Learning Recursive Procedures by Middleschool Children[1]

Yuichiro Anzai
Carnegie-Mellon University & Keio University

and

Yuzuru Uesato
Keio University

Introduction

Recursion is a recurrent theme in human thinking. It has been around for a long time in some fields related to cognitive science: for instance, it has taken place in information-processing models of cognition, in the theory of computation, in cognitive and developmental psychology, or in teaching computer programming to novices.

Intuitively, recursive formulation may lead to understanding of potentially infinite phenomena in compact, finite terms. On the other hand, since recursive definition involves top-down, tightly connected organization of knowledge, it may not be easy to learn, or to be applied to formulation of complex problems. These expectations, however, are less well examined experimentally. Besides, there are some other points such as memory load for executing recursive procedures, the firmly established character of recursive functions in the theory of mathematics, or practical application to teaching computer programming, which make recursion an interesting theme for cognitive science. As one topic related to recursion, this paper discusses the question of whether recursive procedures are cognitively difficult to learn, based on a rule induction experiment conducted on middleschool children. It concludes that recursive procedures may be acquired based on learning of the corresponding iterative procedures.

Learning Recursive Procedures

A recursive function treated here is simply a function whose definition includes the function itself. As a simple but representative example, we use exclusively in this paper the factorial function "*fact*" defined on *N*, the set of positive integers, as follows:

$$fact(n) = fact(n\text{-}1) \times n \text{ for any } n \in N,\ n>1, \quad \text{and } fact(1) = 1.$$

The above definition is recursive, but of course *fact* can be defined iteratively:

$$fact(n) = 1 \times 2 \times \ldots \times n \quad \text{for any } n \in N.$$

The above two kinds of definitions are functionally equivalent, but have many cognitively different points. Let us consider below only the point relevant here: how people acquire the recursive procedure for computing factorials, based on example data. First, suppose that a student is given an iterative sequence of data for factorials:

$$fact(1) = 1 \quad fact(2) = 1 \times 2 \quad fact(3) = 1 \times 2 \times 3.$$

It may be easy for him to generalize the above simple patterned sequence, and to obtain the general iterative definition, $fact(n) = 1 \times 2 \times \ldots \times n$ ($n \in N$). Note that the induced definition itself can easily be interpreted to provide procedures (multiplications) for actual computation.

On the other hand, suppose that the student tries to induce the factorial function based on the following recursively generated data:

$$fact(3) = fact(2) \times 3 \quad fact(2) = fact(1) \times 2 \quad fact(1) = 1.$$

In this case, although the data, if regarded declarative, can be generalized formally to generate $fact(n) = fact(n\text{-}1) \times n$ ($n \in N$), the student needs to consider all the subformulas, $fact(k) = fact(k\text{-}1) \times k$ ($k = 2,\ldots,n\text{-}1$), to actually compute $fact(n)$: the data allow direct generalization by converting, for example, 3 to n and 2 to n-1, but he is necessary to organize the given segments of data to acquire the recursive computational procedure. It may be much more difficult than in the iterative case.

However, we can advance our speculation one more step. The student, while he is engaged in the task of inducing the factorial from the iterative data, might notice the regularity of embedded pattern in the data. The left column of Fig. 1 illustrates it for an iterative data set. If this kind of structural embedding was discovered, acquisition of the iterative definition of the factorial may result in learning the nested procedural structure of the factorial. Then, if the nested structure as shown in the left of Fig. 1 resides in memory, and if recursive data are presented, the data may match the nested structure fairly easily as shown in Fig. 1. Thus, the recursive procedure may be learned by the successive presentation of the iterative and recursive data sets in this order.

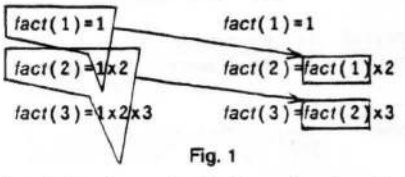

Fig. 1

Nested structure embedded in an iterative data set and its relation to the corresponding recursive data set

The preceding simple discussion gives us the hypothesis that, if a student knew none of the factorial function, or the concept of recursion, he finds it easier to learn the iterative procedure for the factorial rather than the recursive one, but after he learned it, he must already be ready to assimilate the recursive procedure. In the following rule induction experiment, we examine this hypothesis by using middleschool children.

Experiment

Subjects and procedure

88 middleschool children (age about 14) participated in the experiment. The rule to be induced was the numerical function for computing factorials of positive integers. Two kinds of formats for example data were considered. One was the iterative format, and the corresponding to-be-induced function was called WHITE in the experiment. The other was the recursive format, and the corresponding function was named BLACK.

For each format, a sequence of three data sets was prepared. The first data sets for WHITE and BLACK were given as follows:

First data set for WHITE

Let us think about the following computation for a given number. The answer to the computation is called "WHITE" For example, "WHITE of 2" is computed as follows:

(1) Start with 1.

(2) Multiply 1 by 2. The result is 2.

[1]Thanks are due to John Anderson, Robin Jeffries and Herbert Simon for their comments on this work. Please address correspondence to Yuichiro Anzai, Faculty of Science and Technology, Keio University, 3-14-1, Hiyoshi, Kohoku, Yokohama, Japan.

"WHITE of 2" is 2.

Now, compute "WHITE of 4". (Write the computation and the answer.)

First data set for BLACK

Let us think about the following computation for a given number. The answer to the computation is called "BLACK". For example, "BLACK of 2" is computed as follows:

(1) "BLACK of 2" is "BLACK of 1" multiplied by 2.

(2) "BLACK of 1" is 1.

"BLACK of 2" is 2.

Now, compute "BLACK of 4". (Write the computation and the answer.)

In each of the above data sets, two segments of information, *(1)* and (2), for the factorial of 2, and the value of it were supplied. There was provided a problem at the last line, which was to compute the factorial of 4. If a subject gave the correct answer to the problem, then he was considered to have acquired a factorial-computing procedure, iterative or recursive, depending on which data set, WHITE or BLACK, was presented to him.

The second data sets for WHITE and BLACK included three segments of information, and were designed as shown below:

Second data set for WHITE

Let us think about the following computation for a given number. The answer to the computation is called "WHITE". For example, "WHITE of 3" is computed as follows:

(1) Start with 1.

(2) Multiply 1 by 2. The result is 2.

(3) Multiply 2 by 3. The result is 6.

"WHITE of 3" is 6.

Now, compute "WHITE of 5". (Write the computation and the answer.)

Second data set for BLACK

Let us think about the following computation for a given number. The answer to the computation is called "BLACK". For example, "BLACK of 3" is computed as follows:

(1) "BLACK of 3" is "BLACK of 2" multiplied by 3.

(2) "BLACK of 2" is "BLACK of 1" multiplied by 2.

(3) "BLACK of 1" is 1.

"BLACK of 3" is 6.

Now, compute "BLACK of 5" (Write the computation and the answer.)

The third data sets, each of which contained four segments of information, were defined in a similar manner.

The subjects were divided into two groups called G1 ($n = 45$) and G2 ($n = 43$). The group G1 was given the data in the order of W-1, W-2, W-3, B-1, B-2 and B-3, where W-*i* and B-*i* denote the *i*-th data set for WHITE and BLACK respectively. On the other hand, G2 was given the data in the order of B-1, B-2, B-3, W-1, W-2 and W-3. Both groups were given five minutes for each data set, which were ample enough for middleschool children. The data sheets were collected from the subjects for each data set, and no direct feedback of answers was given.

Results and discussion

The results are tabulated in Table 1. The more data sets presented, the greater number of subjects who answered correctly, both for WHITE and BLACK. The percent correct was larger for G1's WHITE (60% for the third set) than for G2's BLACK (33% for the third set), but even the latter gave fairly good performance. Also, if the data for BLACK were presented after WHITE as for the group G1, the performance was better than its opposite: G1 for BLACK gave 16%, 29% and 64% of percent correct for the data sets with two, three and four segments of information, but G2 for BLACK provided 0%, 14% and 33%, which were relatively smaller. On the other hand, the performace for WHITE was similar for the two groups, regardless of the order of presentation.

The result is thus generally in agree with our expectations. It was easier for the children to have worked on the iteratively generated data sets, but acquisition of the recursive procedure was facilitated by learning the iterative one.

Also, note that the WHITE data for G1 and G2 show a similar tendency, and the BLACK data for the two groups provide a *different sort of* similar tendency: the rate of increase of the percent correct decreased for the WHITE data with respect to the number of presented data sets, while it increased for the BLACK data. This particular trend may have reflected the subjects' relative difficulty in discovering regularity in a small number of information segments in a recursive data set.

Table 1

Percent correct for the induction experiment

	G1		G2	
Data set no.	WHITE	BLACK	BLACK	WHITE
1	11(%)	16	0	9
2	42	29	14	30
3	60	64	33	47
No. of subjects	45		43	

(For almost all the subjects, if a subject gave the correct answer for the *j*-th data set, he was also correct for all the *i*-th sets, where $1 \leq i < j$.)

Thus, we think that recursive computation may be apparently difficult for children to learn, but also that it may be acquired by inducing the nested structure, and interpreting it as a procedure, based on the recursive data. Let us provide one possible mechanism that generates the gross characteristics of the experimental results, which is essentially similar to the one briefly described in the previous section. Suppose that the third data sets for WHITE and BLACK given in the experiment were represented as follows:

WHITE	BLACK
(equal (times 1 2) 2)	(equal (black 4) (times (black 3) 4))
(equal (times 2 3) 6)	(equal (black 3) (times (black 2) 3))
(equal (times 6 4) 24)	(equal (black 2) (times (black 1) 2))
(equal (white 4) 24)	(equal (black 1) 1).

Assume that, successively embedding the segments in the WHITE data set, we obtained the nested formula:

(equal (white 4) (times (times (times 1 2) 3) 4)).

Note that, if we identify (times (times 1 2) 3) with (black 3), and also identify "white" with "black", then the formula matches the first segment in the above BLACK set:

(equal (black 4) (times (black 3) 4)).

This kind of correspondence holds also for the first and second data sets. Generalization at this point, which yields the correspondence between (times (times (... (times (times 1 2) 3) ...) *n*-1) *n*) and (black *n*), provides the procedural basis for the recursive definition of the factorial function, which is based on nested arithmetic calculation.

Discussion

The relation between conceptual and procedural understanding in problem solving has raised many issues complex but central for cognitive science. At some deeper level of understanding, a person can both handle with knowledge procedurally, and appreciate it declaratively. Recursion provides a simple example for this matter: since it is usually formulated in a compact form, its declarative representation may be simpler than the corresponding iterative form. But such declarative representation must be accompanied by procedural knowledge for actual computation, and this knowledge might be cognitively complex. The argument presented in this paper suggests that such knowledge can be acquired not directly, but by working on iterative data.

An example of the process of learning a recursive strategy by discovering a nested structure in knowledge of results obtained by weaker, nonrecursive strategies was presented in Anzai & Simon (1979). The strategy acquisition process reported there is essentially similar to the recursion learning process discussed in this paper: the thesis shared by the two studies is that complex recursive procedures for solving a problem may be acquired by working on the problem, using already available, nonrecursive knowledge.

Which way of learning, by discovery or by instruction, is better has long been a controversial problem in instructional psychology. Learning by doing, which is along the line discussed here and in Anzai & Simon, is basically a process of learning by discovery. In this regard, as suggested in this paper, recursive procedures may be learned by discovery. Recursive computation may be intrinsically more difficult than iterative one, since execution of recursive procedures may require more memory resources. But it does not mean that they can not be acquired by discovery.

However, of course we do not deny the possibility of learning recursive procedures by top-down instruction. The two ways of learning are actually complementary in the real world, and both ways may play important and intertwined roles. Also, we should be cautious when we try to extend the consideration to more complex domains such as computer programming. It is because a complex task necessarily involves many different cognitive subprocesses, and it is not always easy to extract from them only the part played by recursion.

References

Anzai, Y. & Simon, H. A. 1979 The theory of learning by doing. *Psychological Review*, **86**, 124-140.