highly significant for



Figure 8. Time to solve problems as a function of instructional condition, chapter, and whether the problems were the first instructional problems in a section or later practice problems. The number of participants out of the original number contributing to the last two chapters is given above the data point for those chapters (Experiment 2).

(a)

chapters 3 and 4, t(48) = 3.22, p < .005, d = 0.93. The high rate of requests in the second experiment makes one wonder to what degree the direct demonstration participants were mastering the material in the later chapters. (Participants in the discovery condition could not ask for hints in either experiment.) Both discovery and direct demonstration participants seemed to be suffering from the lack of earlier practice when they came to these later chapters.

There were no major effects of the presence of global instructions, but there were large effects of whether the participants were in a discovery condition or were receiving directions about the individual steps of the problem. We decided to focus further analysis on this factor. Because all participants completed the first two chapters and hint requests were low for these chapters, we decided to focus on them. All the effects of the discovery condition were already in place for these two chapters. Because the effects in this experiment contrasted so sharply with the effects in the first experiment, we decided to perform a set of analyses that merged the two experiments. As the three instructional conditions of the first experiment showed few differences, we merged them into a single instruction condition and contrasted them with the discovery condition. Thus, our analysis consists of 80 participants who could be classified according to whether they were in an instruction or a discovery condition and whether they received long practice periods or short practice periods. Besides these two betweenparticipant factors, there are the within-participant factors of chapters (1 vs. 2) and position of problem in section (first two vs. the rest). In the first two chapters, participants solved 21 problems in the short condition and 84 problems in the long condition. The first two problems were the same in the sections, and the later problems in the short condition were a subset of the later problems participants solved in the long condition. In these analyses we look only at the 21 problems that participants in both experiments solved in common.

As already noted, the students in the two experiments were drawn from the same undergraduate population, and there was no difference in their prior algebra scores. To ensure that the long and short conditions were equivalent, we looked at the first two problems for section 1.1. These problems appeared before there were any differences in practice. The mean time to solve these two problems was 57.3 s in the long instruction condition, 50.6 s in the short instruction condition, 77.9 s in the long discovery condition, and 87.9 s in the short discovery condition. The difference between instruction and discovery was highly significant, t(76) = 3.35, p < .005, d = 0.77, but the effect of practice length was not, t(76) = 0.20, d = 0.05, nor was the interaction between practice length and instruction, F(1, 76) = 0.94.

Given that the two populations are equivalent, it is significant that 50% of the participants quit the discovery condition in Experiment 2 but none did in Experiment 1. This is quite a significant difference, $\chi^2(1, N = 30) = 7.50$, p < .01. Nonetheless, with respect to the combined analyses that follow, it should also be noted that we are only looking at the first two chapters before there is any participant drop-out in the second experiment.

We performed the same analyses on the combined data as reported in Figures 6 and 7 for Experiment 1. Figure 9 presents the breakdown of total time to solve a problem into the number of transformations that participants perform and the time per transformation. Number of transformations (Figure 9a) shows an interaction between practice and instruction, F(1, 76) = 5.69, p < .05,

Figure 9. Mean time per transformation (a) and mean number of transformations (b) as a function of instructional condition, chapter, and whether the problems were the first instructional problems in a section or later practice problems (Experiments 1 and 2 combined).

but not an interaction between position and instruction, F(1, 76) = 0.12. Time per transformation (Figure 9b) shows an interaction between position and instruction, F(1, 76) = 21.19, p < .0001, but not an interaction between practice and instruction, F(1, 76) = 0.94. The conclusion from this figure is that the difference between the two experiments resides in the fact that discovery students in the second experiment were making a good many incorrect transformations that had to be corrected (Figure 9a).

Figure 10 shows a classification of the mean errors of the two main types. Figure 10a shows the mean number of operator errors per problem. There are strong two-way interactions between practice and instruction, F(1, 76) = 22.65, p < .0001; position and instruction, F(1, 76) = 32.44, p < .0001; and position and practice, F(1, 76) = 15.02; p < .0005. Moreover, the three-way interaction between these factors is highly significant, F(1, 76) = 14.81, p < .0005. This three-way interaction reflects the fact that participants were making many more operator errors on initial problems in the short discovery condition than any other condition. Figure 10b shows the number of transformation errors. There are two-way interactions between practice and instruction, F(1, 76) =





Figure 10. Mean number of operator errors (a) and transformation errors (b) as a function of instructional condition, chapter, and whether the problems were the first instructional problems in a section or later practice problems (Experiments 1 and 2 combined).

6.03, p < .05, and position and instruction, F(1, 76) = 8.30, p < .01. Moreover, the three-way interaction between these factors is highly significant, F(1, 76) = 16.63, p < .0005. This three-way interaction reflects the fact that participants were making many more transformation errors on later problems in the short discovery condition than in any other condition. Our characterization of this figure is that short discovery students were having much greater difficulty in identifying the correct transformations on early problems in a section (Figure 10a), and this led to a residual difficulty on later problems that shows up in transformation errors (Figure 10b). Given more practice, the students in the long condition did not have this difficulty.

Detailed Analysis of Initial Problems in Two Sections

The above analysis suggested that the difficulty of the short discovery condition began with the first problems in a section. For further insight into the initial learning in a section, Figure 11 presents a detailed analysis of behavior on the very first problems in sections 1.7 (data flow equivalent of single transformation equations) and 2.6 (combining constant terms). These two sections

are distinguished by the fact that they each involve exactly two transformations; the first one is new-data-flow equivalents of rewriting x + 3 = 8 as x = 8 - 3 in section 1.7, and 3 + (2x + 3)7) as 2x + (3 + 7) in section 2.6—whereas the second involves the evaluation transformation (data-flow equivalents of rewriting 8 -3 as 5 and 3 + 7 as 10) that they have been practicing from the beginning. (The two transformations for section 2.6 are illustrated in Figure 4.) Figure 11 displays the number of actions in excess of the minimum required taken by participants in the four conditions for each transformation. All the two-way interactions are quite significant between practice and instruction, F(1, 76) = 13.17, p < 100.0005; transformation and practice, F(1, 76) = 12.01, p < .001; and transformation and instruction, F(1, 76) = 15.09, p < .0005. Moreover, the three-way interaction between practice, instruction, and transformation is quite significant, F(1, 76) = 11.54, p < .005. Participants in the short discovery condition were having much greater difficulty with the first transformation than participants in any other condition and much greater difficulty with this transformation than they were having with the second transformation. Of particular note is the comparison of this group with the long discovery participants. Although the short discovery participants were somewhat worse than the long discovery participants on the second transformation, the difference is not significant, t(28) =1.25, d = 0.47. On the other hand, the difference for the first transformation is very large and significant, t(28) = 3.90, p <.001. d = 1.47.

From a certain perspective, it is surprising that the short discovery participants were showing the deficit on the first transformation, which is new, and not the second transformation, which is old. We might expect that the deficit due to lack of practice would show up on the old transformation because the participants had not had as much practice with it, or that the new transformation would be equally novel to both groups and that there would have been no difference. However, the short discovery participants wandered around much more in trying to discover what they need to do to achieve the first transformation in these problems.

The important conclusions of this experiment are with respect to the discovery condition, and we do not want to make very much of the performance of students in the instruction conditions. Any



Figure 11. Mean number of actions more than the minimum on the first problem in a section as a function of instructional condition, section, and transformation (Experiments 1 and 2 combined).

difficulties instruction students had could reflect on specific properties of the instructions rather than the merit of giving instructions at all. The instruction conditions really serve as a reference point for evaluating the discovery condition. The discovery condition was superior to this reference point in the first experiment when all students had ample opportunity to practice their operators, but it was inferior to this reference point in the second experiment when this practice was removed. In contrast, practice seemed to have a much smaller beneficial effect on participants in the instruction condition (at least for the first two chapters).

All the additional practice that students received after discovering the operators for a particular section prepared them to discover operators in later sections. Students in the long discovery condition required less than half as many actions to discover the new operators as students in the short discovery condition (Figure 11). Students in the short condition had such difficulty figuring out what to do that they often seemed unable to determine how they had gotten through the problem when they finally succeeded. Therefore, they often found themselves trying to discover the operators for a section on later problems (Figure 10b). In contrast, students in the long discovery condition engaged in very little searching after the first problems of a section. The deficits begin to snowball in the short discovery condition; students were learning little from the first problems, and there were too few further problems to remedy this deficit. Thus, they were thrown into later sections without mastering the earlier sections.

Why were students so much better at guessing what to do in the long discovery condition than in the short discovery condition? We think that the practice in the long discovery condition gave students a better sense of what actions to try in a new situation, because they had developed a better understanding of the semantics of the data-flow diagrams. For instance, consider the fact (see Figure 11) that short discovery students averaged over 100 actions to find out how to achieve the first transformation for the problem in section 2.6, whereas long discovery students took fewer than 20 actions. This transformation is illustrated in the transitions between states in Figures 4a and 4d. The first correct action is to select one of the two plus boxes (highlighted in red after correct selection in Figure 4b). All 10 of the students in the long discovery condition selected one of these boxes as their very first action, whereas only 5 of the 20 students in the short discovery condition did. Of the remaining 15 students, 14 selected the top box with the multiplication sign (*). This then led them into a part of the problem space that was confusing until they finally backed out of it by deselecting the top box. This confusing digression led to many of the extra actions for the short discovery condition. There is no possible operation in which the top box could be involved in a useful transformation—just as there is no way to usefully transform the 2x in the equivalent linear expression (2x + 7) + 3. Students in the long condition had enough experience with these data-flow diagrams to appreciate this fact, whereas the students in the short condition were driven by superficial features like the position of the box (they had been selecting top boxes up until this point).

General Discussion

Perhaps the most important outcome of this research is the demonstration of a circumstance where discovery, with some minimal guidance, can lead to successful learning. This positive outcome depends on three factors, which were true in the long discovery condition:

- 1. The searches involved in making the discoveries were sufficiently constrained that it was possible for students to find solutions and remember what they had done after they discovered a successful transformation.
- 2. The practice enabled students to understand the semantics of these data-flow diagrams. Because the most effective way to discover operators was to use the semantics to conjecture appropriate actions, discovery students were more likely to incorporate the mathematical constraints of the domain into what they learned.
- 3. Because of the combinatorial nature of the problems, students had to generalize what they learned on instruction problems to novel problems. Students did better at such generalizations if they were basing actions on mathematics of the diagrams rather than superficial features like positions of boxes.

Take away any of these features (constrained search space, practice, combinatorial domain structure) and we might not have observed the superior performance of discovery students. The second experiment showed that without practice, discovery students had a very poor sense of the domain semantics. It should also be noted that these effects were obtained with a particularly able group of students. Although other results with this tutoring system have generalized from Carnegie Mellon undergraduates learning data-flow graphs to high school students learning linear equations, it remains to be shown that this result generalizes.

There are other indications in the literature that discovery can be more effective in conditions of high practice. For instance, Dean and Kuhn (2006) found in the domain of science instruction that with little practice, discovery is inferior to direct instruction (replicating Klahr & Nigam, 2004), but with extended practice it becomes equivalent or superior. Somewhat related is the expertise reversal effect (Kalyuga, Ayres, Chandler, & Sweller, 2003): More knowledgeable students require less guidance to achieve successful learning. For examples, Tuovinen and Sweller (1999) found that practice eliminated the benefit of worked examples over exploratory learning, and Kalyuga, Chandler, Tuovinen, and Sweller (2001) found that exploratory learning became superior with more practice.

With respect to general implications, this research adds to the evidence that minimally guided discovery learning can be successful if the cognitive demands are limited. One of the benefits of discovery learning is that the processes of generating a solution can lead to a characterization of the domain that will help students generalize when they face new problem situations. There is nothing magical about discovery learning in this regard and certainly not about the particular version of the discovery condition that we implemented. For instance, we expect that we would have found every bit as much advantage if participants had been instructed at every point except when they had to enter values, leaving them to discover only what to type in. This "semidiscovery" condition might have been more efficient. It has also been proposed (Aleven & Koedinger, 2002; Roy & Chi, 2005) that the often-demonstrated advantage of self-explanation is that it encourages students to come up with correct characterization of transformations. Thus, requiring participants to generate explanations of the transformations might have been as beneficial as the discovery condition. In terms of designing instructional environments, the two critical features are that the environments do not overwhelm the cognitive resources of the student and that the discovery task encourages the student to encode the semantics that govern the combinatorial structure of the domain.

References

- Aleven, V., & Koedinger, K. R. (2002). An effective meta-cognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science*, 26, 147–179.
- Aleven, V., McLaren, B. M., Roll, I., & Koedinger, K. (2006). Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education*, 16, 101– 128.
- Anderson, J. R. (2007). How can the human mind occur in the physical universe? New York: Oxford University Press.
- Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 4, 167–207.
- Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., & Koedinger, K. (2008). Why students engage in "gaming the system" behavior in interactive learning environments. *Journal of Interactive Learning Research*, 19, 185–224.
- Baroody, A. J., Lai, M.-L., & Mix, K. S. (2006). The development of young children's number and operation sense and its implications for early childhood education. In B. Spodek & O. Saracho (Eds.), *Handbook* of research on the education of young children (pp. 187–221). Mahwah, NJ: Erlbaum.
- Bruner, J. S. (1961). The act of discovery. *Harvard Educational Review*, 31, 21–32.
- Charney, D. H., Reder, L. M., & Kusbit, G. W. (1990). Goal setting and procedure selection in acquiring computer skills: A comparison of tutorials, problem-solving, and learner exploration. *Cognition and In*struction, 7, 323–342.
- Cheng, P. W., Holyoak, K. J., Nisbett, R. E., & Oliver, L. M. (1986). Pragmatic versus syntactic approaches to training deductive reasoning. *Cognitive Psychology*, 18, 293–328.
- Cognitive Tutor. (2007). [Computer software]. Pittsburg, PA: Carnegie Learning.
- Dean, D., & Kuhn, D. (2006). Direct instruction vs. discovery: The long view. Science Education, 91, 384–397.
- Foerster, P. A. (1990). Algebra I (2nd ed.). Menlo Park, CA: Addison-Wesley.
- Fong, G. T., Krantz, D. H., & Nisbett, R. E. (1986). The effects of statistical training on thinking about everyday problems. *Cognitive Psychology*, 18, 253–292.
- Fuson, K. C., Wearne, D., Hiebert, J. C., Murray, H. G., Human, P. G., Oliver, A. I., et al. (1997). Children's conceptual structures for multidigit numbers and methods of multidigit addition and subtraction. *Journal for Research in Mathematics Education*, 28, 130–162.
- Heffernan, N. T., Koedinger, K. R., & Razzaq, L. (2008). Expanding the model-tracing architecture: A 3rd generation intelligent tutor for algebra symbolization. *International Journal of Artificial Intelligence in Education*, 18, 153–178.
- Hiebert, J., Carpenter, T. P., Fennema, E., Fuson, K. C., Human, P., Murray, H., et al. (1996). Problem solving as a basis for reform in

curriculum and instruction: The case of mathematics. *Educational Researcher*, 25, 12–21.

- Hmelo-Silver, C. E., Duncan, R. G., & Chinn, C. A. (2007). Scaffolding and achievement in problem-based and inquiry learning: A response to Kirschner, Sweller, and Clark (2006). *Educational Psychologist*, 42, 99–107.
- Kalyuga, S., Ayres, P., Chandler, P., & Sweller, J. (2003). Expertise reversal effect. *Educational Psychologist*, 38, 23–31.
- Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. *Journal of Educational Psychology*, 93, 579–588.
- Kamii, C., & Dominick, A. (1998). The harmful effects of algorithms in Grades 1–4. In L. J. Morrow & M. J. Kenney (Eds.), *The teaching and learning of algorithms in school mathematics: 1998 yearbook* (pp. 130–140). Reston, VA: National Council of Teachers of Mathematics.
- Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, 41, 75–86.
- Klahr, D., & Nigam, M. (2004). The equivalence of learning paths in early science instruction: Effects of direct instruction and discovery learning. *Psychological Science*, 15, 661–667.
- Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. (1997). Intelligent tutoring goes to school in the big city. *International Journal* of Artificial Intelligence in Education, 8, 30–43.
- Koedinger, K. R., & Corbett, A. T. (2006). Cognitive tutors: Technology bringing learning science to the classroom. In R. K. Sawyer (Ed.), *Handbook of the learning sciences* (pp. 61–78). New York: Cambridge University Press.
- Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? *American Psychologist*, 59, 14–19.
- Reed, S. K., & Bolstad, C. A. (1991). Use of examples and procedures in problem solving. *Journal of Experimental Psychology: Learning, Mem*ory, and Cognition, 17, 753–766.
- Ritter, S., Anderson, J. R., Koedinger, K. R., & Corbett, A. (2007). Cognitive tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review*, 14, 249–255.
- Ritter, S., Haverty, L., Koedinger, K., Hadley, W., & Corbett, A. (2008). Integrating intelligent software tutors with the mathematics classroom. In G. Blum & K. Heid (Eds.), *Research on technology and the teaching and learning of mathematics: Vol. 2. Cases and perspectives*. Charlotte, NC: Information Age.
- Rittle-Johnson, B. (2006). Promoting transfer: The effects of direct instruction and self-explanation. *Child Development*, 77, 1–15.
- Roy, M., & Chi, M. T. H. (2005). Self-explanation in a multi-media context. In R. Mayer (Ed.), *Cambridge handbook of multimedia learning* (pp. 271–286). Cambridge, United Kingdom: Cambridge University Press.
- Schmidt, H. G., Loyens, S. M. M., van Gog, T., & Paas, F. (2007). Problem-based learning is compatible with human cognitive architecture: Commentary on Kirschner, Sweller, and Clark (2006). *Educational Psychologist*, 42, 91–97.
- Shulman, L. S., & Keisler, E. R. (1966). *Learning by discovery*. Chicago: Rand McNally.
- Tuovinen, J. E., & Sweller, J. (1999). A comparison of cognitive load associated with discovery learning and worked examples. *Journal of Educational Psychology*, 91, 334–341.
- von Glasersfeld, E. (1995). Radical constructivism: A way of knowing and learning. Bristol, PA: Falmer Press.

Received November 2, 2008 Revision received April 27, 2009

Accepted April 30, 2009