Title: When does provision of instruction promote learning?

Author names: Hee Seung Lee, Abraham Anderson, Shawn Betts, John R. Anderson

Publication date: July, 2011

Meetings held at: Boston, MA.

Citation:

Lee, H. S., Anderson, A., Betts, S., & Anderson, J. R. (2011). When does provision of instruction promote learning? In L. Carlson, C. Hoelscher, & T. Shipley (Eds.), Proceedings of the 33rd Annual Conference of the Cognitive Science Society (pp. 3518-3523). Austin, TX: Cognitive Science Society.

When does Provision of Instruction Promote Learning?

Hee Seung Lee, Abraham Anderson, Shawn Betts, and John R. Anderson
{heeseung, lobo, sabetts, ja0s} @andrew.cmu.edu
Department of Psychology, Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Contradictory evidence has been reported on the effects of discovery learning approach and the role of instructional explanations. By manipulating the presence of instruction (verbal explanation) and transparency of problem structures, we investigated how effects of instructional explanations differed depending on the transparency of problem structure. We developed an auxiliary representational task that made certain aspects of the problem structure more transparent. Instruction proved irrelevant to those aspects of the problem made transparent by the representation but facilitated learning of those aspects that were left obscure. We suggest that the critical role of instruction is not specifying the steps in solving a problem, but rather making salient the features that are critical to student's ability to infer the steps from examples.

Keywords: discovery learning, instruction, explanation, transparency of problem structure, representation.

Introduction

One of the central controversies in education is how much instructional guidance needs to be provided in a learning environment. Even if an optimal amount of instruction is identified in one learning environment, the effect might differ based on the use of other instructional factors. This study investigated how effects of instructional explanations differed in different instructional conditions.

Debate over discovery learning and direct instruction

Is it better to give students instructions about how to solve problems or is it better to allow students an opportunity to discover the knowledge for themselves? This question led to a long debate over effects of discovery learning versus direct instruction approach. Discovery learning approach is based on a constructivist theory of learning (Piaget, 1970) and it emphasizes learners' active engagement in constructing their own knowledge. Learners are believed to be able to generate their own examples and explore them for learning.

On the other hand, some researchers argue that the discovery learning approach has continuing advocates but without sufficient evidence (e.g., Mayer, 2004) and instructional guidance is critical to successful learning. In discovery learning students in effect have to generate their own worked examples by discovering solutions and making sense of their own solution steps. This can be at a disadvantage to providing instruction in that it can be costly both of time and working memory (e.g, Sweller, 1988) to generate the examples. Also, it can be at a disadvantage to providing instruction in that the structure of the solution is not explained. This often increases floundering thus students may never be able to discover what they are to learn (Ausubel, 1964).

Both positive and negative effects of discovery learning have been reported in many different domains. Several studies demonstrated when students invented their own solution procedures, they showed better understanding of the domain than those who simply followed instructed solution procedures (e.g., Hierbert & Wearne, 1996; Kamii & Dominick, 1998). However, Rittle-Johnson (2006) reported an opposite finding. She found students who were directly taught a correct procedure showed better procedural transfer than those who were told to think of a new way to solve the problem without instruction.

In the science domain, Klahr and Nigam (2004) showed direct instruction on control of variable strategy (CVS) led to better learning outcome and in turn subsequent transfer. However, Dean and Kuhn (2006) demonstrated direct instruction was neither necessary nor sufficient and instead practice appeared more important for enhancing students' scientific inquiry skill. The benefits of direct instruction quickly disappeared without a long term engagement. Consistent with this finding, Brunstein, Betts, and Anderson (2009) also reported that discovery learning was more effective than direct instruction only when combined with high levels of practice in the domain of Algebra learning.

Worked examples and instructional explanations

In contrast to contradictory evidence with respect to direct instruction, there is strong evidence that learning is facilitated by the provision of worked examples (e.g., Carroll, 1994; Tuovinen & Sweller, 1999). Worked examples are believed to help students focus on relevant solution steps by reducing search activity that is irrelevant for problem schema acquisition. Provision of instruction and worked examples can be seen as orthogonal factors with discovery learning being the situation where the student has to generate solutions without the benefit of either examples or instruction. Even when worked examples are provided, if underlying solution steps are not explained and/or relevant features are not appropriately highlighted, students are left to generate own explanations to understand worked examples and discover relevant features for their learning. This process is not always successful. Sstudents often show *illusion of understanding* and fail to solve comprehension problems without instructional explanations (Renkl, 2002).

However, instructional explanations can be also

detrimental by preventing learners from actively making sense of learning materials. For example, Schworm and Renkl (2006) found that provision of instructional explanations reduced learner's self-explanation activities and in turn learning outcomes. Also, when instructional explanation was not presented in an integrated format, it can impair learning by increasing cognitive load (*split-attention effect*, Ward & Sweller, 1990).

This suggests that provision of instructional explanation does not always guarantee a positive learning outcome (for review, see Wittwer & Renkl, 2008). For instruction to be helpful, it is not enough to simply specify solution steps. More importantly, it should reveal any hidden structure in a problem by making salient relevant features in the examples. When instruction fails to perform this function, students will be left in a situation analogous to discovery learning and could well flounder. In this case, provision of instruction will have little effect or even a detrimental effect. At the same time, if this function can be performed by other scaffolding means, instruction may be irrelevant.

We hypothesize that with appropriate scaffolding discovery learning becomes, in effect, a worked example condition where students provide their own worked examples and the scaffolding makes transparent critical features of the examples. To test this hypothesis, we developed an auxiliary representation task that made certain aspects of problem structure more transparent to learners and investigated how effects of instruction differed depending on the transparency of problem structure. We expect when the representation task reveals the hidden problem structure, provision of instruction will have little effect. On the other hand, when obscure relationships are not revealed, instruction will play a critical role and facilitate learning of those aspects that were left obscure.

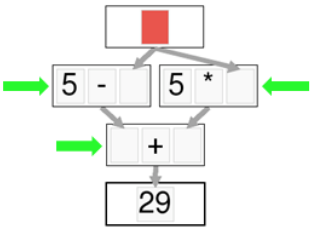
Experiment

We adapted the computer-based tutoring system used by Brunstein et al. (2009). In this system, isomorphs of algebraic expressions are represented as data-flow diagrams with multiple boxes and arrows. The task involved selecting a part of boxes and filling in values into an empty portion of the box. This system allowed us to study college students learning anew the equivalent of algebra.

Figure 1 illustrates stages of an example problem used in the tutor. An unknown number flows from the top box into the boxes below, the arithmetic operations are performed, and the final result is the 29 at the bottom. The student's task is to determine what the unknown number is. The diagram in Figure 1(a) is equivalent to the algebraic expression, $(5 - x) + (5 * x) = 29$ while the diagram in Figure 1(c) is equivalent to $5 + 4x = 29$.

To solve these problems students can click on empty tiles in the boxes and enter values. In the case of problem 1(c) the unknown value can be simply determined by "propagating" the number up from the bottom, unwinding the arithmetic operations – placing 24 in the empty tile above the 29 (equivalent to rewriting $5 + 4x = 29$ as $4x = 24$), then 6 in the tile above it (equivalent to rewriting this as $x = 6$), and finally 6 in the top unknown box. However, in cases like 1(a), where two paths converge in a result, this simple procedure is not possible and at this point students must transform the graph in Figure 1(a) into the form in Figure 1(c). This step, called **linearization**, is the major conceptual hurdle in this artificial curriculum.

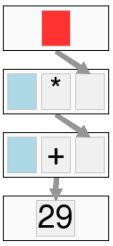
(a) Selection task: Equivalent of $(5 - x) + (5 * x) = 29$



Instruction: "Select the rectangular box with two little, empty boxes along with the rectangular box connected to it because they are in a loop connecting to a common rectangular box above."

Discovery: "Select three rectangular boxes."
(Arrows were not provided.)

(b) Execution task



Instruction: "The answer to this box is 4 because -1 + 5 is 4"; "The answer to this box is 5."
(A corresponding arrow appeared for each statement.)

Discovery: "Click a box and enter a number."
(Arrows were not provided.)

(c) Result of linearization: Equivalent of $5 + 4x = 29$

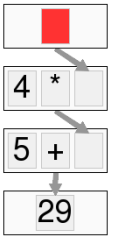


Figure 1: Stages of an example problem. Depending on the instructional conditions, different hints were provided and examples are shown below the figures for direct instruction and discovery condition.

The linearization always involved two major subtasks: *selection* and *execution*. First, in the *selection* task, students have to select the appropriate rectangular boxes to "linearize". In Figure 1(a), the correct boxes are indicated by arrows. After selecting these three boxes, clicking a linearize button transforms the data-flow diagram into the intermediate state in Figure 1(b). The challenge in this selection subtask is to determine what boxes have to be selected. Diagrams vary in structure and Figure 1 is just one possibility. Participants have to find a loop that connects from the top unknown value to the bottom box with empty values and select all of the boxes in the loop except the top box of the loop. In the example in Figure 1(a), the loop starts from the top unknown box where two arrows diverge. Then the loop continues all the way down to the bottom box where the right and left branches finally rejoin. This convergence enables summation of left and right branches.

Second, in the *execution* subtask, participants have to find what values to fill in on the transformed tree. Essentially they need to find the coefficient that multiplies the top unknown value and the constant. In this example, the selected boxes for linearization are simplified into the equivalent of $(4 * x) + 5$. The coefficient 4 ($-1x + 5x = 4x$) goes to the box next to "*" operator and the constant 5 goes into the box next to "+" operator. After filling in these two values, the diagram has been transformed into the linear form in Figure 1(c) and now users can apply the simple propagating strategy. The general characterization of the selection and execution rules worked in the same way across all problems.

Above we have tried to explain the solution of these problems by reference to algebraic transformations. Students find propagating numbers easy and intuitive and do not seem to need any reference to algebra. One student described this as "like doing sudoku". On the other hand, students find it hard to master linearization and reference to algebra seems to make this easier to explain and understand. By changing a representation format, from visual representation of a data flow diagram into a symbolic representation of an algebraic equation, the hidden structure of the problem is revealed and the required computations become a lot easier.

The current study used this relation between data-flow structure and algebraic expression to manipulate the transparency of problem structure. We did this with an additional representation task. By having students re-represent certain parts of the diagram into an algebraic form or an equivalent data-flow tree, the problem structure could become apparent or remain ambiguous. In particular, we expected that seeing algebraic equivalents would make the execution part of the linearization more transparent by showing the relationship between simplifying an algebraic expression and determining the values to be entered into the linearized diagram. It was less clear that showing students this representation would help them select the boxes for linearization because there is not a simple relationship between this selection task and simplifying algebraic expressions. So, our specific hypothesis is that providing algebraic expressions will make instruction less important for the execution portion of linearization.

Method

Participants Fifty undergraduate students (29 male and 21 female, M = 21 years, SD = 2.2 years) at the Carnegie Mellon University participated in the study. Participants received $10/hour plus performance based bonus.

Design, materials, and procedures

The study consisted of two sessions, learning session and transfer session. Each session lasted 2 hours and the transfer session occurred 2 days after the first session. Transfer task was identical across all experimental conditions and experimental manipulations occurred only during the learning session. The learning session had three different problem sections – propagate, easier linearize, and harder linearize problems. In the propagate problems, participants learned to propagate numbers up or down. These problems did not involve the use of linearize function and could be easily solved with simple arithmetic calculations. The easier and harder linearize problems involved the use of linearization and these two sections differed in terms of size of a loop. Although both sections required only single linearization, the latter had a larger loop. That is, participants had to select more boxes and this in turn made computation more complicated. Each subsection had one single demo problem (i.e., worked-example) and 30 problems. Among the 30 problems, for the only first problem hints were always provided for each solution step and for the rest 29 problems hints were provided on request.

A 2 x 2 between-subjects study was designed by crossing provision of instructional explanation (direct instruction vs. discovery) and representation task (algebraic expression vs. data-flow tree). First, there were two instructional conditions, direct instruction versus discovery. In the direct instruction condition, participants were given a worked-example with verbal explanations about each solution step. By clicking the forward button, participants observed how a problem was solved in a sequential way. Text explanations appeared at the bottom of the page and provided general characterization of an action taken in each step (e.g., why certain boxes need to be selected, why a certain value is filled). This instruction was also available for later problems by clicking a hint button.

In the discovery condition, verbal explanation was not provided although the identical step-by-step worked-example was provided in the beginning of the learning session. Because no characterizations of actions were provided, participants were left to discover transformations by themselves. In this condition, participants could also use a hint button. However, the provided texts did not explain underlying principles of either selection or execution task. Instead, the hints simply provided instruction on interface issue such as "click a box and enter a number". Examples of hints provided in direct instruction and discovery conditions are shown in Figure 1.

Across all conditions and all sessions, error messages appeared whenever participants made a mistake. In the direct instruction condition, participants could then find the correct answer by clicking the hint button. In the discovery condition, however, the hints only provided interface information and thus participants had to figure out the answer for themselves. The discovery learning condition was not pure discovery in that a demo problem was provided in the beginning of the learning session, hints on interface of the tutor were available on request, and immediate feedback was given via an error message. There never were any explanations and these features were present to minimize the floundering in discovering a solution. The distinction between two instructional types should be understood as a continuum and the most critical difference between these two conditions was presence or absence of verbal explanations.

Second, the type of representation task was manipulated to manipulate the transparency of the problem structure. Students were asked to re-represent certain parts of the data-flow diagrams either in an algebraic expression or in a data-flow tree. In the expression condition, participants were asked to fill in algebraic expressions by copying numbers and operations from a diagram to an expression. Unknown values were already filled in as an X in the task. This task was to help students understand the problem structure by revealing the relation between the diagram and algebraic expression. In the tree condition, participants were asked to simply copy numbers and operators from a diagram to a tree. Figure 2 shows an example of each representation task. Students were asked to fill in empty tiles either in the algebraic expression (a) or in the tree (b). This representation task appeared whenever participants made an execution error (i.e., filling in wrong values in the linearization). In the demo problem or worked-example, the representation task was automatically performed by the system right after the diagram was transformed, between selection and execution task. Only in the direct instruction condition, the completed representation was used as part of instruction. Verbal explanations were given by highlighting relevant parts of the expression or tree. In the discovery condition, neither verbal explanation nor highlighting was provided.

For each correct solution step, participants could earn money. To prevent people from relying on hints without efforts to solve problems, whenever people asked for hints, 4 cents were deducted from their earned money. Also, when a linearization error was made, 2 cents were deducted. Until all parts were solved correctly, participants could not move on to the next problem. Time-on-task was controlled as 2 hours for learning session therefore each participant solved different numbers of problems. Performances varied largely across participants and only some of the participants could finish all three problem sections on the first day of the study. Regardless of whether students could finish all or part of the sessions on the first day, they were given identical transfer test on the second part of the study.

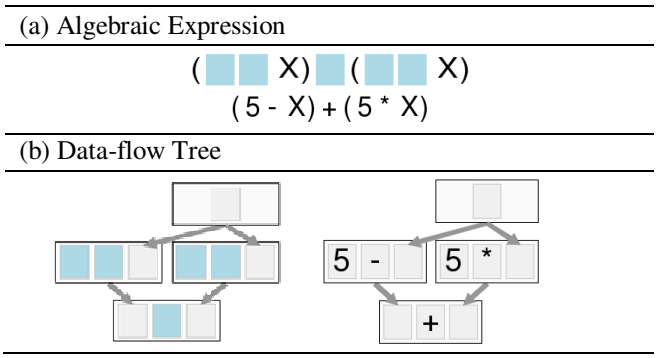


Figure 2: Examples of the representation task. Students were asked to fill in the empty portion of either an algebraic expression (a) or a tree (b) with numbers and operators. Given are the forms of the task before and after completion.

On the second day session, identical transfer problems were given across all conditions. No worked examples and hints were provided. Like the first day session, an error message appeared when an error was made. There was only one problem type, multiple linearize problems, and this required multiple uses of linearization to solve the problem. On the first day, all problems involved only one single use of linearization. This made it a good bit more difficult to decide what boxes to select and what numbers to combine for a particular linearization. There were a total of 40 problems and participants solved problems for 2 hours.

At the end of the study, participants were given a questionnaire for demographic information and background in mathematics learning. Also, students were asked to verbally explain difficulties they had during the task and rules or strategies they used for selection and execution task.

Results

Out of 50, three participants felt lost and wanted to give up in the middle of the study. These participants were one from instruction-tree, one from discovery-expression, and one from discovery-tree and were excluded from the data analysis. Performance during the learning session varied in terms of problem solving speed, number of solved problems and use of hints. Especially, students in the discovery-tree condition solved the fewest problems. Therefore, we focused on the performance of transfer test. In this way, we can understand what 2 hours of practice under different instructional conditions mean for performance in a constant condition-transfer task. Please note that all participants were exposed to identical transfer conditions (no demo problem and no hints) regardless of the instructional conditions.

On the transfer test, there were again large individual differences. Some participants could finish all 40 problems within 2 hours, but some could solve just less than 10 problems. Like the pattern observed in the learning session, the discovery-tree condition had particular difficulty and could solve about 7 problems less than the other conditions. Due to the observed individual differences in terms of the number of solved problems, only the first 12 problems out

of 40 problems were chosen for analysis and the data from two participants who solved less than 12 problems were excluded from the analysis. One was removed from instruction-expression and the other was removed from discovery-expression condition. As a result, data from 45 participants were analyzed: 12 from instruction-expression, 11 from instruction-tree, 10 from discovery-expression, and 12 from discovery-tree condition.

Most of the errors participants made were from one of two categories. First type of the error was *selection error* and this error occurred when participants selected wrong combination of boxes for linearization. The second type of error was *execution errors*. Execution errors occurred when students filled in wrong values into a box. This could occur when students did not understand the rules to fill in boxes after linearization or made a computation error. Analyses of covariance were performed using SAT math score reported by participants. To avoid undue effects of extreme values, numbers of selection errors and execution errors greater than 20 were recoded as 20. Although the data transformation did not change the overall pattern of the reported results, the recoded values tended to be from the discovery-expression condition.

Table 1(a) shows adjusted mean number of selection errors (SE in parentheses) on the first 12 problems of the transfer test. Regardless of the type of representation task, students who received direct instruction made significantly fewer selection errors than those from discovery condition, $F(1, 40) = 8.14$, $p = .007$. There was no overall effect of representation, $F(1, 40) = 0.92$. It is clear that the expression representation did not help the instruction subjects as the instruction-expression condition has slightly more errors than instruction-tree. On the other hand, there was some advantage in the discovery condition for expression representation, but the instruction-by-representation interaction was not significant, $F(1, 40) = 1.58$.

Execution errors in Table 1(b) show an opposite pattern from the one observed in selection errors. Regardless of instruction type, students who represented diagram in an algebraic expression made significantly fewer execution errors than those who did in a tree, $F(1, 40) = 7.85$, $p = .008$. There was no overall effect of instruction $F(1, 40) = 0.31$. For this measure, it is clear that instruction offered no further benefit to participants who saw the expression representation since the instruction-expression participants made more errors than the discovery-expression participants. On the other hand, participants who had the tree representation seem somewhat helped by instruction, but again the instruction-by-representation interaction was not significant, $F(1, 40) = 2.02$.

The expression representation task was intended to make apparent how to determine the values from the pre-linearized diagram to place in the linearized diagram. It seems that providing such a task eliminates the need for instruction. With the algebraic expression in hand, participants were able to determine from the examples what the rules were. On the other hand, the algebraic expression provided little insight into how to select the boxes for linearization and here instruction was critical.

Table 1: Adjusted mean number of selection errors and execution errors (SE in parentheses) by instruction type and representation type.

(a) Selection	Instruction	Discovery	Average
Expression	3.06 (0.54)	3.96 (0.60)	3.51
Tree	2.90 (0.57)	5.21 (0.54)	4.05
Average	2.98	4.58	
(b) Execution	**Instruction**	**Discovery**	**Average**
Expression	4.12 (1.73)	2.58 (1.90)	3.35
Tree	6.67 (1.82)	10.19 (1.72)	8.43
Average	5.95	6.39	

In both error measures, there was the suggestion that participants who had access to neither the algebraic expression or instruction had particular difficulty, although in neither case was the interaction significant. Looking at total errors participants made 7.18 in the instruction-expression condition, 6.54 in the discovery-expression condition, 9.57 in the instruction-tree condition, and 15.40 in the discovery-tree condition. There was no significant difference among the first three error totals, $F(2, 29) = 0.35$, while the discovery-tree condition was significantly worse than each of the other conditions. When there was nothing to reveal hidden problem structure (neither expression representation nor verbal instruction), students had particular difficulty. Many participants in this condition approached the solution by trying numbers randomly and this strategy particularly increased the number of execution errors.

Conclusions

To summarize the results, there were two major findings. First, direct instruction (verbal explanation) helped students find rules for selection as shown in fewer selection errors for students who were provided with instruction than those who were left to discover selection rules for themselves. Representation task did not have a significant affect on this aspect of problem solution. Second, only type of representation task (expression rather than tree representation) affected student's success in finding rules for execution. Regardless of instruction type, students who re-represented a diagram into an algebraic expression showed fewer execution errors than those who copied a diagram into a tree.

The most striking outcome of the experiment was the equivalent performance of instruction and discovery students with respect to calculation in the execution task given a representation that revealed the intermediate steps in performing the representation. When an algebraic expression was provided, structure of the problem appeared to become transparent and the instruction (verbal explanation) seemed irrelevant. Students were able to find rules for execution using algebraic expressions and this led

to better learning outcome. This finding is also interesting in that symbolic representation is able to help understanding of visual representation. It is well known that many teachers use visual representation to help students understand mathematical symbolic notations. When learners are capable of symbolic reasoning like college level students in our study, symbolic representation also can be used to help understanding of other representations. This is consistent with the idea of "power of symbols" (Arcavi, 1994).

In contrast to performance in the execution task, verbal explanation played an important role in finding rules for selection task. Without instruction, there was nothing to uncover the obscure rules for selecting boxes for linearization. Without verbal explanation, it was hard to notice the existence of loop (from the unknown top value, two paths converge at the box with two empty values). Provision of instruction seemed to highlight features that are critical to selection task and guide students to follow the verbal instruction and practice them. This conjecture was consistent with verbal reports provided in the debriefing session. When participants were asked to explain the rules of selection and execution, only students who were given verbal explanation used the word "loop" to describe their selection rules. In contrast, many students from discovery condition reported they could not really understand a rule for selecting boxes and simply tried to select different combinations of boxes in a trial-and-error manner.

While not significant, the discovery-expression students showed the fewest execution errors. Their success depended on two features. First they were provided enough guidance that they did not suffer undue floundering in discovering the solutions. Second, the presence of the expression made it possible to infer the correct rules from their solutions. Thus, instruction is irrelevant if the principles of solved examples (worked or discovered) are transparent. On the other hand, when the structure is not transparent instruction serves in effect to annotate the examples with the features critical to their solution. On this view, students learn from solved examples and not instruction; instruction or auxiliary representations can make transparent critical aspects of these examples.

It should be possible to achieve the same equivalence of instruction and discovery students with respect to selection task by using a representation of the intermediate steps in selection. If relevant hidden steps were revealed by the tutoring system (e.g., visual loop highlighter so that students can actually identify the existence of a loop), it is expected that instruction would prove unnecessary for this aspect of the task as well. On this view, the major limitation of guided discovery is that students cannot always determine hidden structure. However, we should note in closing that this analysis does not imply any superiority for discovery learning, nor indeed were we able to find any statistically significant advantages.

Acknowledgments

The research was supported by IES grant R305A100109.

References

Arcavi, A. (1994). Symbol sense: Informal sense-making in formal mathematics. *For the Learning of Mathematics, 14*, 24-35.

Ausubel, D. P. (1964). Some psychological and educational limitations of learning by discovery. *The Arithmetic Teacher, 11*, 290–302.

Brunstein, A., Betts, S., & Anderson, J. R. (2009). Practice enables successful learning under minimal guidance. *Journal of Educational Psychology, 101*, 790-802.

Carroll, W. M. (1994). Using worked examples as an instructional support in the Algebra classroom. *Journal of Educational Psychology, 86*, 360-367.

Dean, D., & Kuhn, D. (2006). Direct instruction vs. discovery: The long view. *Science Education, 91*, 384-397.

Hiebert, J., & Wearne, D. (1996). Instruction, understanding, and skill in multidigit addition and subtraction. *Cognition and Instruction, 14*, 251-283.

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.

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.

Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction. *The American psychologist*, *59*, 14-19.

Piaget, J. (1970). *Genetic epistemology*. New York: Columbia University Press.

Renkl, A. (2002). Worked-out examples: Instructional explanations support learning by self-explanations. *Learning and Instruction, 12*, 529-556.

Rittle-Johnson, B. (2006). Promoting transfer: The effects of direct instruction and self-explanation. *Child Development, 77*, 1–15.

Schworm, S., & Renkl, A. (2006). Computer-supported example-based learning: when instructional explanations reduce self-explanations. *Computers & Education, 46*, 426-445.

Sweller, J. (1988). Cognitive load during problem solving. *Cognitive Science, 12*, 257-285.

Tuovinen, J. E., & Sweller, J. (1999). Comparison of cognitive load associated with discovery learning and worked examples. *Journal of Educational Psychology, 91*, 334-341.

Ward, M., & Sweller, J. (1990). Structuring effective worked examples. *Cognition and Instruction, 7*, 1-39.

Wittwer J., & Renkl, A. (2008). Why instructional explanations often do not work: A framework for understanding the effectiveness of instructional explanations. *Educational Psychologist, 43*, 49-64.