Title
Constituent Structure in Mathematical Expression

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 22(22)

Authors
Jansen, Anthony R.
Marriot, Kim

Publication Date
2000

Peer reviewed

Constituent Structure in Mathematical Expressions

Anthony R. Jansen (tonyj@csse.monash.edu.au)
School of Computer Science and Software Engineering
Monash University, Victoria, Australia

Kim Marriott (marriott@csse.monash.edu.au)
School of Computer Science and Software Engineering
Monash University, Victoria, Australia

Greg W. Yelland (Greg.W.Yelland@sci.monash.edu.au)
Department of Psychology
Monash University, Victoria, Australia

Abstract

Previous research has suggested that human perception of mathematical expressions is based on syntactic structure. Here, we extend our understanding of how humans perceive algebraic equations in two ways. First, we examined the hypothesis that the internal representation used by experienced mathematicians is based on the phrasal structure of the parse tree. This was tested using a memory recognition task, and the results supported the hypothesis. Second, we explored how much experience with mathematics is necessary before such representations become established. Participants were young students with very little experience with algebra. Surprisingly, the students appeared to encode equations in a manner similar to experienced mathematicians.

Introduction

Mathematical notation and natural language share many common features. Both have a well-defined syntax and semantics, and both allow for the expression of abstract information. However, an important difference is that the layout of mathematical notation is two dimensional in nature, with equations relying on both vertical and horizontal adjacency relationships between the symbols to provide the meaning. It is natural then to ask how humans comprehend mathematical expressions.

The present paper extends our understanding of how humans perceive mathematical expressions in two ways. First we explore the nature of the internal representation used to encode equations. Specifically, we examine whether the information recovered from equations has a parse tree structure similar to that used to represent sentences of natural language. Second, we explore how much experience with mathematics is necessary before such representations become established.

For many years now, phrase structure grammars have been used to understand the way that humans parse natural language sentences (for example, see Akmajian, Demers and Harnish, 1984). This allows the constituent structure of a natural language sentence to be represented diagrammatically by a parse tree, containing various phrases such as noun and verb phrases. Although phrase structure grammars can only be applied to sequential languages, variations of such grammars have been proposed in an attempt to enable computers to understand mathematical notation (for example, see Anderson, 1977). Analogously to natural language, parse trees for equations can be created based on mathematical syntax. An example parse tree is given in Figure 1.

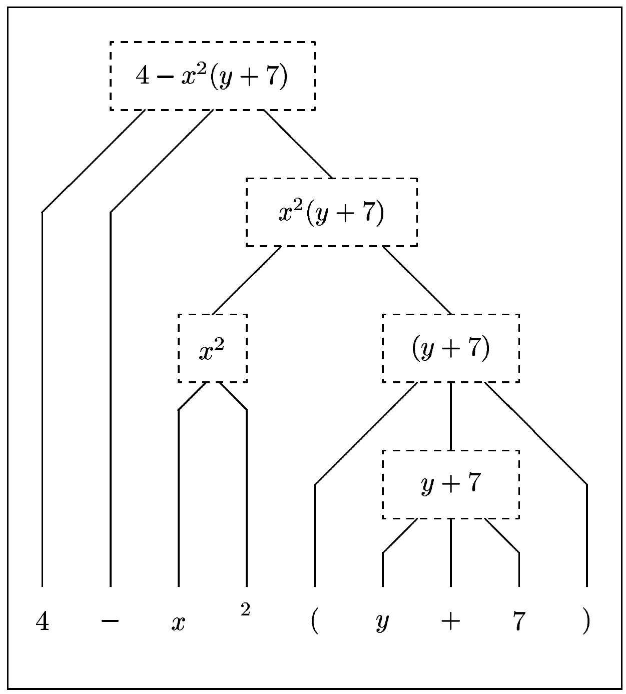

Figure 1: Parse Tree for $4 - x^2(y + 7)$

Our previous work on the comprehension of mathematical expressions (Jansen, Marriott and Yelland, 1999) has shown that in a memory recognition task, experienced users of mathematics can more readily identify those parts of a previously seen equation that are syntactically well-formed (that is, which have a coherent mathematical meaning, such as $y + 7$ in the above example), than those that are not well-formed (for example, $-x^2(y$ which is also part of the equation, but does not convey any coherent mathematical meaning on its own). This result provides support for the notion that the internal representation used by mathematicians is based on mathematical syntax. This accords with results by Johnson (1968, 1970) which show that in the context of natural language, chunking of sentences is also guided by syntax. The work of Ranney (1987) also shows that even after only brief exposure, the structure of algebra expressions provide information about the category of the symbols in that expression (whether they are variables, numbers, operators, etc.). This indicates that the parsing of such expressions is based on structural content.

Experiment 1

It is clear that mathematical syntax plays an important role in encoding equations, however it does not necessarily follow that a parse tree structure underlies the internal representation. The first experiment explores this idea with respect to moderately complex algebra expressions. Our hypothesis is that the internal representation used by experienced mathematicians is based on the phrasal structure of the parse tree. To test this hypothesis, we have set up a recognition task to see if participants can more readily recognize sub-expressions of an equation that form a phrasal node on the parse tree (for example, $y + 7$ in the previous example) as opposed to sub-expressions that are also syntactically valid, but do not form a phrasal node on the parse tree (such as $4-x^2$). If our hypothesis is correct, we would expect to see a recognition advantage for the phrasal sub-expressions.

Method

Participants Twenty-four participants successfully completed the experiment. All were staff members, graduate or undergraduate students from the Computer Science department, all competent mathematicians who were experienced with algebra. All participants were volunteers between the ages of 18 and 35 years, with normal or corrected-to-normal vision. Data from an additional 8 participants were not included due to excessive error rates. [1]

Materials and Design Seventy-five equations were constructed, all consisting of between twelve and fourteen characters. The equations contained at most one fraction and the variable names were x and y, since these are most commonly used. For each equation, sub-expressions of three types were constructed.

a) A *phrasal sub-expression*, which is a syntactically well-formed component of its equation, which conveys the same meaning on its own that it conveyed in the equation. It is a phrasal node in the equation's parse tree.

b) A *non-phrasal sub-expression*, which is also a well-formed component of its equation, but does not convey the same meaning on its own that it conveyed in the equation. It is not a phrasal node in the equation's parse tree.

c) An *incorrect sub-expression*, which was not part of the original equation. It is also a well-formed expression. These act as fillers.

Each of the sub-expressions contained between four and six characters (the average for phrasal sub-expressions was 4.78; for non-phrasal, 4.54; for incorrect, 4.60). See Table 1 for examples of equations and sub-expressions used. As the examples show, a variety of sub-expressions were used, some of which were bracketed, but most of which were not.

In order to present all three sub-expression types for each equation, but ensuring that participants were presented with each equation only once to avoid practice effects, three counterbalanced versions of the experiment were constructed. For each version, there were twenty-five instances of each type of sub-expression. Two additional equations were constructed as practice items. The same practice items were used in each version. Eight participants completed each version, each receiving the items in a different pseudo-random order.

Procedure Participants were seated comfortably in an isolated booth. Items were displayed as black text on a white background on a 17" monitor at a resolution of 1024x768, controlled by an IBM compatible computer running a purpose designed computer program. The average width of the equations in pixels was 187 (range 91–244) with an average height of 45 (range 26–59). The average width of the sub-expressions in pixels was 74 (range 25–111) with an average height of 23 (range 16–52).

Participants were given a statement of instructions before the experiment began. Practice items preceded the experimental items, and the participants took approximately fifteen minutes to complete the task. Progress was self-paced, with participants pressing the space bar to initiate the presentation of each trial.

Each item was presented in the centre of the monitor in the following sequence. First, a simple algebra equation was shown to the participant for 2500ms. The equation then disappeared and the screen remained blank for 1000ms. Then the sub-expression was shown, remaining on the screen until a response was made. The participant was required to decide whether the sub-expression was in that equation, responding via a timed selective button press. They pressed the green button, (the '/' key on the right side of the keyboard), to indicate that the sub-expression was part of the original equation, and the red button, (the 'Z' key on the left of the keyboard), to indicate that the sub-expression was not part of the original equation. Participants were instructed to respond as quickly as possible, while taking care not to make too many errors.

The response time recorded was the time between the onset of the sub-expression and the participant's response. After the response, the participant received feedback. If the re-

[1] Data from participants with an overall error rate of over 30%, or making in excess of 50% errors for any given sub-expression type, were excluded from the final analysis.

Table 1: Example equations and sub-expressions used in examining phrasal properties.

Equation	Sub-Expression		
	Phrasal	**Non-Phrasal**	**Incorrect**
$y = 8 + \frac{8y - 9x^2}{7y^6}$	$8y - 9x^2$	$8y - 9$	$8x + 9$
$3 - \frac{5}{7 - x^2(8 - y)}$	$(8 - y)$	$7 - x^2$	$2 - x^4$

sponse was correct then the word "CORRECT" appeared on the screen. Otherwise, the word "INCORRECT" appeared on the screen. In both cases, the participant's response time in milliseconds also appeared on the screen.

Data Treatment Two measures were employed to reduce the unwanted effects of outlying data points. Absolute upper and lower cut-offs were applied to response latencies, such that any response longer than 2500ms or shorter than 500ms was excluded from the response time data analysis and designated as an error. Secondly, standard deviation cut-offs were applied, so that any response time lying more than two standard deviations above or below a participant's overall mean response time was truncated to the value of the cut-off point.

It was necessary to exclude two items from the final analysis due to error rates in excess of 75%. One further item also had to be removed in order to balance the number of items in each version of the experiment. As a result, the final analyses were over twenty-four items per condition, not the original twenty-five. Response time and error data were analysed by a series of analyses of variance (ANOVAs), over both participant and item data. Where both the subject-based and item-based analyses were significant they were combined in the *minF'* statistic to ensure the generalisability of results over both these domains (Clark, 1973).

Results and Discussion

The mean correct response time (in milliseconds) and error rate for the three sub-expression types are summarised in Table 2, along with the corresponding standard deviations (in parentheses). Planned comparisons of the data were conducted using two-way ANOVAs (versions × sub-expression), carried out separately over subject and item data.

As expected, the participants performed significantly better for phrasal sub-expressions than for non-phrasal sub-expressions. This superior performance was seen in the response times with a 196ms recognition advantage (*minF'*(1, 66) = 33.06, $p < .01$). This advantage held for error rates also (*minF'*(1, 69) = 8.83, $p < .01$). This indicates that the equations are perceived in a way that allows for faster and more accurate recognition of phrasal sub-expressions than non-phrasal sub-expressions.

Table 2: Mean correct response times (ms) and error rates (%) as a function of sub-expression type for Experiment 1.

Sub-Expression	RT(ms)		%Error	
Phrasal	1153	(178)	14.8	(8.7)
Non-Phrasal	1349	(205)	25.2	(11.2)
Incorrect	1382	(246)	20.3	(9.5)

There was also a significant response time advantage for phrasal sub-expressions over incorrect sub-expressions (*minF'*(1, 57) = 35.97, $p < .01$). However, there was no corresponding overall advantage for error rates, despite the fact that the item-based analysis was significant (F_1(1, 21) = 4.21, p = .053, F_2(1, 69) = 5.46, $p < .05$). There was no significant difference between non-phrasal and incorrect sub-expressions for either response times or error rates.

The results of Experiment 1 provide support for our hypothesis that the internal representation used by experienced mathematicians is based on the phrasal structure of a parse tree. This comes from the logic of the experiment. Encoding of the equations significantly favours recognition of phrasal sub-expressions, indicating that knowledge of the constituent structure that underlies a parse tree is relied upon in the encoding process.

This outcome and those of previous work (Jansen et al., 1999) indicate that experienced mathematicians use an internal representation based on mathematical syntax and a parse tree structure. One interesting issue is just how much experience with mathematics is necessary before such representations become established. This is the focus of our second experiment.

Experiment 2

Our hypothesis here is that considerable experience is necessary before humans can parse an equation based on its mathematical syntax. To test this hypothesis, recognition tasks were designed to examine the influence of both syntactic well-formedness and phrasal properties in identifying sub-

Table 3: Example equations and sub-expressions used in examining well-formedness.

Equation	**Sub-Expression**		
	Well-Formed	**Non-Well-Formed**	**Incorrect**
$\frac{9}{x(2y-5)} - 4x^3$	$(2y-5)$	$\frac{}{x(2}$	$x(4y+$
$x = 6yx - \frac{2x+2}{x}$	$2x+2$	$= 6yx-$	$\frac{6x+2}{y}$

expressions of equations. The participants in these experiments were students in their first year of high school (Year 7). This year level was chosen because it is one year before algebra becomes a major component of their mathematics syllabus (in Australia). The students had been introduced to the notion of a variable, but had not been introduced to the exponent notation and had dealt only with very simple expressions.

Due to the complex nature of the equations (at least by Year 7 standards), we expect to see no significant performance advantages for one type of sub-expression over another, indicating that the internal representations of the students are not based on mathematical syntax or parse tree structures. However if any advantages are present, this would indicate a predisposition towards encoding equations into syntactically based constituent chunks, even with very little experience.

Method

Participants Eighteen participants successfully completed these experiments. All were Year 7 students, aged 12 to 13 years, with only limited knowledge of algebra. All participants were volunteers with normal or corrected-to-normal vision.

Materials and Design Experiment 2 consisted of two parts. Part A looked at syntactic well-formedness, the design of the experiment being similar to the experiment described in Jansen et al. (1999) which was conducted with competent adult mathematicians. Sixty equations were used, and sub-expressions of three types were generated for each.

a) A *well-formed sub-expression*, which is a component of its equation, and conveys the same meaning on its own that it conveys in the equation.

b) A *non-well-formed sub-expression*, which is also a component of its equation, but does not convey any coherent mathematical meaning on its own.

c) An *incorrect sub-expression*, which was not part of the original equation. It can be either well-formed or non-well-formed. These act as fillers.

See Table 3 for examples. The equations consisted of between twelve and fourteen characters, and each of the sub-expressions contained between four and six characters (the average for well-formed sub-expressions was 4.77; for non-well-formed, 4.50; for incorrect, 4.66). Only the variable names x and y were used, with at most one fraction being present in any equation.

Part B of Experiment 2 again examined phrasal properties, and was based on Experiment 1. Sixty equations were constructed, along with three sub-expressions per equation (phrasal, non-phrasal and incorrect). The properties of the equations and sub-expressions are the same as described in Experiment 1, with the sub-expressions again containing between four and six characters (the average for phrasal sub-expressions was 4.79; for non-phrasal, 4.50; for incorrect, 4.60). Table 1 contains examples of these. The equations used in part A and part B of this experiment were all different.

For each part of the experiment, three counterbalanced versions were created allowing the presentation of all three sub-expression types for each equation, but ensuring that participants were presented with each equation only once to avoid practice effects. For each version, there were twenty instances of each type of sub-expression. Two additional equations were constructed as practice items. The same practice items were used in each version. The items of each version were presented in a different pseudo-random order for each participant.

Participants did both parts of the experiment in the one sitting, one after the other. Due to the tasks in part A and part B being so similar, the order in which they were done was balanced over all of the participants. Thus half of the participants did part A before part B, with the other half doing the experiment in the reverse order.

Procedure The experiments were carried out in a quiet room, with groups of four or five students at a time. Each participant was seated in front of an IBM compatible computer with a 14" monitor, running at a resolution of 800x600. All items were black text on a white background, presented by a purpose designed computer program.

For part A, the average width of the equations in pixels was

178 (range 135–219) with an average height of 47 (range 26–61). The average width of the sub-expressions in pixels was 72 (range 39–179) with an average height of 26 (range 18–51). For part B, the average width of the equations in pixels was 187 (range 97–244) with an average height of 46 (range 26–59). The average width of the sub-expressions in pixels was 72 (range 25–111) with an average height of 24 (range 16–52).

The procedure for each part was very similar to that used for Experiment 1. There was no difference in the display timing of the stimuli or the response mechanism. However, since the students were not expected to perform very well in the task, they may lose confidence in performance if continually reminded of errors. Consequently, no feedback was given. Otherwise, the experimental procedure was the same. Participants were also given a brief rest period between the two parts and took approximately 25 minutes to complete the entire experiment.

Data Treatment To reduce the unwanted effects of outlying data points, absolute upper and lower cut-offs were applied to response latencies, such that any response longer than 4000ms or shorter than 500ms was designated as an error. The maximum cutoff time here is longer for the Year 7 students than for the experienced mathematicians in previous experiments.

As expected the participants did not perform well in this task, with the accuracy achieved for many sub-expression types being no better than chance. Given that many students were clearly guessing when presented with these sub-expressions, an analysis of response time data would be meaningless. Analysis was therefore only conducted on error rate data, by a series of analyses of variance (ANOVAs) over both participant and item data. Where these were significant, the were combined in the $minF'$ statistic. No participants data was excluded from the analysis.

Results and Discussion

The error rate for the three sub-expression types in part A (which examined well-formedness) is summarised in Table 4, along with the corresponding standard deviations (in parentheses). Planned comparisons of the data were conducted using two-way ANOVAs (versions × sub-expression), carried out separately over subject and item data.

Table 4: Error rates (%) as a function of sub-expression type for Experiment 2A.

Sub-Expression	%Error	
Well-Formed	36.1	(21.5)
Non-Well-Formed	51.7	(14.9)
Incorrect	50.6	(19.9)

The results of interest are the performance differences between well-formed and non-well-formed sub-expressions. Participants performed significantly better in recognizing well-formed sub-expressions than their non-well-formed counterparts with an advantage of 15.6% ($minF'(1, 37) = 7.50$, $p < .01$). In fact, since in each trial participants had a 50–50 chance of success, it is clear that for both non-well-formed and incorrect sub-expressions, participants were doing no better than random guessing. It is only for well-formed sub-expressions that they were performing better than chance.

Table 5 summarises the error rate data for the three sub-expression types in part B of the experiment (which examined phrasal properties), along with the corresponding standard deviations (in parentheses).

Table 5: Error rates (%) as a function of sub-expression type for Experiment 2B.

Sub-Expression	%Error	
Phrasal	34.1	(15.4)
Non-Phrasal	53.6	(16.5)
Incorrect	49.4	(22.9)

The results show a significant 19.4% error rate advantage for phrasal sub-expressions over non-phrasal sub-expressions ($minF'(1, 59) = 12.07$, $p < .01$). As in part A, the incorrect and also the non-phrasal results indicate that participants are doing no better than chance in responding to these sub-expression types. However, performance was clearly above chance for phrasal sub-expressions.

Given the limited mathematical experience of the Year 7 students, these results are unexpected. The fact that the overall accuracy of the Year 7 students is far lower than for competent adult mathematicians, indicates that the development of their internal representation still has a long way to go. However, superior performance in recognizing syntactically well-formed and phrasal sub-expressions provides support for the notion that mathematical syntax plays an important role in the way that these students encode equations. This result therefore does not support our hypothesis that considerable experience is necessary before students can parse an equation based on its mathematical syntax.

Despite the significance of these results, it is not clear whether the students represent a heterogeneous or a homogeneous population with respect to their performances in this task. Therefore, a further analysis of the error rate data from this experiment was conducted. For part A of the experiment, a three-way split was carried out based on the difference in accuracy in recognizing well-formed and non-well-formed sub-expressions. The results of participants in each version were divided into three groups. The top group contained participants with the greatest performance advantage in recognizing well-formed sub-expressions over non-well-formed sub-expressions. The bottom group contained those with the least advantage, or possibly even a disadvantage in recognizing well-formed sub-expressions over their non-well-formed counterparts. The remaining participants formed a middle group, but the results of this group were not of interest. Since there were six participants per version, each group contained the results of two participants from each version. ANOVAs were then conducted to compare the performance of the top and bottom group.

As expected, an even greater performance advantage of 32.5% in identifying well-formed over non-well-formed sub-expressions was found in the top group ($minF'(1, 30) = 19.55$, $p < .01$). However, the performance of the bottom group revealed a slight disadvantage of 1.7% in recognizing well-formed sub-expressions over their non-well-formed counterparts. This result was not statistically significant ($F < 1$ for analysis by both subject and item).

A similar analysis was conducted for part B of the experiment, with the three way split based on the accuracy difference between recognizing phrasal and non-phrasal sub-expressions. The top and bottom groups reflect the participants with the greatest and least performance advantage respectively, in recognizing phrasal sub-expressions over non-phrasal sub-expressions. The top group again had a significant performance advantage of 32.7% in identifying phrasal over non-phrasal sub-expressions ($minF'(1, 60) = 20.17$, $p < .01$). For the bottom group however, the advantage was only 5.8% which was not statistically significant ($F < 1$ for analysis by both subject and item).

This result indicates that within the population sample for Experiment 2, there are two distinct groups, one of students who encode equations based on mathematical syntax, and one of those who appear not to. One possible explanation for this result is that some students have more previous experience with algebra and mathematics than others. However, another possibility is that some students might have a stronger predisposition for using knowledge of mathematical syntax to guide construction of internal representations. Certainly more research will be needed before the cause of this result can be resolved.

Conclusions

Previous research has suggested that adults competent in mathematics encode equations into constituents that have syntactically well-formed structure (Jansen et al., 1999). We have extended upon these results by providing support for the hypothesis that the internal representation used by mathematicians is based on the constituent structure of a parse tree. Evidence has also been presented which indicates that this encoding mechanism is present in young students. This result is surprising given that the students have very little experience in dealing with complex algebraic expressions.

The future direction of this research is to further investigate the encoding mechanisms and internal representations used to process equations, and in particular to examine how the representations of equations are used in mathematical problem solving. Also, the positive result with the Year 7 students leads to the question of just how little mathematical experience is necessary before mathematical syntax begins to play a role in encoding equations. Whether or not the students are establishing representations based on mathematical syntax, or their performance reflects a more general encoding mechanism for such complex stimuli, can only be resolved by conducting similar experiments with children who have no experience with algebraic equations.

Acknowledgements

The authors wish to thank the staff and students at the Prahran Campus of Wesley College in Victoria, Australia, for their helpful co-operation and assistance in conducting the experiments involving the Year 7 students.

References

Akmajian, A., Demers, R.A., & Harnish, R.M. (1984). *Linguistics: An Introduction to Language and Communication* (2nd ed.). Massachusetts: MIT Press.

Anderson, R.H. (1977). Two-dimensional mathematical notation. In K.S. Fu (Ed.), *Syntactic Pattern Recognition Applications*. New York: Springer-Verlag.

Clark, H.H. (1973). The language-as-fixed-effect fallacy: A critique of language statistics in psychological research. *Journal of Verbal Learning and Verbal Behavior, 12*, 335–359.

Jansen, A.R., Marriott, K., & Yelland, G.W. (1999). Perceiving structure in mathematical expressions. In M. Hahn & S.C. Stoness (Eds.), *Proceedings of the twenty first annual conference of the cognitive science society*. Lawrence Erlbaum Associates.

Johnson, N.F. (1968). The influence of grammatical units on learning. *Journal of Verbal Learning and Verbal Behavior, 7*, 236–240.

Johnson, N.F. (1970). Chunking and organization in the process of recall. In G.H. Bower (Ed.), *The Psychology of Learning and Motivation*, (Vol. 4). New York: Academic Press.

Ranney, M. (1987). The role of structural context in perception: Syntax in the recognition of algebraic expressions. *Memory and Cognition, 15*(1), 29–41.