

Thompson, P. & Thompson, A. (1987). Computer presentations of structure in algebra. In J. Bergeron, N. Herscovics, & C. Kieran (Eds.), *Proceedings of the Eleventh Annual Meeting of the International Group for the Psychology of Mathematics Education, 1*, 248-254.

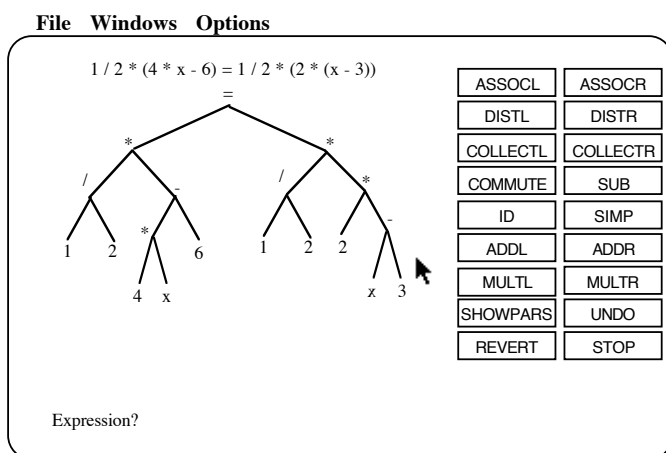
## COMPUTER PRESENTATIONS OF STRUCTURE IN ALGEBRA

Patrick W. Thompson  
Alba G. Thompson  
Illinois State University

*Many errors committed by students of algebra appear to be a result of their long-term inattention to structure of expressions and equations. A special computer program was developed that enabled students to manipulate expressions, but which constrained them to acting on expressions only through their structure. Eight leaving-seventh graders used the program for eight days. An analysis of their actions indicated that errors due to inattention to structure occurred largely while they were first learning a field property or identity, and that afterwards such errors were infrequent.*

Typical errors found in previous studies of students' errors in algebra suggest that students studying algebra frequently fail to realize that formulas in mathematical symbol systems have an intrinsic structure (Lewis, 1981; Matz, 1982; Sleeman, 1982, 1984, 1985). In algebra, expressions are structured explicitly by the use of parentheses, and implicitly by assuming conventions for the order in which we perform arithmetic operations. It is hypothesized that many of students' errors in manipulating an algebraic expression are due to their inattention to the expression's structure.

To test this hypothesis, we built a program, called EXPRESSIONS, that presents expressions and equations in two formats: in usual (sentential) form and in the form of an expression tree. The figure to the right shows the screen after having entered the equation  $4x-6=2(x-3)$  and then having multiplied both sides by  $\frac{1}{2}$ . The equation's expressions are shown in sentential notation at the top of the screen. The tree representation of the equation is shown



To change an expression by the use of a field property or other transformation, students put the mouse pointer on top of one of the buttons along the right side of the screen and then clicked the mouse to select that action. Then they put the pointer on top of the operation in the tree representation of the expression which defined the expression or subexpression to be transformed, and clicked the mouse again. The action was performed on the selected expression or subexpression, and the sentential notation and expression tree were changed accordingly.

To transform an expression by the use of an identity, students put the mouse pointer on top of the ID button, clicked the button, and then clicked the operation sign within the tree which defined the expression to which the identity was to be applied. The computer applied whichever of these identities was appropriate to the chosen expression or subexpression:  $a-b = a+ -b$ ,  $\frac{x}{y} = x*\frac{1}{y}$ ,  $-x = -1*x$ , or  $x = 1*x$ , and then updated the expression tree and sentential display accordingly.

### Sample

The sample consisted of eight leaving-seventh graders—six males and two females—from the ISU elementary laboratory school and who volunteered to participate in the study. Their mean age was 13 years 1 month; their mean cumulative mathematics score on the Iowa Test of Basic Skills was 74.6. In the last quarter of seventh grade mathematics, five students received an A, one received a B, and two received a C.

### Method

The study took place over nine consecutive weekdays in June of 1986. The first session was devoted to administering a pretest; eight sessions (50 minutes each) were given to direct instruction and practice. The pretest involved assessing students' knowledge of the conventions for order of operations (evaluating numeric expressions), their knowledge of field properties, and their knowledge of variables.

Instruction took place in a classroom at ISU, where the instructor used a Macintosh running EXPRESSIONS. The Macintosh was connected to a projector which created a 6' x 6' image of the screen. All instruction was videotaped.

For practice sessions, students were grouped in pairs by matching their cumulative mathematics score on the Iowa Test of Basic Skills. Practice sessions took place with students in two locations: in a computer room and in the classroom, with two students per computer. Students using the classroom computer were videotaped. Each pair of students was videotaped

once. A set of booklets containing examples and practice problems were provided to each student. All students used a version of the program that stored their keystrokes and mouse-clicks in a data file which could be “played back” for later analysis.

Instruction proceeded in this order: order of operations in arithmetical expressions; field properties as transformations of arithmetical expressions; identities and derivations. An outline of the eight days of instruction is given in Table 1.

<u>Day</u>	<u>In Class</u>	<u>In Groups</u>
2	Order of operations; Evaluating expressions	Worksheet 1 (Parts 1 & 2)
3	Parentheses; Expression trees	Worksheet 1 (Part 3); Worksheet 2
4	Discuss Worksheet 2; Commutativity; Associativity; Example from Worksheet 3	-----
5	-----	Worksheet 3 (Part 1)
6	Review commutativity; associativity; introduce distributing and collecting	Worksheet 3 (Part 2)
7	-----	Worksheet 3 (Part 2); Worksheet 4
8	Review field properties; introduce identities	-----
9	-----	Worksheet 5

Table 1. Summary of instruction.

The worksheets comprised an integral part of instruction. Table 2 shows the numeric-transformation problems students worked in sessions 6 and 7. Table 3 shows the identity derivation problems students worked in session 9.

	<u>Start With</u>	<u>Change It To</u>		<u>Start With</u>	<u>Change It To</u>
N1.	$5*(4+3)$	$3*5 + 4*5$	I1.	$(z - q)*u$	$z*u - q*u$
N2.	$5*((4+3)+2)$	$(5*4)+((2+3)*5)$	I2.	$r*(s/t)$	$(r*s)/t$
N3.	$(7+3)*(6+5)$	$(7*6+7*5)+(3*6+3*5)$	I3.	$-(p + q)$	$-p + -q$
N4.	$(6+^5)*(6+5)$	$(6*6) + (^5*5)$	I4.	$(a + b)/c$	$a/c + b/c$
N5.	$3*(8+4) + 9*(4+8)$	$(9+3)*(8+4)$	I5.	$6x + x$	$7x$
N6.	$3*(6/9) + (6/9)*7$	$10*(6/9)$	I6.	$5x - x$	$4x$
N7.	$-5*3 + (2+3)*5$	$0 + 10$	I7.	$x + x$	$2x$
N8.	$(5+9)*(5+9)$	$5*5 + 90 + 9*9$			

Table 2. Numeric transformation problems.

Table 3. Identity derivation problems

## Results

### Pretest

Six of the eight students processed numeric expressions from left to right, ignoring conventions for order of operations (e.g.,  $8 - 6 + 5 * 3$  evaluates to 21), when grouping was not given explicitly. All eight were familiar with commutativity. Seven were familiar with associativity in its simplest form. None was familiar with distributivity. Six differentiated among expressions and equations on the basis of superficial characteristics (e.g. “ $y+2=5$  is different from  $x-2=5$  and  $x+2=5$  because it uses  $y$  and the others use  $x$ .”)

### Analysis of computer use

EXPRESSIONS was modified to store all interactions. The stored files were then later rerun for analysis. Students’ actions were categorized according to the following scheme:

- A Appropriate transformation applied at an appropriate place in the expression, given the current and goal expressions.
- IA Inappropriate transformation, e.g. trying to use the distributive property on  $(a*b)+c$ .
- AWP Appropriate action, but applied in a wrong place. This was inferred if a student tried the same transformation twice in a row, first trying it at an inappropriate place in the expression and then applying it appropriately.
- CD Confused direction. An action was placed in this category if a directional transformation was appropriate (such as using the associative property of multiplication to change the grouping from being on the left to being on the right) but the student chose the wrong direction.

### Transforming Numeric Expressions

Table 4 shows the percents of students’ actions falling within each category while working the numeric transformation problems (Table 2). Table 5 shows the percents of students’ actions falling within each category while working the identity derivation problems (Table 3).

In many cases, the majority of inappropriate actions occurred early in a problem, suggesting that students were exploring the effects of the available transformations upon expressions. To eliminate the effects of exploratory errors upon the percents in Tables 4 and 5, the data were reanalyzed by the same categorization scheme as previously, but with this exception: All actions prior to two consecutive appropriate actions were discarded. Tables 6 and 7 show the percents of “non-exploratory” actions falling within each of the categories.

<i>Problem</i>	<i>A</i>	<i>IA</i>	<i>AWP</i>	<i>CD</i>	<i>Problem</i>	<i>A</i>	<i>IA</i>	<i>AWP</i>	<i>CD</i>
N1	87	0	0	13	I1	41	48	6	6
N2	75	25	0	0	I2	*	*	*	*
N3	69	9	17	6	I3	*	*	*	*
N4	60	31	8	1	I4	69	23	0	8
N5	82	0	18	0	I5	56	38	6	0
N6	88	0	0	12	I6	81	12	6	0
N7	56	39	5	0	I7	80	20	0	0
N8	89	11	0	0					

Table 4. Numeric transformations: Percent per category of all actions. All students completed all problems.

Table 5. Identity derivations: Percent per category of all actions; “\*” indicates incomplete data.

The differences between Tables 4 and 6 and between Tables 5 and 7 suggest that students’ errors were due to initial play involved in understanding the problems, understanding the available transformations, and making connections between the two. Once students internalized the transformations’ structural constraints, they were less likely to commit errors and were more efficient in their solution strategies.

<i>Problem</i>	<i>A</i>	<i>IA</i>	<i>AWP</i>	<i>CD</i>	<i>Problem</i>	<i>A</i>	<i>IA</i>	<i>AWP</i>	<i>CD</i>
N1	100	0	0	0	I1	58	39	3	0
N2	95	5	0	0	I2	*	*	*	*
N3	85	7	4	4	I3	*	*	*	*
N4	92	4	0	4	I4	86	0	0	14
N5	82	0	18	0	I5	100	0	0	0
N6	88	0	0	12	I6	100	0	0	0
N7	100	0	0	0	I7	100	0	0	0
N8	94	6	0	0					

Table 6. Numeric transformations: Percent per category of non-exploratory actions.

Table 7. Identity derivations: Percent per category of non-exploratory actions; “\*” indicates incomplete data .

Exploratory errors were commonly either irrelevant to the problem being solved (e.g., “what does this button do?”) or were attempts at doing something that might take an expression closer to its goal state. For example, one error was to try to use associativity to change  $(a+b)*c$  into  $a+(b*c)$ , to which the computer “responded” by doing nothing. The students wanted  $b$  to be multiplied by  $c$ , and apparently concluded that the associative property would do that regrouping for them. Also, it was common for students to repeat an errorful action. It appeared that repeating an action supported students in their attempts to reflect on the reasoning they used in first

choosing the action, and supported them in understanding the reason that the chosen transformation did not accomplish whatever they had in mind.

## Discussion

Previous studies of students' errorful manipulation of expressions and equations proposed that their errors are due to mal-formed rules—perturbations of correct rules. This study asked whether or not such errors were due to students' inattention to structural features of expressions and transformations thereupon. The results suggest that mal-rules need not be a natural occurrence when students operate in an environment that supports explicit attention to expressions' structures, and where structure also imposes constraints on students' actions. We cannot say from the results presented here that errors reported in previous studies were due to students' inattention to structure, but these results indicate that attention to structure is an important consideration.

Students could attempt errorful transformations of expressions while using the computer, but the computer would not carry them out. It appeared that they interpreted this context as one where experimentation became natural and beneficial. We would like to think that students disposition to experimentation was a result of the software and the use made of it. However, it also could have been a result of the instructor's style of instruction, or it could have been that this particular group of students was predisposed to experimentation and reflection.

A limitation of the study is that students were not assessed outside of the computer environment. It is quite conceivable that had these students been left to their own devices, they would have committed errors on paper and pencil that they learned not to make while using the computer. The issue of transfer from computer to noncomputer environments requires extensive research.

Another limitation of the study is that we do not know the depth of commitment that these students had when they "proved" that two expressions were equivalent, or when they derived an identity. Did students think of an identity as a theorem that could be applied in other contexts? We do not know.

A feature of structure which we could not address here with data, but which was addressed explicitly in the study, was that of variable. Many problems (all of those in Tables 2 and 3) were designed so that students would have to treat a subexpression as a unit. When applying field properties and identities to expressions, students regularly needed to substitute a subexpression in an expression for a letter in the canonical statement of a property or identity. They became quite adept at this. Also, students felt no discomfort when letters were first introduced in to-be-transformed expressions. Apparently, by having them transform numerical

expressions, they became used to the idea that expressions could be manipulated regardless of their constituent elements. Thus, when letters were introduced, students saw no obstacle in continuing what they had already learned to do with numerical expressions. The approach wherein manipulating algebraic expressions is presented as a natural extension of manipulating numerical expressions deserves further research.

The use of expression trees as one of the representational systems within the computer program proved to be a positive feature of instruction. Students found expression trees to be quite intuitive. When doing Worksheet 1, which focused upon evaluating expressions given in sentential form, students used EXPRESSIONS only to check their answers. They were told only that they needed to click SIMP and then click the top of the tree to evaluate an expression. We found four students who constructed expression trees for complex expressions as an aid to evaluating them, *even though there had been no discussion about how expression trees are constructed, and these students had never before seen an expression tree.*

Finally, it should be noted that in eight days of instruction these leaving-seventh grade students went from essentially no working knowledge of order of operations to deriving algebraic identities, and did so with some depth of understanding. Even with the limitations stated earlier in this discussion, the fact that such coverage is possible makes us question assumptions that are built into traditional junior high school pre-algebra and algebra curricula about what one can expect of junior high school students in the United States.

### References

- Sleeman, D. H. (1982). Assessing competence in basic algebra. In D. H. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 186-199). New York: Academic Press.
- Sleeman, D. H. (1984) An attempt to understand students' understanding of basic algebra. *Cognitive Science*, 8, 387-412.
- Lewis, C. (1981). Skill in algebra. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 85-110). Hillsdale, NJ: Erlbaum.
- Matz, M. (1982). Towards a process model for high school algebra errors. In D. Sleeman, & J. S. Brown (Eds.), *Intelligent tutoring systems*. New York: Academic Press.
- Sleeman, D. H. (1985). Basic algebra revisited: A study with 14-year olds. *International Journal of Man-Machine Studies*, 22, 127-149.