In J. Bruner (Ed.), Learning about Learning (a Conference Report). Washington: U. S. Government Printing Office, 1966.

THE PSYCHOLOGY OF ARITHMETIC[9]

Patrick Suppes

The psychological aspects of arithmetic I want to discuss I shall place under four main headings. I do not think that these four headings are particularly the best four. Certainly they are neither exhaustive nor mutually exclusive. I will use them for purposes of somewhat arbitrary classification. I shall begin with problems of meaning in arithmetic; then turn to the psychology of algorithms; followed by discussion of problems of the sequence in which concepts should be introduced; and I end with a discussion of what behavioral account is to be given of the function of responses that correspond in ordinary talk to giving reasons for an answer.

Meaning in Arithmetic

It seems natural to say that the first problem of meaning in arithmetic is to provide an analysis of that fragment of arithmetical language which the child already possesses when he enters school.

The emphasis in the discussion of children's language by linguists in recent years has been almost entirely on the grammar of their language, but it is fair to say that the analysis of the grammar of the child's use of quantitative terms like "one" or "two" does not take us very far in attempting to understand what the child knows about arithmetic as he enters school. All of us are impressed by what the linguists have shown us about the ability of children to make grammatical transformations of a startlingly complex nature even at the age of 3 or 4 years. I am sure we shall be equally impressed with what they are able to show about the child's grasp of the semantics of his language when they turn to this problem as well. I suppose that what I want to say is that it is just as surprising that children learn the meaning of verbs as it is that they learn such things as how to transform from the active to the passive voice.

Many children who enter the first grade without any explicit

[9] This is a preliminary draft of the first two parts of this paper. I intend for the final version to be combined with my earlier working paper, "Towards a Behavioral Psychology of Mathematical Thinking."

training in arithmetic in kindergarten or nursery school are able to count at least to 6 or 7. To say that they are able to count to 6 or 7 does not really tell us what the counting words mean to the children. A deeper and more precise specification of this meaning seems to be an appropriate problem for additional research.

It will be useful to examine one sort of example in more detail in order to indicate some of the problems involved. To begin with, many children between the ages of 4 and 5 years learn the names of numbers used in counting from 1 to 6 or 7, and are able to repeat these names verbally in the right sequence with considerable ease. Often they are able to do this before they are able to count a set of cardinality 6 or 7. On the simple supposition that they first learn the names of the numbers and how to recite them in sequence, we might be inclined to accept the hypothesis that the number names are first learned primarily as ordinal counting names, and only later as names for the cardinal numbers of sets. Unfortunately, any such simple theory of the acquisition of the meaning of the number names is not correct. It is a commonly observed phenomenon that 4-year-old children who are not yet able or who do not yet seem to know how to reproduce the number name "four" will tell you that the cardinality of a set of four objects is "two and two," or that the cardinality of a set of five objects is "three and two." It would, I think, be a useful enterprise to attempt to infer the kind of meaning children attach to number names by a detailed examination of the uses of these number names in their preschool language.

Algorithms in Arithmetic

As an initial model for thinking about algorithms, I would like to propose the following. We have in mind a given collection of problems that we wish the child to be able to solve. To make our anaysis definite at this point, let us consider a set of arithmetical problems. They might be in the form of $8-5=\_\_$, $8+\_\_=10$, $10-\_\_=4$, $\_\_-3=5$, etc. The machinery needed to solve these problems can be roughly divided into two parts. One part consists of direct storage in memory of certain elementary facts. Exactly what these elementary facts are will vary from stage to stage in the curriculum. Towards the beginning of arithmetic, it might consist of storage of the elementary addition facts: $1+1=2$, $1+2=3$, $2+1=3$, $1+0=1$, $2+0=2$, $3+0=3$, $0+3=3$, etc. The second part of the machinery consists of algorithms, or constructive rules, for transforming the elementary facts in memory into new elementary facts, or what is probably more important, for transforming new stimulus presentations into one of these elementary facts stored in memory.

An immediate problem of psychological importance with respect to a given body of problems is how much should be stored in memory and how much should be carried by the algorithmic rule. It is seldom the case that for a given set of problems we want all the answers stored directly in memory—it is certainly contrary to the usual spirit in teaching mathematics, but it is also unusual to want to store in memory only a minimal set of facts. For illustrative purposes, let me describe in some detail a way of teaching arithmetic that would consist of storing in memory a small number of facts and transferring the larger part of the load to the algorithmic rules. I emphasize that the example chosen is not one that is meant to have direct pedagogical applications. This system for computing sums is clearly not the sort of system we would wish to teach.

Let us suppose that our set of problems is just the following 21:

$1+1=n$	$1+n=2$	$n+1=2$
$2+1=n$	$2+n=3$	$n+1=3$
$1+2=n$	$1+n=3$	$n+2=3$
$3+1=n$	$3+n=4$	$n+1=4$
$1+3=n$	$1+n=4$	$n+3=4$
$2+2=n$	$2+n=4$	$n+2=4$
$4+1=n$	$4+n=5$	$n+1=5$

We put the following four facts in memory:

$$1+1=2$$
$$2+1=3$$
$$3+1=4$$
$$4+1=5$$

We have the following four rules of operation:

1. Use the four facts in memory to replace equals by equals.
2. Replace a term of the form $a+(b+c)$ by $(a+b)+c$, or vice versa.
3. Replace a term of the form $a+b$ by $b+a$.
4. Cancel an equation of the form $a+n=a+c$ to get $n=c$.

These four rules are then used to transform a problem, step by step, until we reach an expression of the form $n=c$. Thus,

$2+2=n$	Problem
$2+(1+1)=n$	by (1)
$(2+1)+1=n$	by (2)
$3+1=n$	by (1)
$4=n$	by (1)
$3+n=5$	Problem

or similarly, $3 + n = 4 + 1$	by (1)
$3 + n = (3 + 1) + 1$	by (1)
$3 + n = 3 + (1 + 1)$	by (2)
$3 + n = 3 + 2$	by (1)
$n = 2$	by (4)

There are several immediate criticisms to be made of this setup, as I have described it. First, I have not been really explicit about parentheses in connection with rule 1. And I have not really made clear the role of the associative law, i.e., rule 2. More importantly, I have not written down a genuine algorithm for the set of problems. The four rules are four rules of proof, not an algorithm for solving any one of the 30 problems.

To convert the four rules into an algorithm, it is necessary to specify an order in which they are to be applied, and this order, to be efficient, should vary with the particular problem. Not only is it necessary to specify an order, but it is also necessary to show that the algorithm can be given to a machine and automatically used to solve any of the 30 problems.

To convert the present four rules into a genuine algorithm is somewhat tedious. Let me describe another, simpler system that may be used to solve the same 21 problems.

We put in memory the following five definitions:

$1 = /$
$2 = //$
$3 = ///$
$4 = ////$
$5 = /////$

Our algorithm is then the following:

(1) Replace all Arabic numerals by their stroke definitions and delete all plus symbols.

(2) If there are strokes on both sides of the equal sign, cancel one-by-one starting from the left of each side until there remain no strokes on one side. Ignore n in canceling.

(3) On the one side still having strokes, replace the strokes by an Arabic numeral, using the definitions in memory.

The solution in the form n = c or c = n will result.

Let us apply this algorithm to the two problems previously considered. First problem:

$2 + 2 = n$	Problem
$// \quad // = n$	by (1)
$4 = n$	by (3)

In this case no canceling is required.

Second problem:

$3 + n = 5$	Problem
$/// \quad n = /////$	by (1)
$// \quad n = ////$	by (2)
$/ \quad n = ///$	by (2)
$n = //$	by (2)
$n = 2$	by (3)

It should be clear from these examples how the algorithm may be applied to solve the other 19 problems in the original set, and moreover, how simply by adding new definitions in memory we may, without changing the algorithm, move on to similar problems involving larger numbers.

From a logical standpoint this algorithm is perhaps as simple as any to be found—here I shall not try to justify this statement by attempting a definition of logical simplicity. As far as I know, however, this algorithm has not been used in any first-grade arithmetic program.

Consideration of its possible use by children takes us out of the domain of elementary mathematics—the theory of algorithms for simple mathematical systems—into the domain of behavioral psychology. Once we enter this domain there are so many things to be said, I hardly know where to begin. Let me try to state some of the problems.

1. It seems highly unlikely that any children, without training, actually use the algorithms just described. The perplexing question is, what algorithms do they in fact use? At the level at which this problem is often discussed, the obvious answer is that they use the algorithms taught in the classroom and presented in their textbooks. But even casual inspection of the curriculum shows the inadequacy of this response, for algorithms for the 21 problems listed above (or with the numerical variable n replaced by a blank or box) are not explicitly taught, although some partial hints in terms of counting may be given. A typical curriculum instruction to teachers is to let the children find the answer "intuitively" by working with the numbers. Parenthetically, the use of the word "intuition" in its nominal, adjectival, or adverbial form by a curriculum builder, reformer, planner, or evaluator should be a signal to the behavioral psychologist that unexplained and ill-understood learning behavior is about to be mentioned and, unfortunately, often described as if it were understood.

So the problem remains, How do children in the fourth, fifth, or sixth month of the first grade solve problems like those in our set of 21?

2. A proposal often heard is that children solve such problems by simple rote learning. This is a possible response when any single

set of 20 or 30 simple problems is considered. It does not seem nearly as plausible when we look at the larger set of problems from which our 21 have been drawn. There are 55 ordered pairs of numbers summing to 9 or less ($0+0=0$, $0+1=1$, $1+0=1$, etc.). There are, then, 165 problems of the same type as our 21 ($n+0=0$, $0+n=0$, $0+0=n$, etc.). And the number of problems is increased further by adding triplets of the form $1+2+n=4$, $1+n+2=4$, etc. It is extremely doubtful that this large stock of problems is held in memory, available for direct access. The child solves them by applying some sort of algorithm. Some of the possibilities are the following.

(a) The child counts off the necessary number names, aloud or in silent speech. Thus, the solution to "$4+5=n$" is obtained by counting off five number names after "four," namely "five, six, seven, eight, nine." The solution to "$4+n=9$" is obtained by counting off number names after "four" until "nine" is reached and then judging the cardinality of the set of number names counted off. Even without detailed analysis it is clear that the second kind of problem is harder than the first. The third kind of problem is still harder. The solution to "$n+5=9$" is obtained by counting off enough number names such that five more take the child to "nine." It seems doubtful to me that the algorithm can be successfully applied in this form to the third kind of problem. Notice that no advantage has been taken of the commutativity of addition. Serious training on this problem would enable the child to reduce problems of the third kind to those of the second kind. The relatively greater difficulty that almost all first-grade children have with the third kind of problem, when the unknown is at the far left, indicates that, if the algorithm just described is used, it is not augmented by the commutative law.

For a great many different reasons it seems improbable that the algorithms actually used require very many closely knit steps to obtain an answer. The counting algorithm just described is realistic for problems of the form "$4+5=n$" and not out of the question for problems of the form "$4+n=9$." For problems of the form "$n+5=9$" the child may, without being explicitly conscious of it, make rough estimations of n and test the guess by counting. He remembers, say, that $5+5=10$, and "nine" is close to "ten," so he tries 3 or 4. Or, he may remember, that is, have in immediate storage, that $4+4=8$, and he uses this fact to guess 3, 4, or 5.

(b) In many ways the above discussion sells the counting algorithm short, because counting a set of number names like "five, six, seven, eight" aloud or in silent speech, seems difficult—the memory of "five" may have departed before "eight" is said. When the algorithm is externalized and applied in terms of physical objects (even the fingers) it seems much easier. I have seen something like

the following used quite successfully in Ghana with harder problems than those we are now discussing.

The child has a counting set of pebbles on his desk. To solve the problem "$4+5=n$" he first counts out 4 pebbles from his pile. He stops, and then counts out five more. This counting is done by simultaneously saying the number names "one, two, three, four" and pulling one pebble from the pile as he says each name. After counting out the set of four and then counting out the set of five, he then counts the separated set of nine pebbles and gets the answer. He solves the problem "$4+n=9$," by first counting out a set of nine pebbles and then taking 4 away, that is, by counting off a set of four from the set of nine. (It is to be emphasized that each of these counting operations is a highly physical thing.) After taking away the set of four, he then counts the remaining set of five to obtain the answer. Notice that the act of taking away four from the set of nine pebbles can be clearly and succinctly taught even though the subtraction symbol has not been introduced. As already remarked, lots of people have observed that for American children the "$n+5=9$" sort of problem is harder than the "$4+n=9$" sort. For the counting algorithms just described they would seem to be on an equal footing. I think, but do not have real evidence at hand, that the Ghanaian children have the same sort of relative difficulty. The explanation is most likely to be found in the decoding required to pass from the written problem to the physical execution of the algorithm. The detailed analysis of how the stimulus arrangement expressing the problem sets off the algorithm will not be discussed here, but I may say in passing that this kind of example provides an excellent opportunity to analyze the behavioral semantics of the simplest sort of language. Briefly put, I interpret a problem format like "$4+n=9$" as a command in the imperative mood. The symbol "9" standing by itself to the right of the equal sign means for the pebble model "Count out a set of nine pebbles." The symbol "4" means "Count out a set of four pebbles from the set of nine." And, roughly speaking, the remaining phrase "$+n$" means "Count the remaining set of pebbles and record the answer." For this kind of semantic the classical notion of truth is replaced by that of a response, or class of responses, satisfying a command. What I have sketched here in the roughest sort of way can be made precise by using with only slight modification the standard methods and concepts of formal semantics.

From the standpoint of the usual way of characterizing algorithms, the pebble-counting algorithm is unusual, for the operations of the algorithm are performed on the pebbles and not on the number symbols themselves. In this case the number symbols have meaning and this meaning is used to give instructions for performing the

algorithm. It would seem that it is this sort of algorithm that many people now advocate in arithmetic in order to avoid the pupils' development of great facility with algorithms defined wholly in terms of the number symbols, since these may be learned without any "understanding of numbers."

(3) It is familiar talk to characterize the part of mathematics that may be reduced to algorithms as trivial. The classification of mathematical systems or theories as having or not having an algorithmic solution has been an important area of research in mathematical logic. What is not adequately recognized is that no adequate account of mathematical thinking can be given in terms of non-algorithmic concepts like that of mathematical proof. Any adequate psychological account of mathematical thinking must surely be aimed mainly at finding the higher-order algorithms the student uses in finding a proof. What I had to say earlier about the study of association processes in the context of discovering proofs is directed to this point.

Mathematicians may not like the claim that the "creative act" of discovering proofs is itself an algorithmic matter. The only alternative seems to be a retreat into the mystical land of intuition.