

**When Minimal Guidance Does and Does Not Work: Drill and Kill Makes
Discovery Learning a Success**

Angela Brunstein

Shawn Betts

John R. Anderson

Psychology Department

Carnegie Mellon University

Tuesday, July 30, 2013

Abstract

Two experiments were performed contrasting discovery learning with a variety of different instructional conditions. Students learned to solve data-flow isomorphs of the standard algebra problems. In experiment 1, where students practiced each new operation extensively, they performed best in a Discovery condition. The Discovery condition forced participants to develop correct semantic characterizations of the algebraic transformations. In Experiment 2, where students practiced each operation minimally, they performed worst in the Discovery condition and most of them failed to complete the curriculum. With less practice students' attempts to discover transformations became less constrained and more random. This search for transformations became so extended that students were unable to remember how they achieved transformations and so failed to learn. These interpretations of the advantages and disadvantages of discovery learning were confirmed with a simulation model that was subjected to the various learning conditions. Discovery learning can lead to better learning outcomes only when the challenge posed by the demand of discovery does not overcome the student's resources.

There has been a long history of advocacy of discovery learning including such intellectual giants as Rousseau, Dewey & Piaget. Bruner (1961) is frequently credited as the source for the modern research on discovery learning in the last 50 years. While discovery learning continues to have its advocates (e.g., Fuson et al., 1997; Hiebert et al., 1996; Kamii & Dominick, 1998; Von Glasersfeld, 1995), there has been accumulating evidence and argument 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. criticisms of minimally guided learning, the authors (Hmselo-Silver et al., 2007 and Schmidt et al, 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 inquiry as minimally guided. The conclusion of the research to be reported here is that one cannot make blanket claims about the superiority or inferiority of discovery learning. Rather one must assess carefully the information-processing consequences of each learning situation. A careful reading of the Kirschner et al paper finds such a nuanced perspective and they note cases where discovery learning can lead to superior results. We will try to develop an understanding of the information-processing consequences of the learning conditions we are studying by developing computer simulation models that reproduce the basic effects of our experiments.

Many domains have a sufficiently rich combinatorial structure that it is not possible to provide students with direct instruction on all possible cases. They have to generalize what they learn on specific cases to new cases. For instance, in this research after students learn to rewrite $(4 + x) + 3$ as $7 + x$ they are given the problem $(5 + x) - 3$. While the majority of students correctly generalize to this problem, a significant minority display the error $2 - x$. Making the correct

generalization to this case can be viewed as mini-discovery informed by knowledge of the constraints of algebra. This research will show that under some circumstances students are better situated to make this generalization if they have learned to solve the original problems in a discovery mode.

This research is part of an effort to understand the contribution of instructional content in cognitive tutors for mathematics. Figure 1 shows some screen images involving equation solving in the Carnegie Learning Tutor. Presentation-wise these are the simplest parts of the algebra curriculum but they reflect the general model of interaction with the Cognitive Tutor. In part (a) the student is presented with the equation $8y = 9 - (6y) + 9 = 10$ and the student selects an operation to perform from a pull-down menu – in this case, the student has erroneously selected “Distribute”, will receive feedback, and eventually chose the correct operation of “Add/Subtract Terms”. When this correct operation is chosen, the tutor presents a display like part (b) of Figure 1 where the student must indicate the result of adding like terms by filling in a series of boxes. The resulting equation is represented in part (c) and the student the student must choose a correct operation again. Upon doing so, the tutor once again presents a series of boxes in part (d) 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 provide instruction on those aspects.

The research to be reported here uses an experimental system for solution of linear equations that had this basic structure. Figure 2a illustrates the basic interface. The student selects parts of these equations by pointing and clicking with a mouse. The selected portion is highlighted in red and the student picks operations to perform on that portion from the menu buttons to the right. In this case the student has chosen the two x-terms, selected “collect”, and a new line has been created with green boxes where information is to be entered. In Figure 2a the student has selected the smaller box and is about to type the operator $*$.

The curriculum is based on the material in the first four chapters of the classic algebra text by Foerster (1990). The overall interface and interaction structure has been reduced from the commercially available tutor to facilitate data analysis and to make it easier to run model simulations with. Nonetheless, the basic character of the interaction is similar. There are some simple help options should a student get stuck: There is a “hint” button that the student can click to receive instruction on what to do next, a button to go back to where the first mistake was made, and arrows for moving back and forward in the problem.

For purposes of exploring instructional design an isomorph of algebra was created that can be used with adults. If these adults fail to learn the algebra isomorph (as they do in some instructional conditions) it will be at no cost to their competence in real algebra. Figure 2b illustrates the data-flow interface for a comparable point to Figure 2a. Students point to boxes in this graph, select operations, and key in results. The actual motor actions are isomorphic to the actions for a linear equation and in many cases physically identical. Figures 3a and 3b shows data-flow equivalents of a relatively simple equation and a relative complex equation in this

system. Part (a) is the isomorph of the equation $5x + 4 = 39$ and part (b) is the isomorph of the equation $(2x - 5x) + 13 + 9x = 67$. In such a diagram, a number comes in the top box, flows through a set of arithmetic operations, and the result is the number that appears in the bottom. 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 3, these simplifications will result in a box with the input value. This is the equivalent to solving for x . However, some diagrams (e.g. see Figure 4) 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, Chapter 5) reports a behavioral comparison of children working with linear equations and adults working with the data-flow tutor. While children were a bit more error prone, they behaved very similarly.

Experiment 1

The tutor provides students with instruction on each step of a problem that involves a mix of a verbal direction and worked example. The first experiment to be reported here was an attempt to assess separately the contribution of the instruction and worked example. There was a *Verbal Direction* condition in which participants just received abstract verbal instruction without any specific directions about how to solve a specific problem and a *Direct Demonstration* condition in which participants were told what to do in a specific case without stating any general characterization of the action. To complete a factorial design we crossed the use of verbal direction with direct demonstration. This created the *Both* condition that was somewhat like the

original condition of Anderson (2007) where both verbal directions are given as well as direct 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 experiments have tended not to look at situations in which the participants receive no direction as our Discovery condition.

Figure 4 illustrates that basic cycle that occurs throughout the curriculum for problem that is concerned with collection of like terms (Section 2.6 in the Foerster text). 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 $(7 + 3) + 2x$ and the second row shows steps in transforming this to $10 + 2x$. As the curriculum progresses the problems became more complex and require more varied transformations but always they had the character of the problem illustrated in Figure 4:

1. The diagram begins in some neutral display (parts a and d) and the student must select some boxes to operate on. Later problems could require selection of as many as 5 boxes and there could be a number of alternative correct choices about which sets of boxes to operate on next.
2. The selected boxes would be highlighted (parts b and e) and the student would select some operation by clicking a button to the right of the diagram.
3. The diagram would be transformed with a number of green boxes (parts c and f) and the student would type information into these boxes.

4. When the boxes were filled in, the diagram would return to a neutral state (parts d and g) ready for the next selection of some set of boxes.

When the transformations were complete the student would click the Next Problem button. If the transformations had been correctly performed the student could go onto the next problem. If there was an error, the student would be informed that she could not go on to the next problem but had to correct the error. The First Mistake button would take the student to the state of the diagram before the first mistake. The “->” and “<-” buttons allowed students to move back or forth a single transformation.

The material used in this experiment comes from 12 sections over 4 chapters in the Foerster text that covers what is needed to solve linear equations. The first 1 or 2 problems in each section were used for instruction. The problem in Figure 1 was used for instruction in Section 2.6 on combining. Table 1 shows the instruction that accompanied 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-state instruction. For Section 2.6, it will turn out that the most critical transformation is between states like c and d where the participant must specify the content of the boxes in a way that preserves the value of the graph structure. In the *Verbal Direction* condition participants would receive instruction like "Find two boxes with addition or subtraction and click them" which provided guidance as to how to perform the operation of this and similar problems without saying exactly what to do (for example in this case, the actual boxes had to be determined by the student). 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 while in the Discovery condition they saw neither.

Participants in the Discovery condition could try various operators and learn from the feedback they received. Specifically, there were the following sorts of feedback (which were also available in the other conditions):

- (1) If students tried an inappropriate operator for the boxes chosen they would receive the feedback that the operator chosen could not be applied to the boxes selected as in “Combine cannot be done with the selected boxes”.
- (2) On the first 1 or 2 problems for which other participants received instruction, if discovery participants entered an incorrect result into a green box that result would be rejected with the error “Your answer is incorrect”. On later problems participants in all conditions were allowed to make such transformation errors and go on with the problem in an error state. If the student made a transformation error and got to the end and asked for the next problem they would be presented with the message "Your answer is incorrect. Use the <- and -> buttons (or the left and right arrow keys) and the First Mistake button to review your work and correct the mistake."
- (3) If at any point they thought they were finished when they were not and asked for the next problem they received the message "You are not finished. You need to do more to the problem."

Thus, as in any Discovery condition there was some guidance but it is minimal. Students get some information, sometimes delayed, that their actions are wrong but no information about

what the correct actions are. Also, participants in all conditions saw a general statement of the purpose of the section (see initial general instructions in Table 1).

Method

Participants. Forty undergraduates (23 male and 17 female; $M = 23$ years, $SE = 1.6$ years) took part in this study. They reported relatively high last algebra grades (24 As, 8 Bs, 4 Cs, 4 missing). Students participated in three single participant sessions each lasting between one and 2.5 hours and were paid either per time (\$5 per half hour) or by performance (\$ 0.07 per correct performed operation in the tutor). Ten participants were randomly assigned to each of four conditions.

Materials. Altogether, participants solved 174 data-flow problems that require performing at least 674 operations. Below are the 12 sections and a description of the problems in their linear algebra equivalent:

- **Section 1.1. Evaluating Diagrams** (14 problems) teaches students how to evaluate the contents of boxes in the data flow diagrams – e.g., rewrite $(9 - 4) * 2$ as $5 * 2$ and this as 10.
- **Section 1.2. Input Boxes** (9 problems) teaches students to evaluate a diagram given a value for an input box – e.g. rewrite $(24 / x) - 1$, $x = 12$ 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 – e.g., 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 like operations – e.g., 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 – e.g., $2x + 3 = 19$, and to deal with asymmetric operators – e.g., rewrite $10 - x = 2$ as $x = 10 - 2$ and this as $x = 8$.
- **Section 3.1. Reordering Operations** (6 problems) teaches students the graph equivalent of distribution – e.g., rewrite $5*(x + 2) + 9$ as $(5x + (5 * 2)) + 9$ as $(5x + 10) + 9$ as $(10 + 9) + 5x$ as $19 + 5x$.
- **Section 3.2. Reordering and Subtraction** (9 problems) teaches students to use reorder with subtraction in problems such as $9 - 2*(x - 4)$.
- **Section 3.4. Combining multiple input boxes** (13 problems) teaches students the equivalent of collecting variable terms – e.g., rewrite $7x + 5x$ as $(7 + 5)*x$ as $12x$ and rewrite $5x + (6 - 2x)$ as $6 + (5 - 2)*x$ as $6 + 3x$.
- **Section 3.5. More on Combining input boxes** (12 problems) deals with special cases like $2x + x$ and $(6x + 3) - (6 - 2x)$.
- **Section 4.1. Finding Input Values in more Complex Problems** (11 problems) puts the operations together building up to solve equations like $((3x + 4) + 5x) + 6x = 32$.
- **Section 4.2. Finding Input Values in more Harder Problems** (21 problems) builds up to equations like $3*(2x - 1) + 2*(x + 5) = 55$.
- **Section 4.3. Finding Input Values when Two Data Flow Diagrams are Equal** (18 problems) presents equations like $3x + 55 = 8x$.

Procedure. There are 7 basic operations to be mastered (the 7 buttons to the immediate left of the data-flow diagram in Figure 2b). In the first session, participants performed the first five sections of the tutor that introduced *evaluate*, *invert* (unwind), and *combine* (collect) operators. This session took on average one hour. In the second session, participants performed the next four sections introducing *reorder* (distribute), *canonicalize*, and *undo-minus* operators. In the third and last session, participants performed the remaining three sections of the tutor extending earlier operators and introducing the *subtract* operator. The second and third sessions took on average approximately 1.5 hours.

The first problem in each section involved guided instruction like that in Table 1. 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 would often flounder on the second problem and request instruction. For these reasons, we will treat the first two problems as the instructional problems and the remainder as the practice problems. In all conditions but the Discovery condition, participants could click a hint button to request instruction on any problem.

Results

Figure 5 shows the mean total time (time from completion of previous to successful clicking of Next Problem 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 time to solve problems in different chapters reflecting the different number of

transformations required to solve a problem. We will ignore the factor of chapter in our statistical analyses and simply using graphs like Figure 5 to show that the basic effects replicate over chapters. Therefore, our statistical analyses are 4 x 2 ANOVAs with the factors being instructional condition and position in section (first 2 problems versus later problems). In the case of total time, there are no significant effects of instructional condition ($F(3,36) = 1.29, p > .25$; $MSE = 2598$) or position ($F(1,36) = 0.30, MSE = 554$) 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 ($F(1,36) = 53.17, p < .0001$) while the residual effects in the interaction are not significant ($F(2,36) = .40$). It is not surprising that participants have difficulty on the initial couple of problems in the Discovery condition. What is interesting is their apparently high performance on the remaining problems. Individual t-tests confirm that the Discovery condition is statistically superior to the Both and the Verbal Direction conditions ($t(18) = 2.78, p < .05$ and $t(18) = 3.35, p < .005$) on the rest of the problems in the section, but the difference between Direct Demonstration and Discovery does not reach significance ($t(18) = 1.40, p < .20$).

The total time to solve problems can be decomposed into the number of transformations that participants perform and the time per transformation. These measures are shown in Figure 6. Part (a) of Figure 6 shows the number of transformations and, for reference, the minimum number of transformations required if performance was always perfect. There are main effects of condition ($F(3,36) = 3.62, p < .05$; $MSE = 0.915$), 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 tended to involve more transformations. The interaction reflects the fact that there is almost no effect of condition on the first two problems while the conditions separate on later problems in a section. Participants do not do much more than the minimum number of transformations on the first two problems because of the special guidance. On the remaining problems the Discovery condition shows the fewest number of transformations. A contrast for this effect is highly significant ($F(1,36) = 19.51, p < .0001$) while 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: $t(18) = 4.05, p < .001$; Verbal Direction: $t(18) = 3.39, p < .005$; Direct Demonstration: $t(18) = 3.74, p < .005$).

Part (b) of Figure 6 shows the time per transformation¹. The effect of condition is not significant ($F(3,36) = 1.69, p < .05$; $MSE = 170.15$) while the effect of position is ($F(1,36) = 136.71, p < .0001, MSE = 55.42$). The effect of position reflects the speed up with practice that had been documented in Figure 4. 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 ($F(1,36) = 27.34, p < .0001$) while the residual effects are not ($F(2,36) = 0.76$). However, this time the effect mainly comes from the slower performance on the initial transformations where participants discover how to perform the new transformations. This effect is particularly pronounced for the first two chapters where most of the operations in

¹ 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 are not new.

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 p 's $> .10$). Thus, the total time results for the Discovery condition in Figure 5 reflect a decrease in the number of transformations performed (Figure 6a) rather than a speed-up in the time per transformation (Figure 6b).

There are two major categories of errors in the tutor. The first, the operator error, involves selecting a wrong operator for the boxes chosen (the transitions from state (b) to state (c) and from state (e) to state (f) in Figure 4). 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. If the participant selects a correct set of boxes and an appropriate operator, one or more green entry boxes will appear as in Figures 5c and 5f. The second type of error, the transformation error, involves entering the wrong values for these boxes. The tutor will accept these wrong values and transition to the next state (e.g., wrong versions of states (d) and (g) in Figure 4). It is only this category that the tutor does not reject and, as a consequence, this is the category that can lead to the need for extra transformations when the students try to correct their mistake. The first category will just lengthen the duration of a transformation as the students try again for a different box-operator combination and so should impact the performance measure in Figure 6b. The second category will increase the number of transformations in Figure 6a. These two categories of errors are presented in Figure 7.

Part (a) of Figure 7 shows the mean number of operator errors per problem. The effect of condition is significant ($F(3,36) = 6.25, p < .005; MSE = 2.67$) while 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 are doing on the first problems where they have to discover box-operator combinations. The main effect of condition also reflects this effect on the first problems. Again, a contrast for this effect on the first problems is highly significant ($F(1,36) = 61.02, p < .0001$) while 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 p 's $> .10$).

Part (b) of Figure 7 shows the number of transformation errors. The effect of condition is marginally significant ($F(3,36) = 2.54, p < .10; MSE = 0.189$) while the effect of position is quite significant ($F(1,36) = 38.16, p < .0001, MSE = 0.152$) 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 while the Discovery condition is better on later problems where there is more opportunity for wrong transformations. Again, a contrast for this effect is significant ($F(1,36) = 6.88, p < .05$) while the residual effects in the interaction are marginal ($F(2,36) = 2.64; p < .10$). Individual t-tests confirm that the Discovery condition is statistically superior to all conditions on the rest of the problems in the sections (Both: $t(18) = 2.88, p < .01$; Verbal Direction: $t(18) = 2.48, p < .05$; Direct Demonstration: $t(18) = 2.56, p < .05$).

In summary, after the first couple of learning problems the Discovery condition enjoys a surprising 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 minutes to go through all 174 problems whereas the average in the other conditions is 226 minutes – an advantage of over half an hour that is quite significant ($t(38) = 3.40, p < .005$). The advantage of the Discovery condition can be traced to the fewer number of 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 about how to correct themselves when they are in an error state.

Detailed Analysis of Transformation Errors Section 2.6

The transformation errors in Figure 7b reflect most fundamentally on an understanding of the domain because a transformation error (e.g. transforming the graph equivalent of $3 - x = 2$ into $x = 2 + 3$) changes the value of numbers that pass through the diagram. It is the equivalent of changing the value of the solution to an equation or the value of an expression. To illustrate the source of these errors we conducted a detailed analysis of Section 2.6 whose sequence of steps was illustrated in Figure 4 and whose instruction was given in Table 1. It is a relatively simple section where the Discovery students showed consistent advantages over the other students. The section consisted of 20 problems. On the final 18 problems, Discovery students took 24.6 seconds for solutions compared to 39.2 seconds for the other conditions and they made an average of 0.01 transformation errors per problem compared to an average of 0.20 in other conditions. In t-tests, Discovery is better than any of the other conditions on both statistics at

significance levels of at least .05. So, this relatively simple section is a microcosm of the whole learning experience.

To understand the advantage of the Discovery participants we need to understand the nature of these errors that they are avoiding. Therefore, we will focus our analysis on transformation errors in the other conditions. Figure 8 shows, problem by problem in the section, the mean number of transformation errors by the 30 non-discovery participants and a comparison to the predictions of a model that we will describe. When these participants made an transformation error over 95% of the time that error was an error in entering the value of the combination (transition from state (c) to state (d) in Figure 4) and over 95% of the time they continued until they submitted their wrong answer in state (g), received feedback, and asked what their first error was and started over again. The error rate in Figure 8 is definitely not uniform. Below we discuss the peaks above with a greater than .5 error rate:

- (1) Stated in linear algebraic form, the combine step in problem 4 corresponds to translating “ $(5 + x) - 3$ ” into “ $(5 - 3) + x$ ”. There were 14 students making errors. One participant made two distinct errors and so there were 15 distinct errors altogether. Another student repeated the same error 3 times. Not counting repetitions, the errors and their frequencies were twelve instances of “ $(5 - 3) - x$ ”, two instances of “ $(5 + 3) - x$ ”, and one instance of “ $(5 + 3) + x$ ”.
- (2) Stated in linear algebraic form, the combine step in problem 6 corresponds to translating “ $(15 - x) + 9$ ” into “ $(15 + 9) - x$ ”. 18 participants made 20 distinct combine errors and there were four repetitions of the same error. The non-repeated errors and their

frequencies were twelve instances of $(15 - 9) + x$, five instances of $(15 - 9) - x$, and three instances of $(15 + 9) + x$.

- (3) Stated in linear algebraic form, the combine step in problem 17 corresponds to translating $(54 * x) / 9$ into $(54 / 9) * x$. Six participants made nine distinct combine errors with eight repetitions. The non-repeated errors and their frequencies were five instances of $(54 * 9) / x$, two instances of $(54 / 9) / x$, one instance of $(9 / 54) * x$, and one instance of $(9 * 54) * x$.

The common feature in the majority of errors that participants make for each of these problems is that they use the main operator from the first form as the main operator for the second form (“-“ in case 1 above, “+” in the second case, and “/” in the third case). This erroneous generalization is quite tempting in a data-flow-form representation because the box that holds the main operator after the transformation is in the same place on the screen as before the transformation. Figure 8 also shows the results we obtained from a simulation model that we will describe in more detail after the second experiment.² The model had a 50% probability of making an incorrect interpretation of the instructions for this transformation. It only makes errors on those trials for which this interpretation leads to the wrong answer. Each time it makes an error it gets another opportunity to reinterpret the instruction. It slowly is able to correct these errors as it reinterprets the instructions and reinforces the correct interpretation. The correlation between model and data is .8 and it is able to capture all the peaks in the data but one.

² Instructions on how to run this simulation are available along with the model and the experimental software at a website to be revealed when author identity can be revealed. The instructions are in the file read&start.lisp.

The simulation does not make such errors in the discovery condition because the only way it can decide what to enter is by determining what will preserve the value of graph. In contrast, instructed participants can interpret the verbal directions and demonstrated solutions in multiple ways, only some of which always correspond to value-preserving transformations. This difficulty is a fundamental problem in a combinatorial domain like algebra where one cannot train each possible algebraic form. The students must be able to generalize the examples on which they receive instruction.

As an indication that some participants in the Instruction conditions are making the correct generalizations, Table 2 presents a classification of the 40 participants in the experiment according to their condition and the four quartiles for error rates (on transformations on all problems in the rest of all sections) . Participants in all conditions can be found in the top three quartiles. Thus, it is not that no participant in the direct Instruction conditions could come to correct characterizations of the transformations; it is just that some chose other ways of characterizing the transformation. Discovery learning succeeds in this case because it forces participants the only way they can discover the transformations is by exploring operations that preserve the semantics of the equations.

Discovery Learning in Detail

Section 1-7 is a good section for illustrating how students discover the transformations in the first place. It involves a relatively simple problem but nonetheless one for which Discovery students engage in a fair amount of search on the first problem. There are two transformations

involved. Part (a) of Figure 9 illustrates the search for the first transformation, the invert transformation, which is new to the section. Part (b) of the figure illustrates the search for the evaluation operation, which has been used in the previous two sections, but in response to a different graphical structure. The figure illustrates the various states of the tutor for the first problem in section 1-7 and the mean proportion of transitions going from each state to the various other states that can be directly reached (first proportions for the experiment, second for Experiment 2). Many of the states have transitions going back to that same state. This is because many of the actions that students take do not result in any change to the state of the tutor. The proportion of such no-change actions is given on these circular arrows.

The problem starts out as illustrated in the box in the upper left of Figure 9a. The only actions that participants can perform that result in any state changes are clicking boxes. Clicking a black box selects it and turns it red while clicking a red box unselects it and turns it back to black. In addition to clicking boxes, participants click operation buttons (particularly the invert button) or try to type something (particularly the value of the problem, 5). Such actions fail to change the state change except for clicking the invert button when both boxes are selected. As can be seen the students are wandering almost randomly among the 4 states defined by which of the two boxes are red. If they do select the “invert” operator when the two boxes are red they move to a state in which a green box appears. If they click the green box an entry box will appear and it is at this point that they have to enter the critical transformation. While they only enter the correct transformation “ $8 - 3$ ” a little more than 20% of the time, their other choices are anything but random. The only other expressions they enter are “ $5 + 3$ ” and “ $3 + 5$ ”. Their other errors are just interface errors as they try to enter one of these expressions.

There is some search in part b for the evaluation operation. While they have done evaluation operations before, the layout of the boxes is different and it has never been after an invert transformation. To perform the operation they need to select just the box to be evaluated and click the evaluate operator. As Figure 9b illustrates, there is some tendency to select the bottom box which invalidates the evaluate operator. When they do click the evaluate operator for the correct single box, a little green box appears where the value will go. At this point they seem relatively successfully in applying the rest of the evaluation procedure and in particular, almost always correctly make the transformation from 8-3 to 5.

As these figures illustrate, in novel states the participants' discovery process can be characterized as random search among a set of interface actions but semantically informed choices about the values to enter for the transformations. While there is this extended search through the interface options on this first problem, participants are able to learn enough from this one exploration that they are better than participants in the other conditions after this one problem.

Experiment 2

The first experiment seemed to reveal very rapid learning after a single discovery episode. The later problems in a section gave us evidence of what participants had learned but did not seem to be important to learning. For instance, discovery students average 26.85 seconds per transformation on the first two problems, 10.89 seconds on the next two, and 11.87 second on the last two. Thus, there seems no speed up after the first two problems. With respect to the critical transformation errors, they are making a low 2.1% transformation errors per opportunity on the

second two problems (it is hard to make transformation errors on the very first two because of the interface) and 1.7% on the last problems. So, it would appear that we could have gotten the benefit of discovery learning with many fewer problems – that it could have been achieved without all of this drill and kill. However, we suspected that practice was giving participants a familiarity with the overall system that enabled them to learn so effectively from discovery. To investigate this, the second experiment greatly reduced the number of problems from 174 to a 45. We kept the same first two problems for each of the twelve sections but only used 21 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 sect 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, 1 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 performed to investigate a second issue about the first experiment. As Table 1 indicated, 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 * or / operators. We wanted to determine the contribution of these general instructions to learning.

There were no dramatic differences among the three instruction conditions in the first experiment. Therefore, this experiment just used one of the conditions, contrasted 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 undergraduates (27 male and 13 female; $M = 23$ years, $SE = 2.1$ years) took part in this study. Students participated in one single session that lasted a maximum of 2.5 hours and were paid either per time (\$5 per half hour) or by performance (\$ 0.07 per correct performed operation in the tutor). Ten participants were randomly assigned to each of four conditions. They reported relative highly last algebra grades (20 As, 11 Bs, 2 Cs, 7missing data) – a very comparable distribution to the first experiment.

Procedure. Except for the 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

Qualitatively, results were very different in the Discovery conditions in this experiment than in the previous. Six participants quit in the Discovery condition with global instructions and four participants quit in the Discovery condition without global instructions. They reached points 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 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. This means only 1 participant completed the Discovery condition with global instructions and only 4 completed without global instructions. 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 ($\chi^2(1)=19.06$, $p < .0001$). While there was a slightly greater tendency for greater participant loss in the Discovery condition with global instructions this was not significant ($\chi^2(1)=2.40$). Figure 10 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 are being eliminated on later chapters, the discovery participants are 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 -- Rest of Chapter 3: Global worse than no global ($t(18) = 2.23$, $p < .05$).

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 were asking for a great many hints as they solve the rest-of-the-section problems in these chapters. In Chapters 1 & 2 they averaged 0.04 hint requests per problem while they average 3.76 for Chapters 3&4. For comparison, instructed participants in

Experiment 1 averaged .02 requests on the same problems for Chapters 1 & 2 while they averaged 1.25 for Chapters 3&4. The difference between experiments is not significant for Chapters 1&2 ($t(48) = 1.41$) while it is highly significant for Chapters 3&4 ($t(48) = 3.22$, $p < .005$). Participants in the Discovery condition could not ask for hints. While the problems in the Chapters 3 & 4 are longer and more difficult, creating the opportunity for hint requests even in the first experiment, 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. 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 are large effects of whether the participants were in a Discovery condition or are receiving directions about the individual steps of the problem. We decided to focus further analysis on this factor. Since all participants completed the first two chapters and hint requests were low for these chapters we decided to focus on the first two chapters. All the effects of discovery were already in place for these two chapters. Since the effects in this experiment were in such a contrast 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 decided to merge them into a single instructional condition and contrasted them with the Discovery condition. Thus, our analysis consists of 80 participants that could be classified according to whether they received instruction or were in a Discovery condition and whether they received long practice periods or short practice periods. Besides these two between-participant factors there are the within-participant factors of chapters (1 versus 2) and position (first 2 versus rest).

In these analyses we will only look at the problems that participants in the both experiments solved in common. 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.

The first thing we wanted to establish is that the Long and Short conditions started out the same before the practice manipulation set in. We can test this by looking at the first two problems for Section 1.1. The mean time to solve these two was 57.3 seconds problems in the Long Instruction condition, 50.6 seconds in the Short Instruction condition, 77.9 seconds in the Long Discovery condition, and 87.9 second in the Short Discovery condition. The difference between instruction and discovery was highly significant ($F(1,76) = 11.19$; $p < .005$) but the effect of practice length was not ($F(1,76) = .04$) nor was the interaction between practice length and instruction ($F(1,76) = .94$).

Figure 11 shows the total time to solve an equation broken out into the 4 binary factors of the experiment. For brevity we will again refrain from reporting statistical tests for chapters, but just use this to note that the effects are in the same directions for the two chapters. All the main effects are significant but it is the interactions among the other three factors that are critical. The interaction between practice and instruction is significant ($F(1,76) = 8.53$; $p < .005$) as is the interaction between position in the section and instruction ($F(1,76) = 10.45$; $p < .005$). The first interaction reflects the fact that the Discovery condition is doing much worse in the condition of short practice. The second interaction reflects the fact that the disadvantage of discovery is much greater for the first two problems where participants are figuring out the operators. There

is not a significant 3-way interaction between practice, instruction, and position ($F(1,76) = 1.53$; $p > .2$).

Figure 12 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 (part a of Figure 12) shows the same interaction as total time between practice and instruction ($F(1,76) = 5.69$; $p < .05$) but not the interaction between position and instruction ($F(1,76) = 0.12$). This interaction indicates that participants in the Short Discovery condition are considerably more lost on the same problems. Time per transformation (part b of Figure 12) shows the interaction between position and instruction ($F(1,76) = 21.19$; $p < .0001$) but not the interaction between practice and instruction ($F(1,76) = 0.94$). The interaction of instruction with position reflects the substantial search participants in the Discovery condition have to engage in on the initial problems of a section.

Figure 13 shows a classification of the mean errors of the two main types. Part (a) shows the mean number of operator errors per problem. There are strong 2-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 ($F(1,76) = 14.81$; $p < .0005$). By this measure participants are having much more difficulty on initial problems in the Short Discovery condition than any other condition. Part (b) of Figure 13 shows the number of transformation errors. There are 2-way interactions between practice and instruction ($F(1,76) = 6.03$; $p < .05$) and position and instruction ($F(1,76) = 8.30$, $p < .01$) Moreover, the three-way interaction

between these factors is highly significant ($F(1,76) = 16.63$; $p < .0005$). Participants are making many more transformation errors on the Short Discovery condition on the later problems in a section.

Figure 14 presents a detailed analysis of behavior on the very first problem in Sections 1.7 and 2.6 which illustrate something that we think gets to the heart of difficulty participants are suffering in the Short Discovery condition. These two sections are distinguished by the fact that they all involve exactly two transformations – the first one is new (invert in Section 1.7 and combine in Section 2.6) while the second involves the evaluation transformation that they have been practicing from the beginning. Figure 14 displays the number of actions more than the minimum required taken by participants in the four conditions for each transformation³. All the two-way interactions are quite significant: between practice and instruction ($F(1,76) = 13.17$; $p < .0005$), 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 among practice, instruction, and transformation is quite significant ($F(1,76) = 11.54$; $p < .005$). It is clear that participants in the Short Discovery condition are having much greater difficulty with the first transformation than are participants in any other condition and much greater difficulty than they having with the second transformation. Of particular note is the comparison of this group with

³ For the transformations in Section 2.6, the minimum number of actions for the first transformation (going from state (a) to (d) in Figure 4) is 7: two boxes must be selected to get from (a) to (b), the combine operator must be selected to get from (b) to (c), and the two green boxes must be separately selected and then the results entered to get from state (c) to (d). The minimum number of actions for the second transformation (going from state (d) to (g)) is 4: –one box must be selected to get from (d) to (e), the evaluate operator must be selected to get from (e) to (f), and the green box must be selected and the result entered to get from state (f) to (g). The corresponding minimum number of actions for Section 1.7 is 5 for the first, invert transformation and 4 again for the second, evaluation transformation.

the Long Discovery participants. While they are somewhat worse than the Long Discovery participants on the second transformation, the difference is not significant ($t(28) = 1.25$). On the other hand, the difference for the first transformation is very large and significant ($t(28) = 3.90$; $p < .001$).

From a certain perspective it is surprising that the short participants are showing the deficit on the first transformation, which is new, and not the second transformation, which is old. One might have expected that the deficit due to lack of practice would show up on the old transformation because they have not had as much practice with it. One might have thought that the new transformation would be equally novel to both groups and that there would have been no difference. However, the short participants wander around a great deal more in trying to discover what they need to do to achieve the first transformation in these problems. We will discuss the source of this difference and other differences between the experiments in the discussion.

Discussion

Frequently advocates of discovery learning are also critics of “drill and kill” programs that involve a lot of practice. However, we found that discovery students only succeeded when they had a lot of practice. The most successful students came from the condition of high practice and discovery and the least successful students came conditions of low practice and discovery. In contrast, practice seemed to have little beneficial effect on participants with direct instruction (at least for the first two chapters).

In an attempt to understand why practice was so important to discovery learning we made some modifications to an ACT-R model of this task developed by Anderson (2007). ACT-R (Anderson et al., 2004) is a cognitive architecture that has been used to model people's behaviors in a great variety of tasks including categorization, learning algebra and geometry, driving while talking on a cell phone, and air traffic control (see <http://act-r.psy.cmu.edu/> for the variety of tasks researchers have modeled). By specifying behavioral models within such a framework, one is forced to make the theory computationally explicit, thus allowing for true evaluation of the theory as well as allowing for predictions in novel circumstances. According to the ACT-R theory, cognition emerges through the interaction of a number of relatively independent modules. Each of these modules reflects a set of assumptions derived from the empirical literature. By using the same modules for a wide range of tasks one can test the generality of these assumptions

The original ACT-R student model had simple parsing rules for converting instruction into declarative representation of operators for transforming equations. It also had a set of simple inference rules for extracting such representations from worked examples. ACT-R has a general system for using declarative representations of operators to solve problems in any domain (see Anderson et al, 2004; Anderson, 2005, 2007; Taatgen et al. in press). We have made two modifications to this model to accommodate the results of this experiment, one for the direct Instruction condition and one for the discovery learning condition:

1. **Direct Instruction.** To model the transformation errors that participants were making in the Instruction condition we made the interpretation of instructions for the critical transformation in section 2.6 random between a correct interpretation of the instruction and a superficial characterization of the transformation. This was the basis for the

detailed model performance in Figure 8. Note that this implies, consistent with Figure 13b, that the difference between the Discovery and Instruction conditions in erroneous will first appear in Chapter 2. With this one change the model was able to simulate the performance of participants through the first 4 sections of the material. We would need to create further superficial instruction interpretations to simulate the later sections.

2. **Discovery Learning.** To model the Discovery condition we had to provide ACT-R with a means for discovering the transformations. We provided the model with a set of operators that guessed various actions subject to the constraint that they preserved the semantics of the data-flow diagrams (again just for the first 4 sections). The probability that any guessing operator applied in a situation depended on its similarity to that situation. We set these similarities to match the distribution of choices observed in the participants, and therefore set different values for the Short and Long conditions for Sections 1.7 and 2.6. The model randomly guessed actions according to these probabilities until it eventually found a sequence of actions that produced a successful transformation. Since the model less often chose useful operators in the Short condition it had to engage in more search before discovering a new transformation.

We ran this model on the first four sections of material (Sections 1.1, 1.2, 1.7, and 2.6). For comparison we also ran the instruction model in the condition of direct demonstration. Figure 15 provides a problem-by-problem comparison of the data and model for the various problems in

the first four sections⁴. In part (a) it compares the data and predictions for the Instruction condition and in part (b) it compares the data and predictions for the Discovery condition.

Part (a) of Figure 15 for the Instruction condition can be summarized with four observations:

1. The largest discrepancy between model and data concerns performance of the participants in the Short condition for the last problem in Section 1.7 (without the data point the correlation improves from .603 to .764). Participants are much worse than the model predicts and much worse than participants in the Long condition. For participants in the Short condition this is the first problem that has involved a fraction and they have difficulty figuring out how to enter fractions into the interface. Participants in the Long condition were able to learn how to enter fractions on earlier problems. These earlier problems are not shown in Figure 15 because that figure only shows the problems in common. The model starts out knowing how to enter fractions and so does not predict this difficulty.
2. The model only predicts one major difference between the Short and Long conditions and this prediction is confirmed. This is the last problem on Section 2.6, which is the only opportunity to manifest the operator error analyzed in Figure 8 for the Long Instruction condition. The model and participants in the Short condition are struggling much more with this problem than in the Long condition. They are not having the same difficulty in the Long condition because they have had earlier opportunities to learn from cases where the major operator is not the same after the transformation.

⁴ Instructions on how to run this simulation are available along with the model and the experimental software at a website to be revealed when author identity can be revealed. The instructions are in the file `read&start.lisp`.

3. For the remaining problems in Figure 14a the model correctly predicts that participants are only slightly worse in the Short condition than the example condition. This slight difference reflects the slower retrieval times for the less practiced declarative information.

It seems in cases where there is direct instruction the advantage of practice is just to expose participants to cases that were not explicitly covered in the instruction.

Part (b) of Figure 15 for the Discovery condition can be summarized with four observations, all but the first are successfully predicted by the model:

1. As in part (a) participants in the Short condition have problems with the last problem in Section 1.7, which is their first encounter with fractions. This is the same phenomenon as observed above in point 2 for the direct demonstration condition.
2. There is very little difference predicted or observed between the Long and Short conditions for the first two sections. This reflects the fact that participants in the two conditions start out approximately equally.
3. However, the next two sections show substantially worse performance for participants in the Short condition on the first problems in these sections. The model predicts this because of the greater search.
4. The deficits from their struggle with the first problem in a section extend to later problems in the section, but the deficits are not as large. The model tries to learn from its exploration on the first problem by tracing back its actions when it finally succeeds in producing a transformation. However, when the exploration gets too long the model can no longer recall all the past steps (which are stored as elements in ACT-R's declarative

memory). Therefore, the search for correct operators tends to repeat itself on later problems. Moreover, there are not enough later problems to allow the model completely master the material for a section before even more new material is added in later sections.

So the deficits begin to snowball.

In summary, the Discovery condition in the Short condition is at two deficits – it is not learning as much from the first problems and it does not have enough further problems to remedy this first deficit.

The Short Discovery condition involves poorer guesses of operators in searching for a transformation. The differences between the Short and Long Discovery conditions largely result from this difference. We think that in the Long condition discovery participants are learning how to discover new transformations. We think this takes the form of learning what not to do. For instance, given the interface it never is a good idea to start a problem by typing anything or selecting an operator. One always starts by selecting boxes in the data-flow diagram.

Participants in the Long condition have learned this by the time they reach Section 2.6 while participants in the Short condition have not. The learning goes beyond learning to discover interface actions. It also extends to learning the critical transformations. While subjects in the long discovery condition were make the fewest transformation errors in Chapter 2, subjects in the short discovery condition were making the most (Figure 13b).

Perhaps the most important outcome of this research is the demonstration of a circumstance where discovery leads to superior learning. This positive outcome depends on two factors:

- (1) The search involved in discovery was sufficiently constrained that it was possible for students to remember what they had done after they finally discovery a successful transformation.. With such constraints in place and enough experience with the interface to guess good actions, students were able to discover transformations without becoming lost.
- (2) Successful discovery depends on attending to the semantics of the domain. The only thing that could guide discovery students as to what to type was the constraint that it had to preserve the values of the original data flow diagram. This forced them to attend to the semantics at each discovery point.

As noted, some participants in non-Discovery conditions seemed to be learning transformations that also respected the semantics of the diagrams. There is nothing magical about discovery learning 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 guided at every point except when they had to enter values, leaving them only to discover what to type in. This “semi-discovery” condition would have been more efficient. It has also been proposed (Aleven and 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. As argued in the introduction, the implications of a careful cognitive analysis of instructional interventions is going to be more nuanced than simply supporting blanket claims about the superiority or inferiority of discovery learning.

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 (2), 147-179.
- Anderson, J. R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science*, 29, 313-342.
- Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* New York, NY: Oxford University Press.
- Anderson, J. R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of mind. *Psychological Review*, 111, 1036-1060.
- Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4 (2), 167-207.
- 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 Instruction*, 7(4), 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.
- Cronbach, L. J., & Snow, R. E. (1977). *Aptitudes and instructional methods: A handbook for research on interactions*. New York: Irvington Publishers.
- Foerster, P.A. (1990). *Algebra I, 2nd Edition*. Menlo Park, CA: Addison-Wesley Publishing.

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.

Glaser, R. (1987) Further notes toward a psychology of instruction, In R. Glaser (Ed.) *Advances in instructional psychology Vol. III*, Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Hmelo-Silver, C. E., Duncan, R. G. and Chinn, C. A. (2007) Scaffolding and achievement in problem-based and inquiry learning: A response to Kirschner, Sweller, and Clark (2006). *Educational Psychologist* 42 , pp. 99-107.

Kalyuga, S., Ayres, P., Chandler, P., & Sweller, J. (2003). Expertise reversal effect. *Educational Psychologist*, 38, 23-31.

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., and 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 (2) 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, NY: Cambridge University Press.

Matsuda, N., Cohen, W. W., Sewall, J., Lacerda, G., & Koedinger, K. R. (2007). Predicting students performance with SimStudent that learns cognitive skills from observation. In R. Luckin, K. R. Koedinger & J. Greer (Eds.), *Proceedings of the international conference on Artificial Intelligence in Education* (pp. 467-476).

Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? *American Psychologist*, 59, 14 – 19.

Paas, F., Renkl, A., & Sweller, J. (2004). Cognitive load theory: Instructional implications of the interaction between information structures and cognitive architecture. *Instructional Science*, 32(1-2).

Reed, S. K. & Bolstad, C. A. (1991). Use of examples and procedures in problem solving. *Journal of Experimental Psychology: Learning, Memory, 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. (in press, a). Integrating intelligent software tutors with the mathematics classroom. In K. Heid & G. Blum (Eds.). *Research on Technology and the Teaching and Learning of Mathematics: Syntheses, Cases, and Perspectives*. Greenwich, CT: Information Age Publishing.

Rittle-Johnson, B. (2006). Promoting transfer: The effects of direct instruction and self-explanation. *Child Development*. 77(1), 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. and Paas, F. (2007) Problem-based learning is compatible with human cognitive architecture: Commentary on Kirschner, Sweller, and Clark (2006). *Educational Psychologist* **42** , pp. 91-97.

Sweller, J. (1994). "Cognitive Load Theory, learning difficulty, and instructional design". *Learning and Instruction* **4**: 295-312.

Taatgen, N.A., Huss, D. & Anderson, J.R. (2008). The Acquisition of Robust and Flexible Cognitive Skills. *Journal of Experimental Psychology: General*, 137, 548-565.

Table 1

Instruction for Section 2.6

	Verbal Direction	Direct Demonstration
Initial General Instructions	"One can collapse two boxes with + or - into a single box and preserve the value of the diagram. One can do the same thing with two boxes with * or /."	
State (a)	"Find two boxes with addition or subtraction and click them"	"Click-This" Arrows
State (b)	"Click the Button Labeled Combine"	"Click-This" Arrow
State (c)	"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."	"Type 7+3" Arrow
State (d)	"Find a box with two numbers and an operator and click it."	"Click-This" Arrow
State (e)	"Click the Button Labeled Evaluate".	"Click-This" Arrow
State (f)	"Click the little green box."	"Click-This" Arrow
	"Find the answer by evaluating the box above and enter it."	"Type 10" Arrow
State (g)	"Your answer is correct type the Next Problem button."	"Click-This" Arrow

Table 2

Distribution of Transformation Errors per Problem

Mean # Errors	Both	Verbal Direction	Direct Demonstration	Discovery
0.02 - 0.11	2	2	2	4
0.11 - 0.21	2	1	3	4
0.21 - 0.39	3	3	2	2
0.39 - 1.75	3	4	3	0

Figure Captions

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.

Figure 2. Illustration of the interface for the algebra tutor during the filling in of a transformation. (a) The linear representation; (b) The data-flow representation.

Figure 3. The data flow equivalents of (a) $5x + 4 = 39$ and (b) $(2x - 5x) + 13 + 9x = 67$.

Figure 4. A comparison of the performance of an ACT-R model with that of: children learning a linear form of algebra and adults learning the data-flow form. The dependent measure is time to perform a transformation and the independent variable is the position of the problem in the curriculum sequence. From Anderson (2007).

Figure 4. The steps in the solution of a combine problem from Section 2.6. The two lines reflect the two transformations. The first line starts with the problem (part a), then a part of the graph is selected and highlighted in red (part b), then the combine operation is selected resulting in part d, and the parts are filled in resulting in part d. The second line starts with part d from the previous line, then a part of the graph is selected for evaluation (part e), the evaluation operation is selected resulting in part f, and the value is filled in resulting in part g.

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)

Figure 6. (a) Mean time per transformation and (b) mean number of transformations 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)

Figure 7. Mean number of (a) operator errors and (b) transformation errors 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)

Figure 8. Transformation errors on the 20 problems in Section 2.6. The data is averaged from 30 participants and the model predictions are based on 1000 simulation runs. (Experiment 1).

This measure is not bounded above by 1 because participants can make multiple errors.

Figure 9. Search of discovery students through the problem space for the first problem of Section 1.7: (a) The states in performing the first invert transformation; (b) The states in performing the second evaluate transformation. The arcs between states are labeled with the proportion of times that participants took that arc in the Discovery condition of Experiment 1 (long) and the Discovery conditions of Experiment 2 (short).

Figure 10. 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 contributing to the last two chapters is given above the data point for those chapters. (Experiment 2)

Figure 11. Time to solve problems in the first two chapters as a function of instructional condition and whether the problems were the first instructional problems in a section or later practice problems. (Experiments 1 & 2 combined)

Figure 12. (a) Mean time per transformation and (b) mean number of transformations 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 & 2 combined)

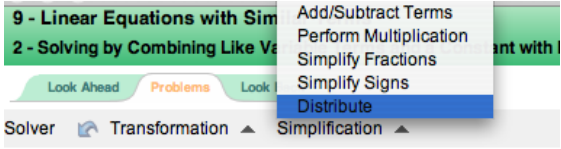
Figure 13. (a) Mean number of operator errors and (b) transformation errors as a function of instructional condition, chapter, and whether the problems were the first instructional problems in a section or later practice problems. (Experiments 1 & 2 combined)

Figure 14. Mean number actions more than minimum on the first problem in section as a function of instructional condition, section, and transformation. (Experiments 1 & 2 combined)

Figure 15. Performance of the Long and Short participants and model predictions on the problems in the short curriculum in the first four sections: (a) Direct Demonstration; (b) Discovery

Figure 1

(a)



Solve for y

$$8y + (-6y) + 9 = 10$$

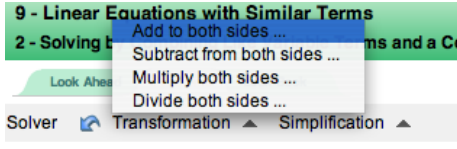
(b)

$8y - 6y + 9$

y	constant terms
$(\text{ }) + (\text{ })$	()
$(\text{ }) + (\text{ })$	()

Result

(c)



Solve for y

$$8y + (-6y) + 9 = 10$$

$$2y + 9 = 10$$

(d)

$$2y + 9 = 10$$

$$2y + 9 - \text{ } = 10 - \text{ }$$

Figure 2

(a)

$$((3 * x) + (4 * x)) + 8 = 22$$

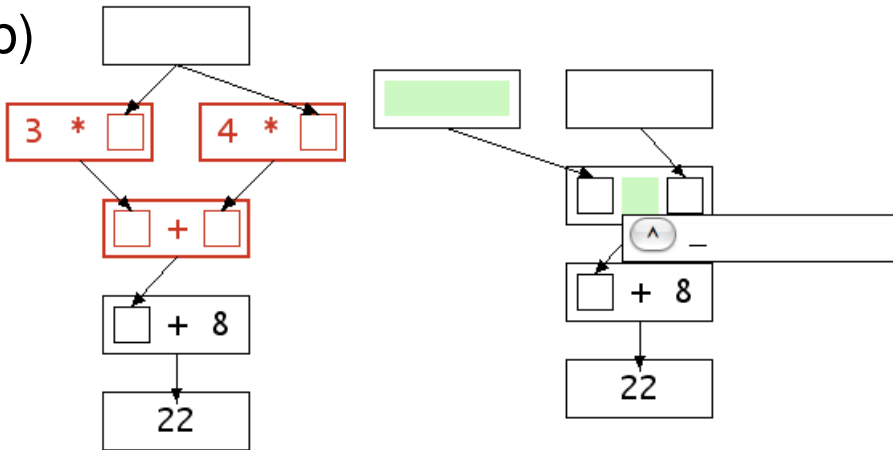
$$((\text{ }) \text{ } x) + 8 = 22$$

\wedge -

- Evaluate
- Unwind
- Collect
- Distribute
- Canonicalize
- Undo Minus
- Subtract

- <-
- >
- First Mistake
- Hint
- Next Problem

(b)

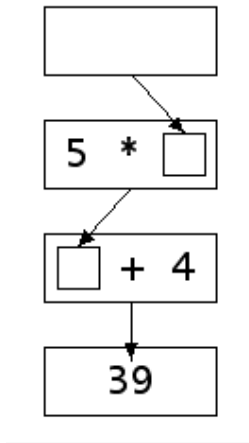


- Evaluate
- Invert
- Combine
- Reorder
- Canonicalize
- Undo Minus
- Subtract

- <-
- >
- First Mistake
- Hint
- Next Problem

Figure 3

(a)



(b)

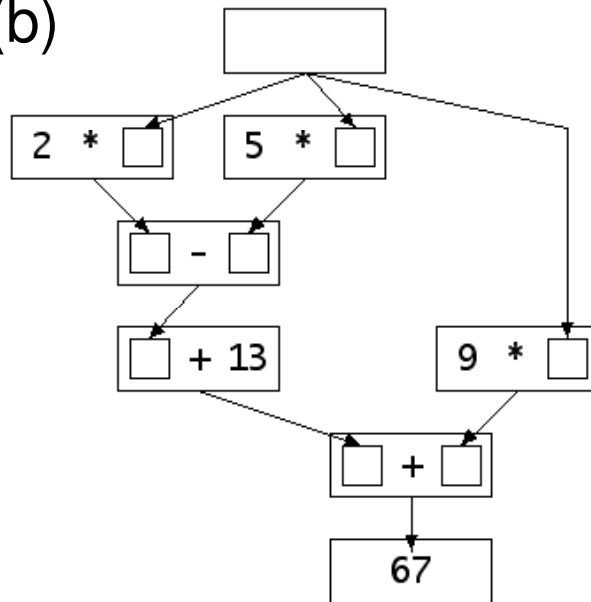


Figure 4

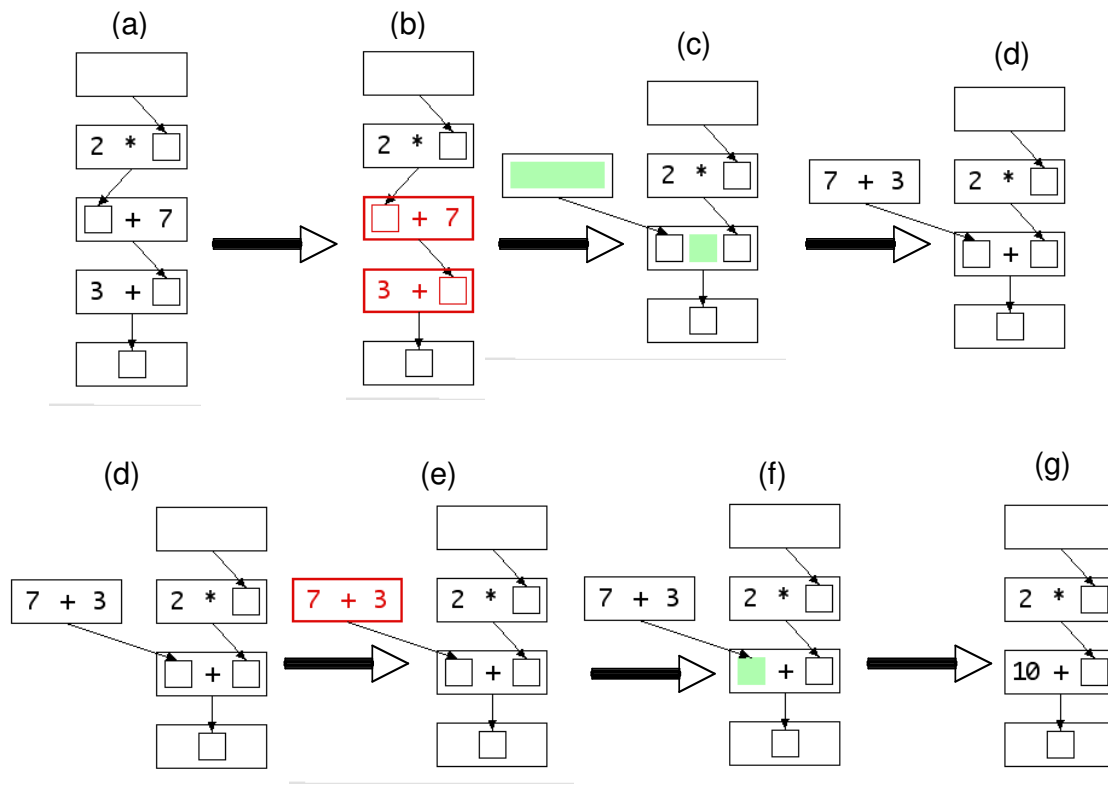


Figure 5

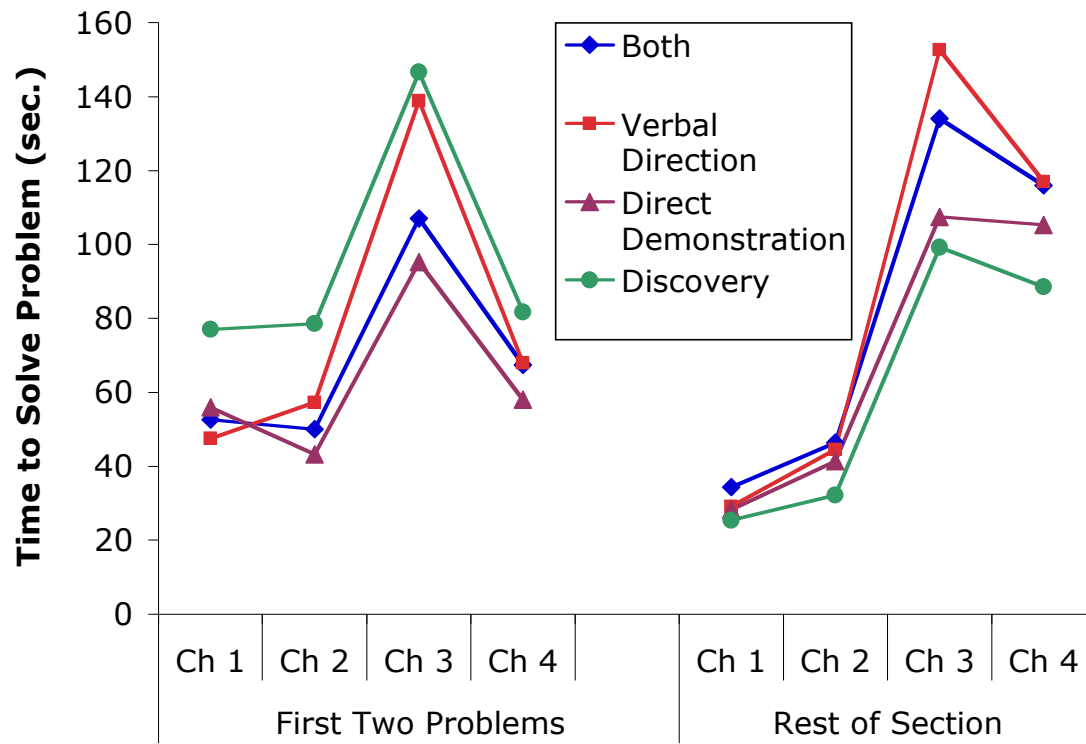


Figure 6a

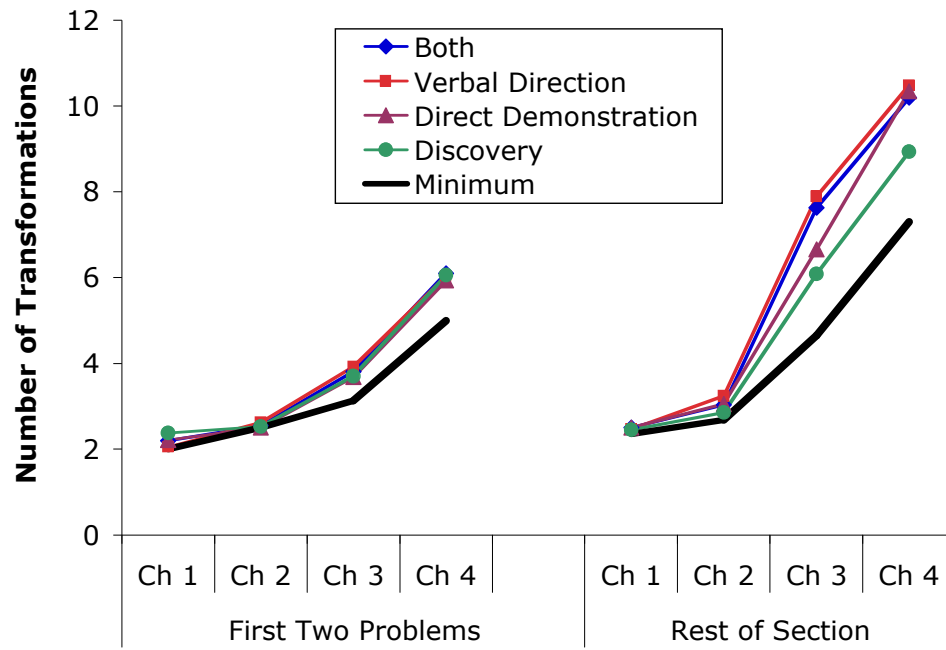


Figure 6b

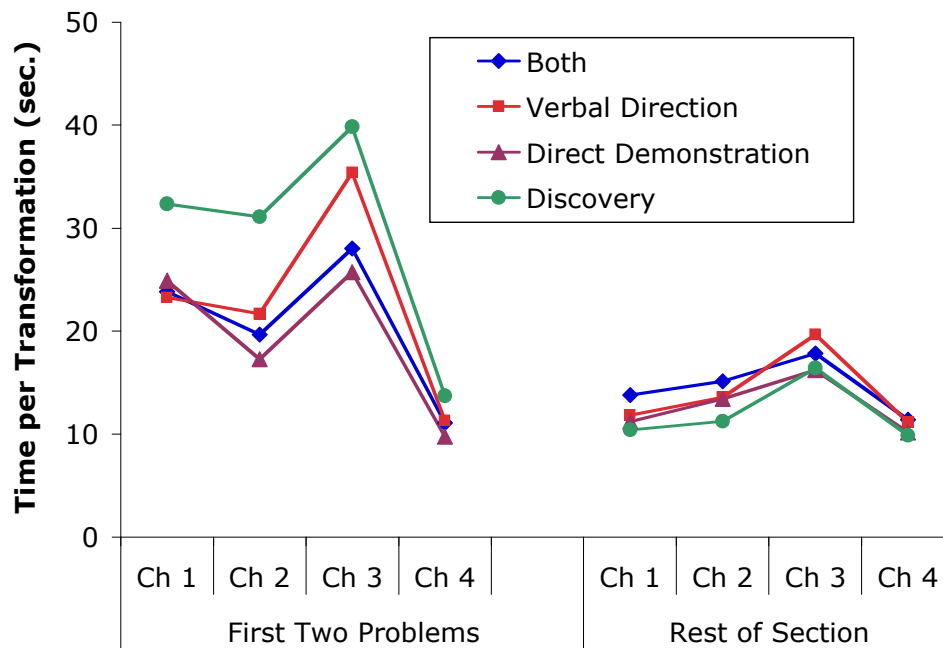


Figure 7a

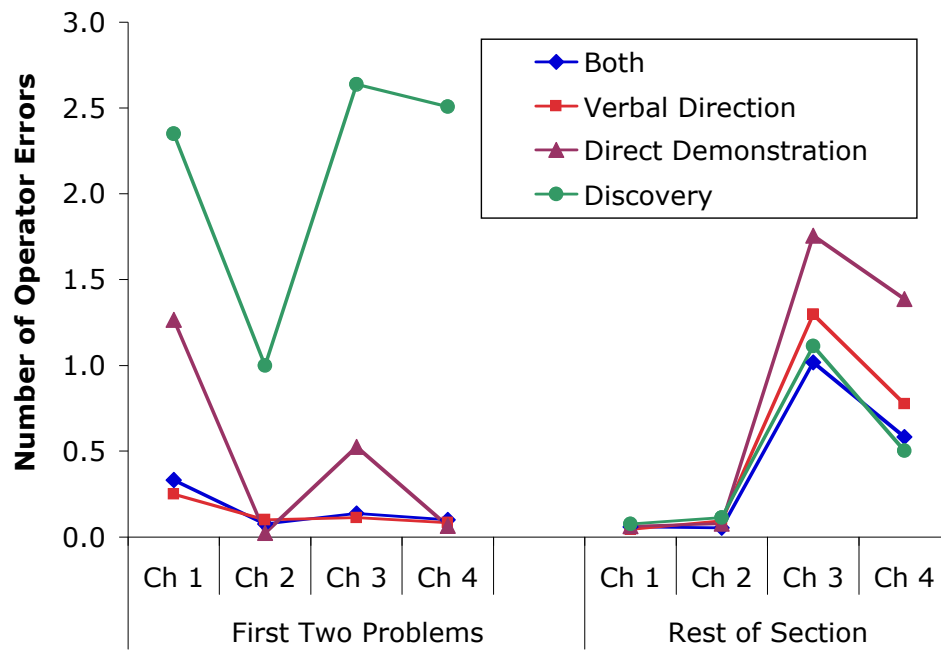


Figure 7b

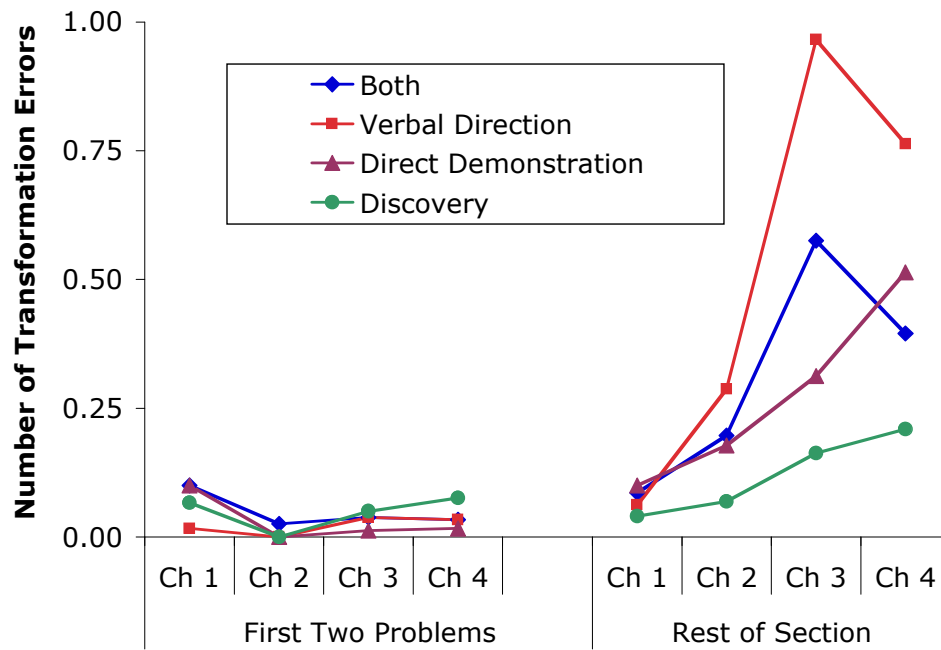


Figure 8

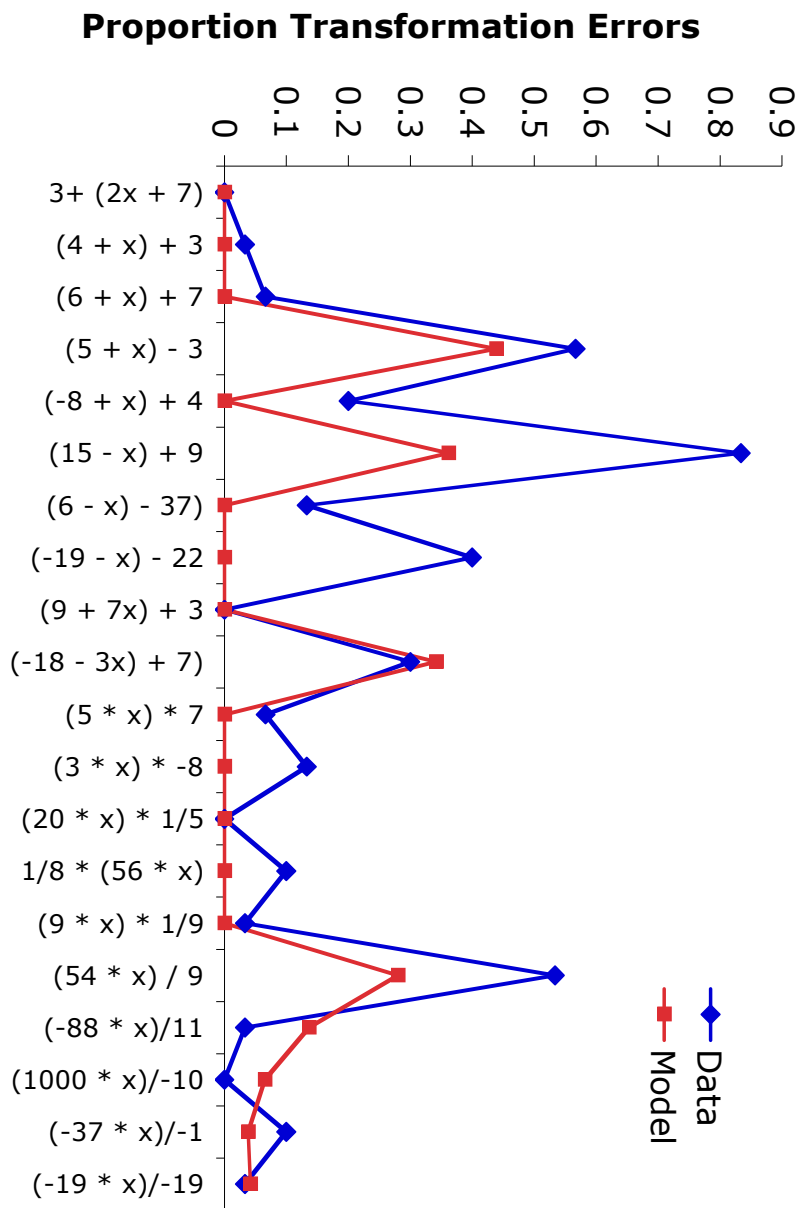


Figure 9a

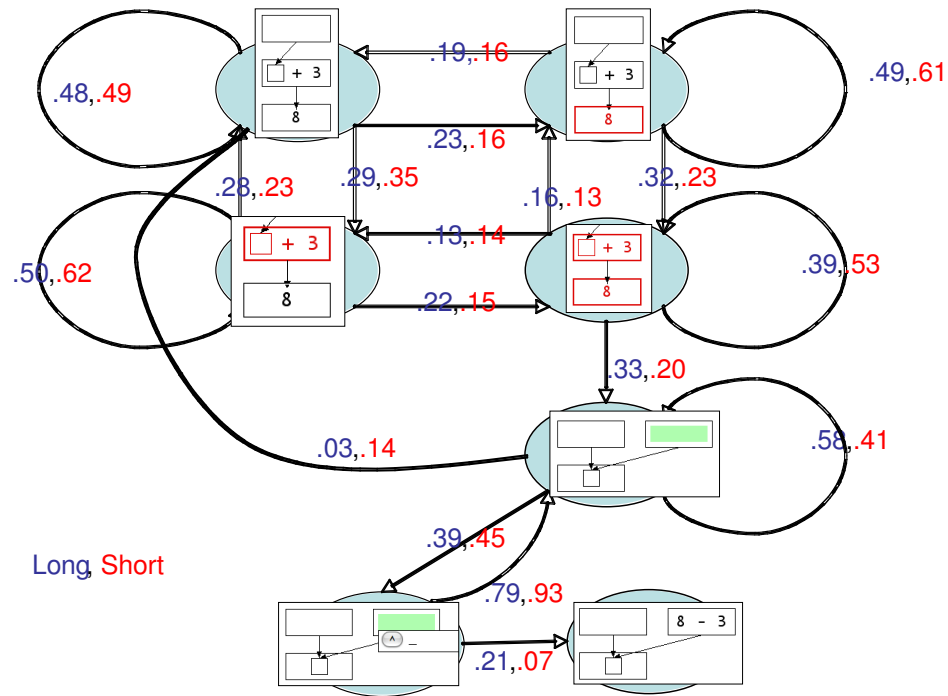


Figure 9b

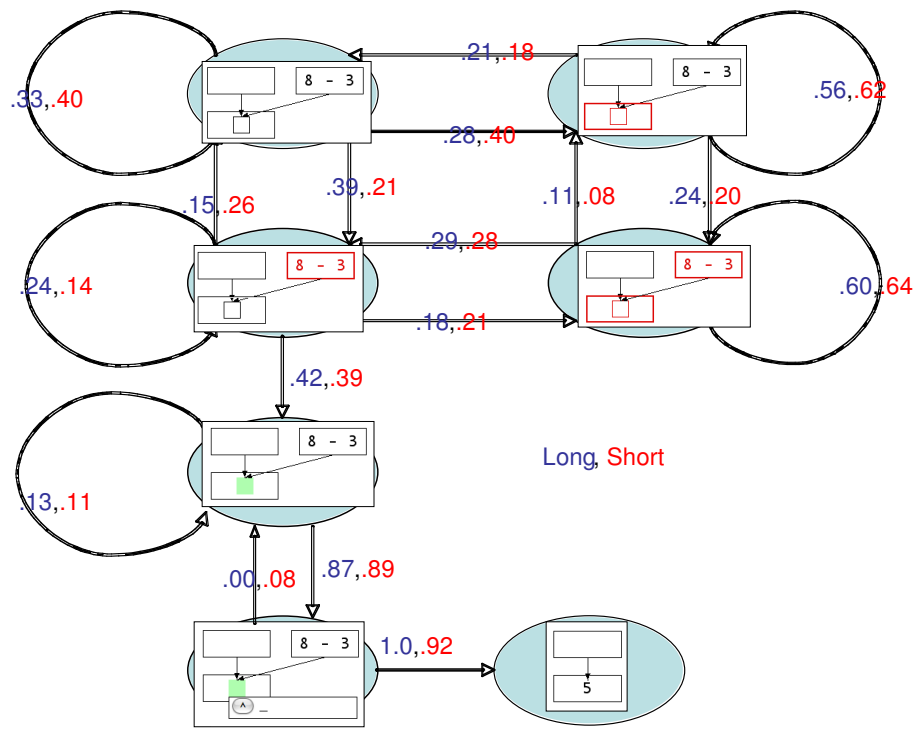


Figure 10

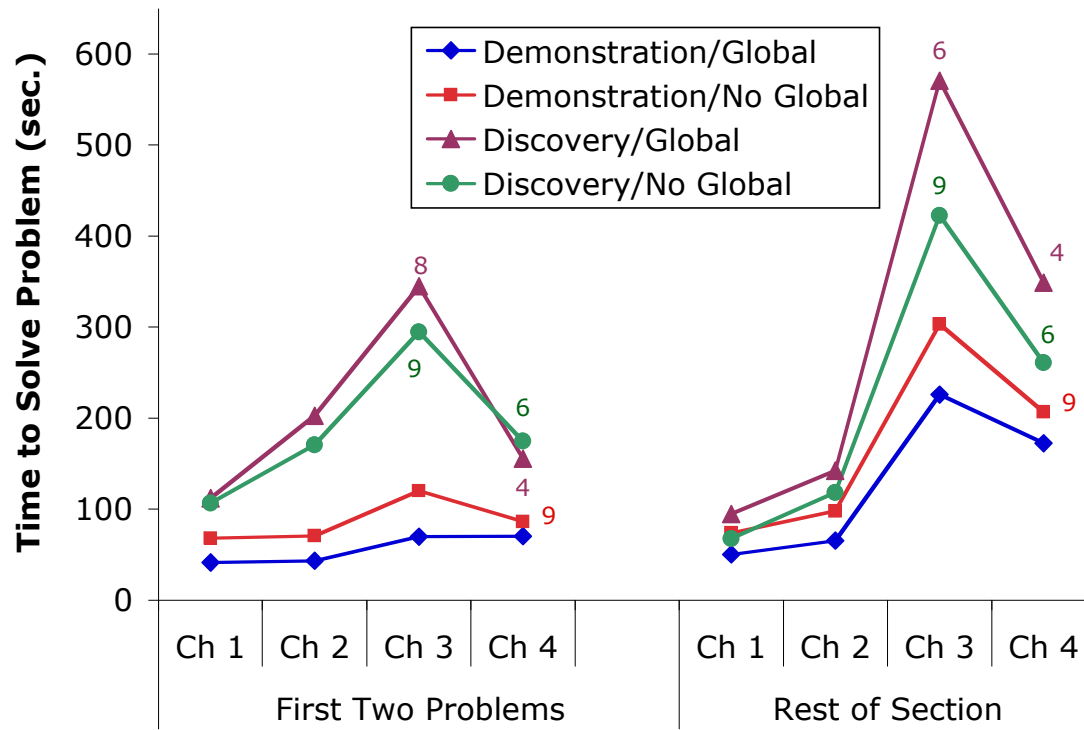


Figure 11

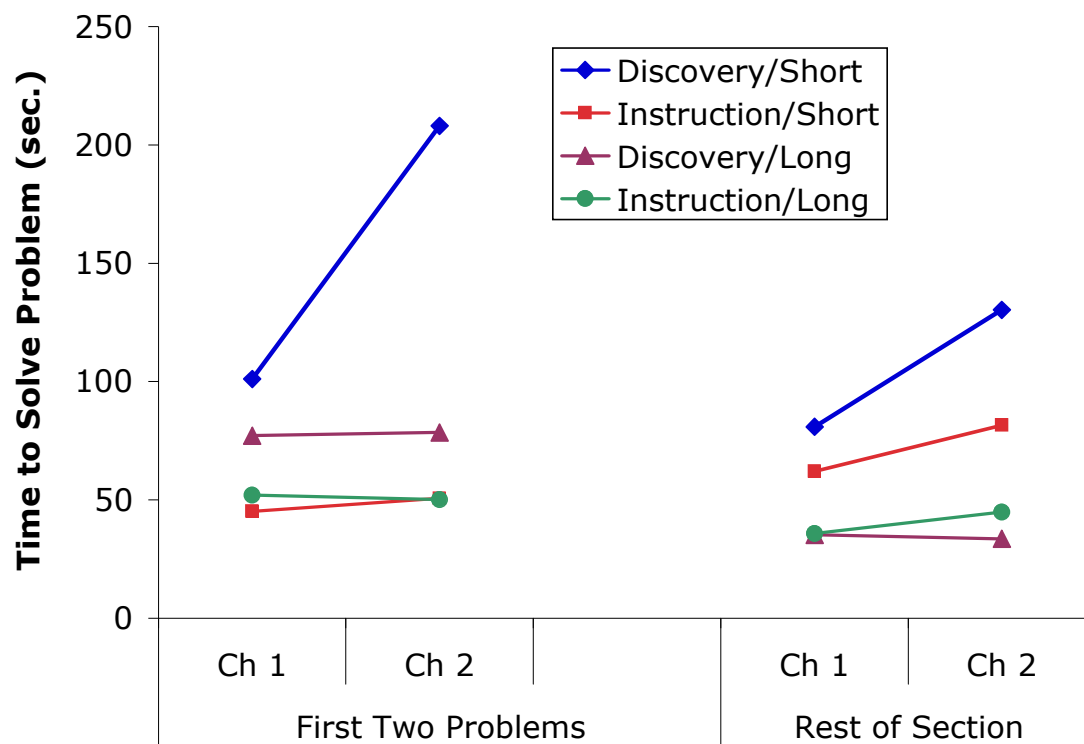


Figure 12a

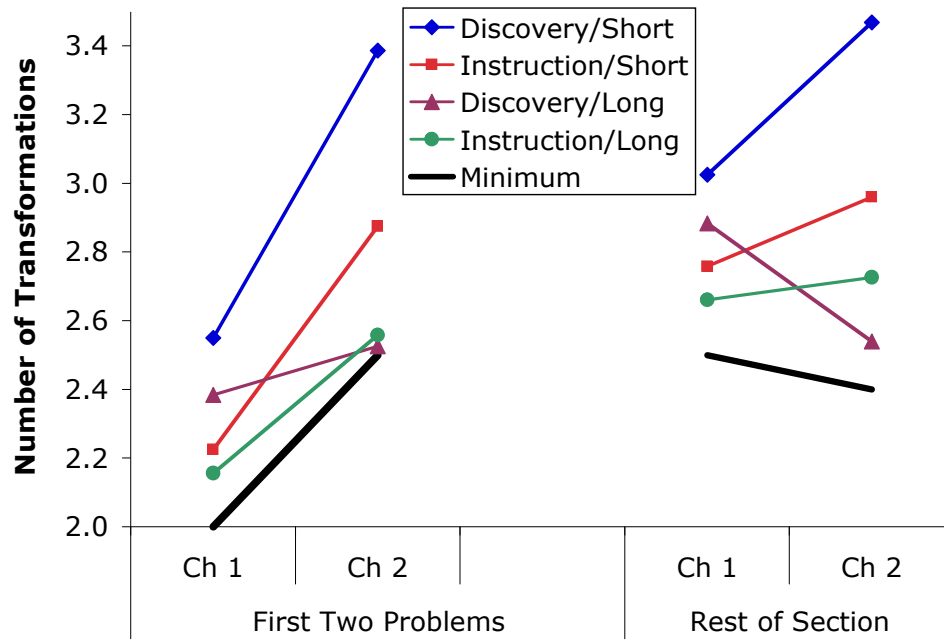


Figure 12b

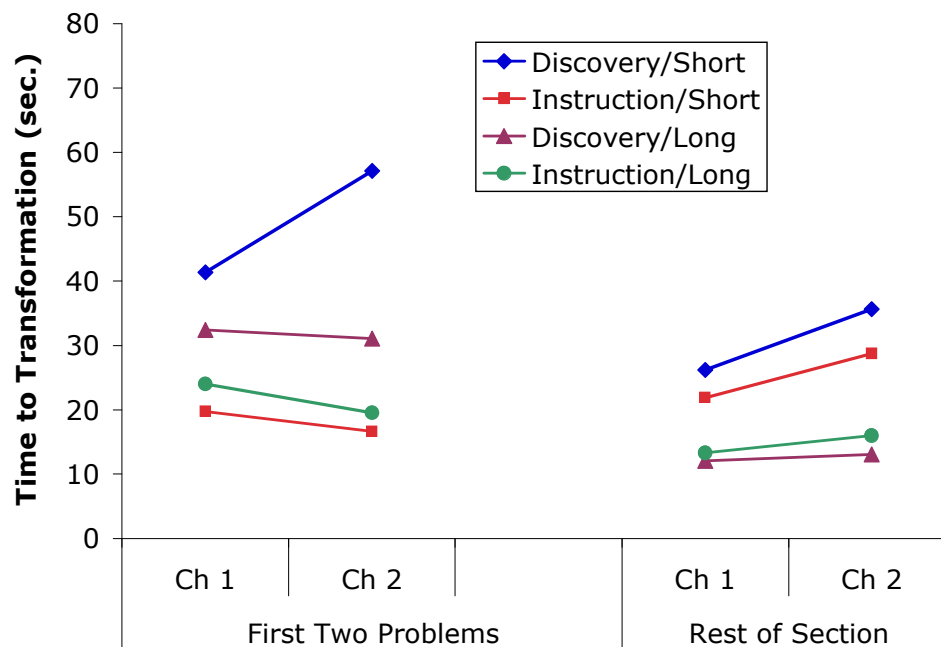


Figure 13a

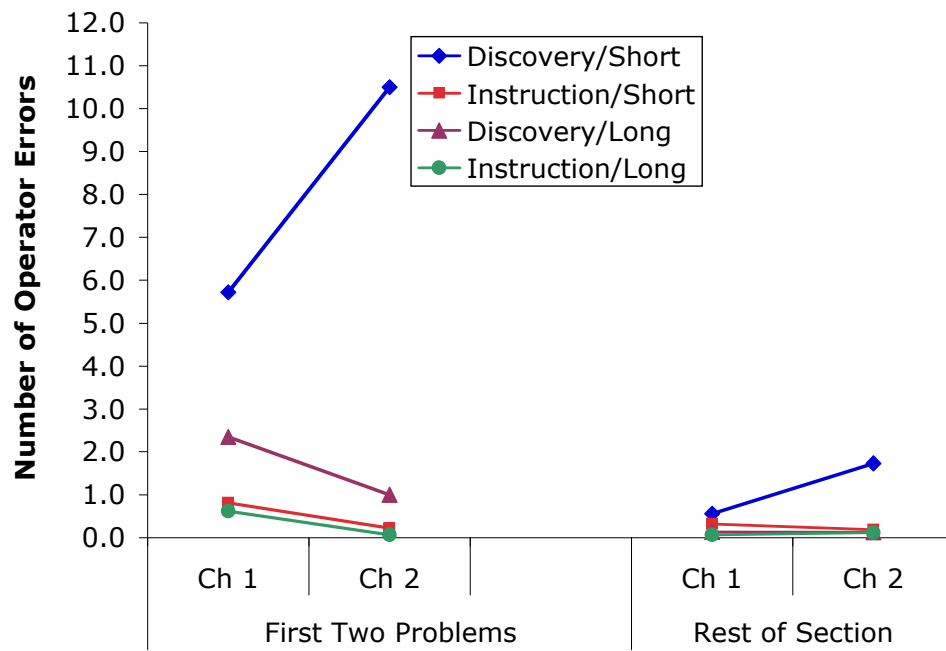


Figure 13b

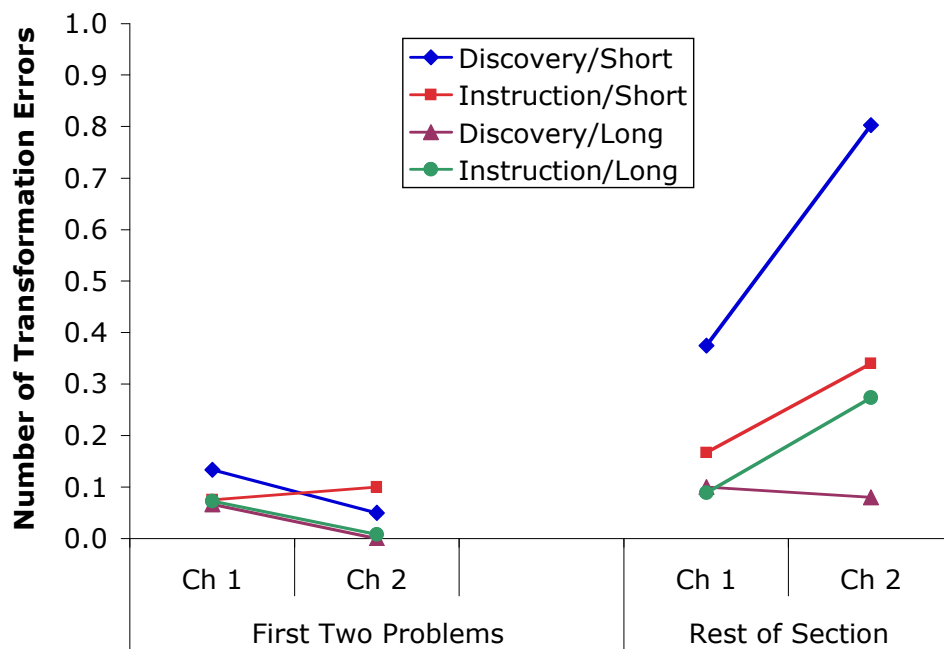


Figure 14

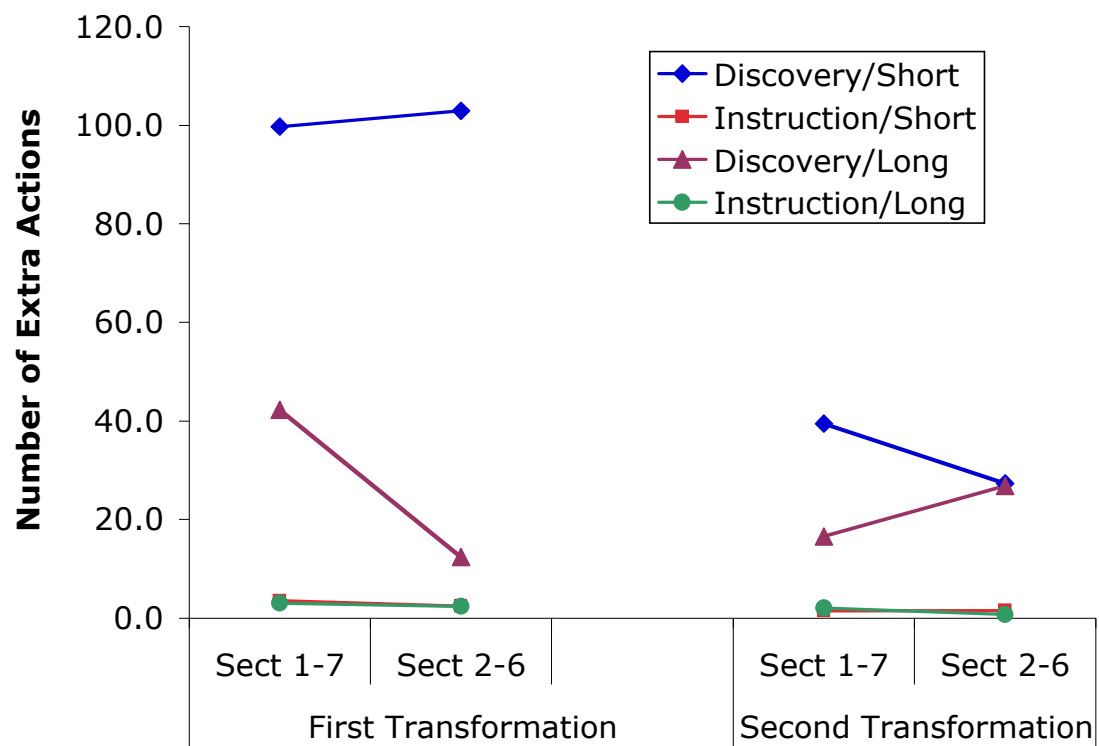


Figure 15a

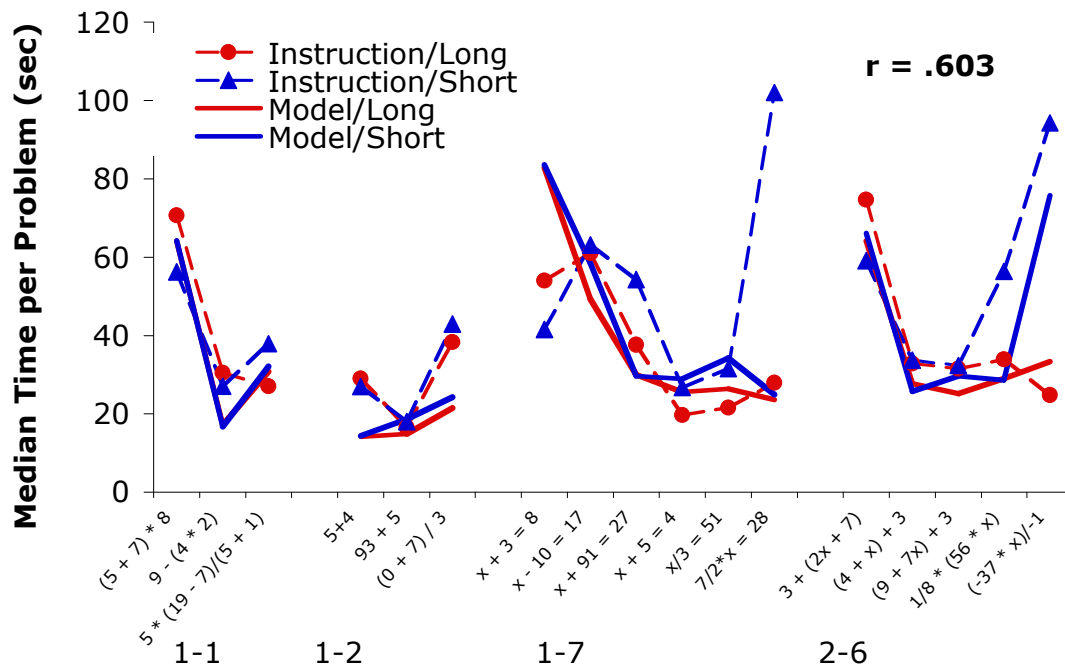


Figure 15b

