UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Implicit Strategies and Errors in an Improved Model of Early Algebra Problem Solving

Permalink

https://escholarship.org/uc/item/0rd281bk

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 19(0)

Authors

Koedinger, Kenneth R.
MacLaren, Benjamin A.

Publication Date

1997

Peer reviewed

Implicit Strategies and Errors in an Improved Model of Early Algebra Problem Solving

Kenneth R. Koedinger (koedinger@cmu.edu)
Benjamin A. MacLaren (ben@cs.cmu.edu)
Human-Computer Interaction Institute
Carnegie Mellon University, Pittsburgh, PA 15213

Abstract

We have been refining a cognitive model, written in ACT-R, of student performance in early algebra problem solving. "Early algebra" refers to a class of problems and competencies at the boundary between arithmetic and algebra. Our empirical studies in this domain establish a striking contrast between students' difficulties with symbolic algebra and their relative success with certain kinds of "intuitive" algebraic reasoning. To better understand this contrast, we analyzed student solutions to identify the strategies and errors exhibited and then set out to account for this detailed process data with the utility-based choice mechanism of ACT-R. Our first model contained production rules for explicitly selecting strategies and for making certain systematic errors or bugs. It provided a good quantitative fit to student performance data ($R^2=.90$), however, it had two qualitative shortcomings: 1) the productions for strategy selection appeared to serve no computational purpose and 2) the model systematically underpredicted the frequency of non-trivial errors on more complex problems. We created a new model in which explicit strategy selection was eliminated (strategic behavior is emergent) and in which failure to fire a production (an implicit, non-buggy error) is an option at every model choice point. Compared to the first model, this model achieved an equivalent quantitative fit with fewer productions and without the systematic deviations from the error data. We consider the implications of implicit strategies and errors for instruction.

Introduction

As part of a broader research effort to provide a scientific basis for improved mathematics instruction (e.g., Koedinger & Anderson, 1993; Koedinger, Anderson, Hadley, & Mark, 1995), we have been performing detailed empirical and theoretical investigations of students' developing quantitative problem solving skills. We have been performing studies, called "difficulty factors assessments" (Koedinger & Tabachneck, 1995), and are using the ACT-R theory and software (Anderson, 1993) to create detailed models of algebraic competence and its development. The focus of the work described here is on early algebra problem solving. "Early algebra" refers to a class of problems and competencies at the boundary between arithmetic and algebra.

Our empirical studies of early algebra have established a striking contrast between students' difficulties with symbolic algebra and their relative success with certain kinds of "intuitive" algebraic reasoning. Much to the surprise of most math teachers and educators (Nathan, Koedinger, & Tabachneck, 1996), high school students at the end of an algebra course are better able to solve certain algebra word problems (e.g., "A waiter gets $4.50/hr and $20 in tips one night. If he took home $38, how many hours did he work?") than the corresponding algebra equation (e.g., "4.5x + 20 = 38").

Difficulty Factor Assessments of Problem Solving

A "Difficulty Factor Assessment" (DFA) involves the use of a large set of test forms to systematically investigate what problem factors affect student difficulties in problem solving. DFAs aid in the "knowledge acquisition" process of decomposing and codifying student problem solving knowledge.

The two ACT-R models we report on here attempt to account for the affects of three factors in data from two DFA studies. Two of these factors are illustrated in Figure 1, unknown position and presentation type. The pair of problems in each row of Figure 1 differ in where the problem unknown is positioned. The problems in column 1 are called Result Unknown Problems because the unknown is the result of the process described. The problems in column 2 are Start Unknown Problems because the unknown is the start of the process described. Problems in the columns illustrate a second factor. They require the same underlying arithmetic, but differ in the representation in which they are presented. The "Story Problems" in the first row are presented verbally and include reference to a real world situation (e.g., wages). The "Word Equations" in the second row are also presented verbally but do not include a situation. The "Equations" in the third row are presented symbolically and have no situational information. Other factors we have looked at that are not illustrated in Figure 1 include number difficulty (integers versus non-integers) and the cover story used in different story problems (e.g., the "waiter story" below, or purchasing a basketball).

	Result Unknown Problems	Start Unknown Problems
Story Problems	When Ted got home from his waiter job, he multiplied his hourly wage, $2.65, by the 6 hours he worked that day and added the $66 he received in tips. How much money did Ted make that day?	When Ted got home from his waiter job, he took the amount he made that day and subtracted the $66 he made in tips. He divided the resulting amount by the six hours he worked and got $2.65, his hourly wage. How much did Ted make that day?
Word Equations	If I multiply 2.65 by 6 and then add 66, I get a number. What number do I get?	Starting with some number, if I subtract 66 and then divide by 6, I get 2.65. What number did I start with?
Equations	$2.65 * 6 + 66 = X$	$(X - 66) / 6 = 2.65$

Figure 1: Examples Combinations of Difficulty Factors

Two DFA studies of early algebra problem solving (DFA1 and DFA2) were performed with students near the end of a year-long high school algebra class. The studies revealed large effects for unknown position, problem presentation and number difficulty (integers vs. decimals). Not surprisingly students are significantly better at result-unknown (arithmetic) problems than start-unknown (algebra) problems and are significantly better at problems with integer quantity values than problems with decimal quantity values. However, it comes as a surprise to many, that these algebra students had the greatest difficulty with the equations which were significantly harder than the word equations ($p<.001$ in both studies)which in turn were only slightly harder than the story problems ($p=.23$ in DFA1 and $p<.01$ in DFA2). The effects of these three factors are for the most part independent and additive. This fact is a good indicator of the decomposability and modularity of knowledge.

There was one interaction amongst the three factors and it accounts for the small difference between word equations and story problems. Only on the decimal number problems (and not on the integer number problems) were story problems easier than word equations. As our modeling work and further data analysis revealed, the difference was accounted for by the fact that students were less likely to make a decimal alignment error in doing arithmetic (e.g., $15.90 + 66 = 16.56$) in the context of a story than without a story context. The story contexts involved money and thus students better knew to add dollars to dollars instead of dollars to cents. This example nicely illustrates the need for if parts in ACT-R productions to represent how the same operation may be performed differently in different contexts.

Two Cognitive Models of Early Algebra Problem Solving

Prior models of algebra story problem solving (e.g., Bobrow 1968; Mayer 1982; Lewis 1981) have assumed a two-step process. Story problems are converted into equations and the equations are then solved using symbolic algebra. Such a model predicts that performance on story problems must be worse than performance on equations since equation solving is a subgoal of story problem solving. Most teachers and math educators appear to share this same prediction (Nathan, Koedinger, Tabachneck, 1997). Students' greater success on word problems in DFA1 and

DFA2 indicates they must not be following this two step process. They are often using alternative informal methods for finding answers.

Students translated verbal problems to algebra equations on only 13% of problems and 53% of these attempts led to success. More often students attempted to solve verbal start-unknowns using one of two informal strategies, "guess-and-test" and "unwind." In guess-and-test, a value for the unknown is guessed at and that value is propagated through the known constraints. The guess is then adjusted and the process repeated until the correct answer is arrived at. Guess and test was used about 20% of the time on the verbal start unknown problems. By far the most common strategy, the informal "unwind" strategy, was used almost 40% of the time. Unwind is a verbally mediated strategy, where students work backwards from the given result value, inverting operators along the way, to produce the unknown start value.

Our modeling work has the following goals:

1) to characterize students' early algebra problem solving strategies and common errors,

2) to provide an explanation for students' surprising success on word problems over equations, and

3) to capture the essential knowledge differences between good and poor early algebra problem solvers,

4) to create a developmental model of the learning trajectory and strategies that account for students' transition to competence in early algebra problem solving

A Comparison of EAPS1 and EAPS2

In MacLaren & Koedinger (1996), we reported on our initial ACT-R model of this data. We will refer to that model as EAPS1 for "Early Algebra Problem Solver 1." After exploring the limitations in that model (discussed below), we created EAPS2. In both models the general sequence of events is 1) comprehend the problem presentation (whether story, word, or equation) to extract relevant arithmetic operators and their arguments, 2) manipulate the operators as necessary (e.g., invert them), and 3) solve any arithmetic subgoals that are produced.

Both models are also capable of two types of translations between representations. A given verbal problem may be translated into an equation and then the equation is solved algebraically (the traditional two step process). Alternatively, a given equation problem may be interpreted

verbally and thus solved using an informal strategy like guess-and-test or unwind.

EAPS1 made an initial decision about what strategy to employ, using explicit strategy selection productions, which constrained future matching. In contrast, EAPS2 has no explicit notion of strategy: it simply recognizes and executes operations it can perform on a given representation.

At the initial strategy selection choice point, EAPS1 could choose the give-up "strategy", a strategy students use that has low benefit, but also low cost. EAPS2 does not have an explicit give-up strategy, but more generally at any choice point if no production has a high enough utility then the model gives up on the problem.

We model two types of errors: arithmetic and conceptual. Conceptual errors include things like forgetting to change the sign when removing an operator in the verbal representation or confusing the order of operations in the symbolic representation. For arithmetic errors, we model bugs (miss-alignment of decimal places in doing arithmetic) and slips (e.g., $2 * 3 = 5$). Bugs and slips are each modeled by a single production (abstracting over detailed arithmetic errors, such as carry errors and borrowing from zero). In the new model, the give-up production results in different errors, depending on when it fires. Giving up at the beginning of a problem or before "writing" the results of an arithmetic operation results in No-Answer. Giving up an arithmetic operation produces an arithmetic error. Giving up at any other choice point produces a conceptual error.

Coding Student Solutions for Errors and Strategies. Student solutions for result-unknown problems were coded into 4 categories: correct, arithmetic error, conceptual error and no-answer. For each of the six types of result-unknown (arithmetic) problems (rows in Table 1) we computed the frequency of these codings in students' solutions. These frequencies are shown as percentages in bold in the sub-columns labeled "data" in Table 1 (the sub-columns labeled "d1" and "d2" are model prediction deviations for EAPS1 and EAPS2 which will be described later). Note that the sum of the data frequencies in each row of Table 1 add to 100% since every solution gets coded into one of these categories. Also note that the coding of all but the conceptual error category is fairly straight-forward, thus this category also includes all solutions which were otherwise categorized.

For each of the six types of the more algebra-like start-unknown problems, we not only coded correctness and the three broad error categories, but also did a broad strategy coding identifying when solutions involved the use of the formal algebraic equation solving strategy versus an informal strategy, either guess-and-test or unwind. The columns of Table 2 show the combined strategy-error categories for the two strategy codes and four error codes (correct, arithmetic error, and conceptual error codes are separated into those occurring within an informal vs. formal strategy). Again, the data frequencies in the rows of Table 2 sum to 100%.

Fitting the Models to the Data. After developing a knowledge-level model that could be guided through the space of decisions, we set ACT-R's conflict resolution parameters to stochastically select productions consistent with the "average student" from DFA data. ACT-R includes a rational control mechanism based on decision theory, which uses parameters such as the likelihood that executing a production will eventually satisfy the current goal and the cost of executing a production. Also, ACT-R predicts that Gaussian noise will sometimes cause a production to be selected other than the one with the highest estimated utility. These features enabled us to model the student data by setting the noise and production parameters so that the model would make choices that

Table 1: Data & model differences for result-unknown (arithmetic) problems.

Representation	Correct			Arithmetic Errors			Conceptual Errors			No Answer		
	data	dif1	dif2	data	dif1	dif2	data	dif1	dif2	data	dif1	dif2
Easy Story	77	3	-2	1	3	-1	17	-5	-6	5	0	9
Easy Word	84	-5	-6	5	2	-5	5	5	5	7	-3	6
Easy Equation	65	-1	-8	7	-5	-7	12	-6	8	16	11	7
Hard Story	63	6	-1	17	-5	-3	11	1	1	9	-2	3
Hard Word	42	7	5	36	-3	-4	21	-9	-11	0	7	11
Hard Equation	33	2	3	24	4	1	9	-1	6	33	-4	-8

Table 2: Data & model differences for start-unknown (algebra) problems.

Rep.	Informal Strategy									Formal Strategy									No Answer		
	Correct			Arith Error			Conc. Error			Correct			Arith Error			Conc. Error			(Giveup)		
	data	d1	d2	data	d1	d2	data	d1	d2	data	d1	d2	data	d1	d2	data	d1	d2	data	d1	d2
Easy St	64	-4	-3	2	3	-2	14	-2	6	6	0	-3	0	1	0	2	1	1	14	1	-2
EsyWd	70	-8	-11	0	5	0	19	-7	3	0	6	6	0	1	0	2	2	2	9	3	1
EasyEq	35	2	-9	0	2	0	19	-10	8	19	0	-3	0	3	0	9	-5	1	19	10	2
HardSt	45	7	4	10	4	2	24	-10	-2	4	-4	-1	0	9	2	1	2	4	15	-5	-7
HrdWd	27	0	7	9	14	19	30	-15	-5	6	-3	-5	0	2	2	9	-8	-7	18	-6	-9
HardEq	12	14	4	6	9	3	18	-12	8	6	6	3	0	7	8	15	-12	-8	42	-13	-15

correspond with the frequency of strategy and error codes in the DFA data.

In fitting the model and data, we categorized the problems from DFA1 and DFA2 into 12 categories by crossing the levels of the three difficulty factors: unknown position (result-unknown vs. start-unknown), representation (story vs. word vs. equation), and number difficulty (integer vs. decimal arithmetic). The 12 problem categories are shown in the column labels of Table 3 and the row labels of Tables 1 (just the six result-unknown problem categories) and 2 (just the six start-unknown categories).

Parameter setting in mathematical models is typically done using an iterative gradient-descent algorithm and depends on fast computations of model predictions given any vector of parameter values. Parameter setting with an ACT-R model is made more challenging because 1) computations of ACT-R model predictions are not as simple as evaluating mathematical expressions. Rather, they involve interpretation of production rules and, more importantly, 2) the stochastic nature of ACT-R means the model has to be run multiple times (e.g., 200) on each problem category. Thus an iterative approach may not be practically feasible in many cases.

In fitting parameters for EAPS1 we developed an alternative "incremental complexity" strategy. We started by setting the parameters for the simplest group (result-unknown integer verbal arithmetic) that contains core productions common to every group. Then, we fit parameters for the new productions needed in each slightly more complex group of problems as determined by changing one difficulty factor at a time.

To set the parameters for the EAPS2, we created a mathematical model in an Excel spreadsheet that corresponds with the behavior of the EAPS2 ACT-R model. Equations in the mathematical model corresponded to percentages for broad error categories (no error, arithmetic errors, conceptual errors and no answer).

Each equation for a broad error category consisted of sums of products, where each product corresponded to a path through the model that resulted in that broad error category. For example, one of the terms in the equation for arithmetic errors for easy verbal arithmetic would be VE*AR*VE*SL representing the path to perform the first verbal extraction (VE) and arithmetic operation (AR) correctly but after the second correct verbal extraction (VE), the model performs an arithmetic slip (AR). We then used Excel's solver tool to find best fitting utility parameters to fit the DFA data, constraining the conceptual errors already accounted for in EAPS1 to remain the same, and inserted the resulting parameters into ACT-R.

Table 3 shows the central productions in EAPS2 that we tuned (in the left-most column) and for each problem type (along the top), what productions apply for that type. For example, for easy story arithmetic (Arth Easy Stry) there is one argument extraction production, Verbal-Extract-Args. Since no operator inversion is required for arithmetic, only the arithmetic productions apply, and because the arithmetic is easy, only the correct production Arith-Procedure and the simple Arith-Proc*Slip applies. In contrast to the simplest problem type, hard algebra equations on the far right have several more productions that apply.

EAPS2 has 11 parameters: two for argument extraction, one for translating a verbal representation into an equation, two for manipulating an equation, (one correct, one incorrect), and four for arithmetic. EAPS1 had 13 parameters: six for strategy selection, two for argument extraction, one for give-ups resulting in incomplete, one for operator inversion, and three for arithmetic (correct, correct arithmetic on story problems, and arithmetic bugs).

Table 3: Summary of parameters and problems they apply to for EAPS2

Productions	Expected Gain	Arth Easy Stry	Arth Easy Wrd	Arth Easy Eq	Arth Hrd Stry	Arth Hrd Wrd	Arth Hrd Eq	Alg Easy Stry	Alg Easy Wrd	Alg Easy Eq	Alg Hrd Stry	Alg Hrd Wrd	Alg Hrd Eq
Argument Extraction													
Vrb*Extrct-Args	6.2	X	X	-	X	X	-	X	X	-	X	X	-
Sym*Extract-Args	5.6	-	-	X	-	-	X	X	X	X	X	X	X
Translate-Vrb-to-Sym	4.9	-	-	-	-	-	-	X	X	-	X	X	-
Translate-Sym-to-Sym	4.8	-	-	-	-	-	-	X	X	X	X	X	X
Sym*Order-of-ops-bug	0.0	-	-	X	-	-	X	X	X	X	X	X	X
Operator Interp/Inv													
Vrb*Unwind-Correct	6.4	-	-	-	-	-	-	X	X	X	X	X	X
Vrb*Unwind-Error	4.0	-	-	-	-	-	-	X	X	X	X	X	X
Arithmetic													
Arith-Procedure	18.6	X	X	X	X	X	X	X	X	X	X	X	X
Arith-Proc*Sit-Assist	18.8	-		-	X	-	-	-	-	-	X	-	-
Arith-Proc*Slip	4.0	X	X	X	X	X	X	X	X	X	X	X	X
Arith-Proc*Bug	17.9	-	-	-	X	X	X	-	-	-	X	X	X

385

Model-Data Fit

The results of our parameter tuning can be seen in Tables 1 and 2. The comparison is presented as sets of triples: first the DFA data, then the EAPS1–DFA difference, and then the EAPS2–DFA difference. Table 1 shows the results for arithmetic (result unknown) problems. Table 2 shows the results for algebra (start unknown) problems broken down into formal and informal strategies.

Both models do a good job of capturing the effects of the three difficulty factors on student error and strategy selection behavior. This is illustrated with the two model predictions (shown as deviations from the data) for 66 data points in Tables 1 and 2. 90% of the data points for EAPS1 and 93% of the data points for EAPS2 deviate from the DFA data by less than ten percentage points. The R^2 value for EAPS1 was 0.90 using 13 parameters; and for EAPS2 0.92 using 11 parameters.

The major weakness of EAPS1 was qualitative. It underpredicted the frequency of conceptual errors on algebra problems. Notice that for informal strategies in Table 2, the conceptual errors for EAPS1 were consistently underpredicted (most were -10 or worse). The conceptual errors on formal strategies also tended to be too low. The problem is that EAPS1 only makes conceptual errors through buggy productions, but many of students' conceptual errors may result from lack of knowledge rather than inappropriate knowledge. To model this in EAPS2 we modified the simulation so that it might give up at any choice point in the solution. The results was that unlike EAPS1, the predictions of EAPS2 are not systematically different from the data (see informal conceptual error column in Table 2).

One limitation of both models is that students appear much more likely to give up on hard algebra equations (42% of the time) than the model predicts (29%). Neither model considers number difficulty in the productions that begin the processing of a problem, that is, strategy selection productions in EAPS1 and comprehension productions for argument extraction in EAPS2. In contrast to both models, students may be considering downstream arithmetic difficulties up front and this anticipated difficulty combined with a weak (low estimated utility) production for equation comprehension leads to a greater frequency of providing no answer. In principle, EAPS2 could model this effect by giving up at the first arithmetic subgoal, before performing any written arithmetic. However, in practice we found that focusing the parameter fitting process on this data point lowered the overall fit of EAPS2.

In summary, EAPS2 achieved an equivalent quantitative fit as EAPS1 with fewer productions and without the systematic deviations from the error data.

Conclusion

In EAPS1 we modeled student's informal solutions to start-unknown algebra problems as strategies that were explicitly selected and globally applied. Inspecting the resulting model, we noticed the productions doing strategy selection were not achieving any significant computational purpose. We found we could eliminate them without substantially changing the model's behavior or reducing it's fit to the data. Rather than global strategy selection and application, in EAPS2 any production that applies to the current task demand may fire – there is no global strategy selection. What appears as strategic behavior on the surface is emergent from individual local choices.

A second weakness of EAPS1 was a qualitative deviation from the data whereby it systematically underpredicted the frequency of conceptual errors on start-unknown problems. This deviation was the consequence of EAPS1 having explicit bugs (e.g., order of operations confusion) as its only way of producing conceptual errors. In EAPS2, we added the possibility that the model might fail to find anything to do at any particular choice point. In this way, it produced conceptual errors not through explicit buggy knowledge but implicitly through the failure to have any productions with a sufficient estimated utility.

The notions of implicit strategies and errors emphasize that much of what we learn is tacit knowledge. Such knowledge is acquired in context and by doing. Trying to directly communicate strategies may not be an effective instructional method for such tacit procedural knowledge. Similarly, trying to diagnose deep bugs to account for student's errors is not always an effective approach. Rather, what may be more critical is creating activities that challenge students on just the knowledge they are in reach of learning, in other words, activities that are within a student's zone of proximal development (Vygotsky, 1978).

The EAPS model provides a principled way for identifying students' developmental capabilities. We have begun to fit EAPS2 to different subsets of the students that participated in DFA1 and DFA2 that showed different levels of competence. The utility parameters on the productions within various competence levels illustrate the underlying continuities in the learning process. Inspecting the parameter fits for lower competence levels (e.g., students who can only solve verbal result-unknown problems) reveals that despite generally failing on more difficult problems (e.g., verbal start-unknown problems) these students have some level of competence, that is, non-zero utility estimates, on productions relevant to those more difficult problems (e.g., the Vrb*Unwind*Correct production).

The difference between students at the same competence level but with different zones of proximal development (i.e., the level that be achieved with assistance) can be characterized in terms of differences in production utilities on relevant but not yet mastered skills. While both students are below some threshold on this skill, one student may have a utility value that is much closer to the threshold than the other.

Our current research involves simulating the learning process using ACT-R utility estimation algorithms. We are beginning to perform experiments to test how different activity selections and other forms of assistance may effect the rate or trajectory of skill development.

Acknowledgments

References

Anderson, J. R. (1993). Rules of the Mind. Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1990). *The Adaptive Character of Thought*. Hillsdale, NJ: Erlbaum.

Brown, A. L. (1994). The advancement of learning. *Educational Researcher, 23, 8,* 4-12.

Carpenter, T. P., & Fennema, E. (1992). Cognitively Guided Instruction: Building on the knowledge of students and teachers. In W Secada (Ed.), Curriculum reform: The case of mathematics in the United States. Special issue of the *International Journal of Educational Research,* (pp 457-470). Elmsford, N.Y.: Pergamon Press.

Koedinger, K. R., & Anderson, J. R. (1993). Effective use of intelligent software in high school math classrooms. Artificial Intelligence in Education: Proceedings of the World Conference on AI in Education, AACE: Charlottsville, VA.

Koedinger, K. R., Anderson, J. R.., Hadley, W. H., & Mark, M. A. (1995). Intelligent tutoring goes to school in the big city. In *Proceedings of the 7th World Conference on Artificial Intelligence in Education*, (pp. 421-428). Charlottesville, VA: Association for the Advancement of Computing in Education.

Koedinger, K.R., & Tabachneck, H.J.M. (1994). *Two strategies are better than one: multiple strategy use in word problem solving.* Presented at the annual meeting of the American Educational Research Association, New Orleans, LA.

Koedinger, K.R., & Tabachneck, H.J.M. (1995). *Verbal reasoning as a critical component in early algebra.* Paper presented at the 1995 annual meeting of the American Educational Research Association, San Francisco.

Lewis, C. H. (1981). Skill in algebra. In J. R. Anderson (Ed.), *Cognitive Skills and Their Acquisition* (pp. 85-110).

MacLaren, B. A., & Koedinger, K. R. (1996). Toward a Dynamic Model of Early Algebra Acquisition. In Proceedings of the European Conference on AI in Education: Lisbon, Portugal.

Mayer, R. E. (1982). Different problem-solving strategies for algebra word and equation problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 8,* 448-462.

Nathan, M. J., Koedinger, K.R., Tabachneck, H. T., Difficulty Factors in Arithmetic and Algebra: The Disparity of Teachers Beliefs and Students Performance. Paper prepared for The 1996 American Education Research Association Annual Meeting, New York.

Tabachneck, H. (1992). Computational differences in mental representations: Effects of mode of data presentation on reasoning and understanding. Doctoral Dissertation. Carnegie Mellon.

Tabachneck, H. J. M., Koedinger, K. R., & Nathan, M. J. (1994). Toward a theoretical account of strategy use and sense-making in mathematics problem solving. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Tabachneck, H. J. M., Koedinger, K. R., & Nathan, M. J. (1995). A cognitive analysis of the task demands of early algebra. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Vygotsky, L. S. (1978). Mind in society. Cambridge, MA: Harvard University Press