Assessing Behavior Potential: Adequacy of Basic Theoretical Assumptions†

JOSEPH M. SCANDURA and
JOHN H. DURNIN
University of Pennsylvania, U.S.A.

INTRODUCTION

General background

Up until about three years ago, the project director and his students were primarily engaged in research on rule learning and with the development of a new scientific language for formulating research in the area. It was called a scientific language, rather than a theory, since its development required relatively little in the way of formulating and testing psychological hypotheses. The emphasis was mainly on developing a precise language with which to talk about the phenomena involved. This earlier work has been reported at some length in the literature (e.g., Scandura, 1968, 1969a, 1969b) and is not reviewed here.

One basic aspect of this research which kept recurring over and over involved the tendency of the experimental Ss to respond in a consistent and highly predictable manner. This occurred with many different kinds of tasks and a wide variety of Ss. Another was the importance of knowing very precisely what relevant capabilities the S had when he entered into the learning or behaving situation. Even where the tasks were novel, individual Ss developed ways of attacking problems which if not taken into account would have made the data inconclusive and often uninterpretable. Others who have

†This article was first reproduced during November, 1971 as technical report 62 in the structural learning series and is published here in its original form. It is based on research supported by Bureau of Research, U.S. Office of Education. An abridged version was published in Scandura, J. M. Structural Learning: I. Theory and Research. New York: Gordon and Breach, 1973, Chapter 7. Subsequent research, including some important extensions and theoretical refinements, is reported in Scandura, J. M., Problem Solving: A structural/process approach with instructional implications. N.Y.: Academic Press, 1977.
The problem

The research reported here was concerned primarily with the problem of how to assess the behavior potential of individual Ss. More specifically, given a class of rule-governed (RG) behaviors the aim was to propose and test a new technique for identifying those behaviors in the class which individual Ss were capable of and those which they were not. This technique was based on what might be called memory-free theorizing. Memory-free theorizing is concerned with questions of performance, learning, and problem-solving in situations where the S is not encumbered by permanent memory or by his limited capacity for processing information. Its goal is to introduce psychological mechanisms which explain learning and performance associated with structured knowledge under ideal conditions.

The potential value of such theorizing is two-fold. First, there are many practical situations in which memory and information processing capacity play a minimal role. These factors can frequently be ignored, for example, in dealing with familiar tasks, such as performing simple arithmetical computations. (Even where the numbers are large, man-machine systems of various sorts can extend the effective information processing capacity of man almost indefinitely.) Secondly, such theorizing may provide a useful beginning from which other theorizing may proceed (cf. Scandura, 1971b). That is, by concentrating on simpler (but not trivial) phenomena, and understanding them first, it may be possible to gain important insights into less restricted kinds of behavior as well (e.g., involving memory).

Scandura first became concerned with the problem of assessing behavior potential while doing his dissertation. Later, during the summer of 1962, Greeno and Scandura (1966) found in an experiment on verbal concept learning that Ss either give the correct response the first time they see a transfer stimulus or the transfer stimulus is learned as its control. The thought occurred to Scandura (1966) that if transfer obtains on the first trial, if at all, then responses to additional transfer items, at least under certain conditions, should be contingent on the response given to the first transfer stimulus. In effect, a first transfer stimulus could serve as a test to determine what had been learned during the original training, thereby
making it possible to predict what response S would give to a second transfer stimulus.

Since that time, the project director and his students have collected a fairly substantial body of data which provide strong support for this contention (Scandura, 1966, 1967, 1969a; Scandura, Woodward and Lee, 1967; Scandura and Durnin, 1968; Roughhead and Scandura, 1968). In these studies the Ss were presented with a number of instances of a rule and told to learn the instances as efficiently as they could. After learning, the Ss were presented with a test stimulus and instructed to respond on the basis of what they had just learned. They were told they were correct no matter what the response. Then, they were presented a second test stimulus.

The results were surprisingly clear. Whenever the response given by S to the first test stimulus was in accord with a particular given or derived rule, the response to the second test stimulus was almost always also in accord. Second test behavior has been predicted with anywhere between 80 and 95 percent accuracy.

The results of one study (Scandura and Durnin, 1968) on extra-scope transfer further suggest that "what is learned" can often be determined by the appropriate selection of just two test instances. In particular, Scandura and Durnin found that successful performance with two stimuli which differ along one or more familiar dimensions implies successful performance with other stimuli which differ only along these dimensions. This result suggests that success on two instances which differ simultaneously along all possible dimensions involved in a rule may be adequate to define what rule is learned in an essentially unique fashion. The question remains, of course, as to what is meant by a well-learned dimension.

In general, it would appear that when S thinks he is right and the new situation remains relevant, he will continue to respond in a consistent manner.

As a first guess it was originally thought that the entire notion of response consistency involved capitalizing on Einstellung to ascertain what is learned and to predict performance on future items. As it turns out, this is incorrect. The hypothesis is too restrictive. It does not appear necessary to assume that S will continue to use the same rule on all test instances. Rather, it is sufficient to simply assume that, in a given situation, if S has an appropriate rule or procedure available, then he will use it. This allows S to use one available procedure on one test instance and another one on a different instance. For example, in adding numbers, S might apply the ordinary addition algorithm to a problem like 726 + 398, whereas a doubling technique of sorts might be used on a problem like 250 + 250.

With this in mind, let us turn more specifically to the problem at hand: how to determine a S's behavior potential relative to a given class of rule-governed (RG) behaviors, on the basis of S's performance on a small finite number of test
instances. (By a class of RG behaviors is meant a class of stimuli and a class of responses such that to each stimulus in the first class there is a unique response in the second class which may be associated with it by a rule. To make things more definite, one may think of a particular class of S–R pairs or instances†—where the stimuli, say, are pairs of numbers and where the responses are the corresponding sums. 

(Mathematically, a class of RG behaviors corresponds to what recursion theorists call a computable function.‡ This means that there is a rule, effective procedure or algorithm—the terms may be used interchangeably—such that, given any stimulus associated with the class, one might generate the corresponding response. Stated differently, there is an effective procedure which accounts for the class of behaviors. In fact there are denumerably many different procedures which might account for any given RG class.)

The definition of “What rule is learned” in Scandura (1970), takes what may be a small step in the right direction. The basic idea is that given a performance profile, characterized by success on m of n instances in a given RG class and failure on the remaining n–m instances, then what is learned may be defined as the class of rules which provides an adequate account of the data. As indicated, this definition provides a way for deciding whether or not any given rule is to be included in the class. It can also be useful in making predictions about behavior potential. What the definition does not do, however, is to provide a systematic way to go about this most difficult task. For one thing, the definition says nothing directly about the test instances needed to determine behavior potential. To specify the behavior potential of an individual relative to a given class of RG behaviors, one must be able to specify a subset of test instances which make it possible to predict S’s performance on all the remaining items.

The traditional approach to this problem has been to resort to probability statements on one sort or another. Thus, predictions about behavior are made on the basis of performance on a random sampling of test items. In this case, one might expect, on the basis of S’s performance on a sample of test items, to make such statements as, “On the average, he should get eight out of ten items correct.”

The point of view taken here is that, under memory-free conditions, there is no need to resort to probability in describing the behavior potential of human beings. This does not imply perfect prediction but only that uncertainty, rather than being an explicit part of the theory or prediction model, has its source elsewhere. The reason that perfect prediction is impossible, in principle, is that in addition to the role played by memory, there is always some residual

†Each S–R pair is called an instance of the class of RG behaviors.
uncertainty about the capabilities and motivations that an S brings to any
given task. And, this is exactly where this uncertainty should be put.

Rather than make probabilistic predictions, based on a random sampling
of test instances, an operational definition of behavior potential is proposed
which makes possible deterministic predictions relative to the adequacy of
certain assumptions concerning the motivations and prerequisite capabilities
S brings to the situation. The basic idea, involved in selecting an adequate set
of test instances, is to partition the given class of RG behaviors into a set of
mutually exclusive and exhaustive subsets so that each subset of instances is
atomic. By an atomic subset is meant an equivalence class in which success on
any one instance in the class will be indicative of success on any other instance
in the class, and similarly for failure. Studies conducted by Scandura and his
students and others (e.g., Restle and Brown, 1970) over the past several years
strongly suggest the existence of such equivalence classes. Scandura and
Durnin’s Ss, for example, almost always performed uniformly (well or poorly)
on classes of similar items.

Once such a partition has been found, of course, the next step is obvious.
One simply selects one test instance for each element (i.e., equivalence class) in
the partition. Predictions on new instances, then, are made in accordance with
whether or not the S succeeds on the corresponding test instance—that is, the
test instance in the same equivalence class of the partition.

Unfortunately, the problem of how to determine such a partition in the first
place is not trivial. Some idea of the difficulty may be seen by considering
simple addition. Any attempt to identify equivalence classes of addition
problems, immediately raises such questions as whether 25 + 30 is more like 5
+ 30 or more like 20 + 30. Clearly, some alternative to sheer guesswork is
needed.

A solution

After mulling this problem over for some time, it was tentatively concluded
that the most feasible way of identifying such a partition is to tackle the
problem intentionally—in terms of underlying procedures. (This does not rule
out the possibility that an extensional or dimensional analysis may also
work.)‡ The problem here is one of reducing the number of possible

‡Originally, it was proposed that behavior potential might be defined relative to certain (test)
stimulus dimensions which were assumed to be well-defined for the subject. In particular,
Scandura and Durnin (1968) found that if a subject is able to give the appropriate response to two
stimuli (associated with some rule-governed class) which differ along one or more dimensions,
then he is also almost invariably able to give the appropriate response to any other stimulus
which differs only along these (same) dimensions. They recognised the need to better define
the notion of a well-defined dimension but did not attempt any such definition.

Although developed on more pragmatic grounds, the item form method proposed by Hively
Theoretically, there are an infinite number of procedures which might account for any given class of rule-governed behaviors.

Fortunately, this does not seem to pose undue difficulty in practice. Given any familiar RG class, it is usually possible to determine the various procedures which the Ss in question are most likely to use. There are relatively few basically different procedures for adding fractions, for example.

It is also necessary to say something about the mechanisms which govern S's behavior. In this case, it is sufficient to simply assume:

A) Given a goal situation for which S has at least one available procedure, which applies in the situation, then the S will use one of them.

As trivial as this assumption may be, it is nonetheless an assumption. It does not follow logically that just because an S wants to achieve a certain goal and has one or more rules available for achieving it, that he will necessarily use one of them. Furthermore, it is an assumption which is accepted here without direct testing. The value of such an hypothesis, of course, depends on its usefulness in subsequent research. While it is beyond the scope of this report to consider this question, there is some reason to believe that this hypothesis, as simple as it appears, may ultimately come to play a central role in a more encompassing theory of behavior (cf. Scandura, 1971b).

Clearly, according to this simple assumption, if one knew what procedures S had at his command, one could predict precisely what kinds of behavior he would and would not be capable of. However, this is not generally known.

Nonetheless, given any finitary algorithm, it is always possible to break the procedure down into what might be called “atomic” subrules. By an atomic rule is meant a rule with respect to which success on one instance (of the associated RG class) implies success with others and similarity for failure. To see this, it is sufficient to note that Suppes (1969) has recently proved (essentially) that any finite connected automaton can be represented in terms of a finite set of S–R pairs. From this, it follows directly that any finitary rule can be represented in terms of either associations or (more general) constituent rules with finite domains (cf. Scandura, 1970). Thus, given any such rule, it is always possible to “break it down” far enough so that the constituent rules act in atomic fashion with any given S (or group of Ss).

(1963) also addresses itself to this problem. His data, however, clearly indicate that this approach is inadequate insofar as identifying homogeneous sets of items (i.e., equivalence classes) is concerned.

It now appears that both types of extensional analysis are more naturally interpreted in terms of underlying procedures.

†An algorithm equivalent in computing power to a finite connected automaton.
This is trivially true in the case where the constituent rules are associations. Furthermore, given even the barest minimum of information about an S's capabilities, it is usually possible in practice to make intelligent guesses concerning whether or not a given subrule is apt to act in atomic fashion. Thus, for example, in adding fractions many Ss either know to find least common multiples, or they don't. There is no inbetween (at least using that one procedure).

The precise nature of the relationship between a procedure and the atomic rules of which it is composed can perhaps be seen most easily by representing the former as directed graphs in which lines correspond to atomic rules and points or nodes to branches in the procedure. For example, consider the procedure for generating the "next" numeral in base three in which the atomic rules are listed as follows:

1) Read (encode) the last digit of the given numeral.
2) If the digit is a "2," change the "2" to "0," write it down, (and go to 4).
3) Increment the digit by 1, write the new numeral, (and stop).
4) If there is another digit to the left, encode it. (and go to rule 2).
5) Write "1" in the next position to the left of the last "0" written (and stop).

The S–R instances generated by the algorithm would be of the following sort: 0 → 1, 1 → 2, 2 → 10, 10 → 11, ..., 1022 → 1100, .... This algorithm may be represented by the directed graph,

![Directed graph diagram]

where the individual arcs of the graph correspond to atomic rules (which correspond to steps in the algorithm). Notice in this particular case that there are four distinct paths through the algorithm:

![Path diagrams a, b, c, d]

When looked at in this way it is easy to see that a person might master certain paths of a given procedure but not others. Furthermore, given that the rules in such a procedure (including the decision rules) are atomic, one can logically conclude that each path in the procedure must also act in atomic fashion. From this it follows that each procedure effectively partitions the original RG
class into equivalence classes of the type mentioned earlier. If \( E \) knows the atomic rules \( S \) has and the ways in which they might be combined to form procedures, then he can determine which paths \( S \) has learned and which paths \( S \) has not by observing \( S \)'s performance on a single instance from each equivalence class in the partition. The experimental hypothesis may, then, be stated as follows: Given that \( S \) has mastered a set of atomic rules, and assuming all learning relative to these atomic rules has taken place prior to testing, then if \( S \) can solve any task in a given equivalence class, he should also be able to solve any other task in the same equivalence class—and similarly for failure. Correct predictions on this basis, will indirectly support hypothesis (A).†

This intentional approach also provides a systematic way of devising a procedure which directly parallels \( S \)'s behavior potential—that is, for constructing a procedure which generates exactly those \( S \)--R instances on which success can be expected. This procedure, in turn, effectively defines behavior potential.

The definition goes as follows:

Given a class of RG behaviors and assuming a constant goal and no learning during testing (the consistency conditions), and given a set of underlying procedures, each of which is composed entirely of atomic rules, then the \( S \)'s behavior potential may be defined, relative to these assumptions, as that subclass of the given RG class which can be generated by procedures obtained from the original procedures by deleting those atomic rules which contribute only to the generation of incorrect responses during testing.

Implicit in this definition is the assumption that the paths of a given procedure can be partially ordered. A path which contains all the atomic rules of another path plus some of its own would be, relatively speaking, a higher

†The usefulness of this approach in future research, as well as practice, requires more than just testing the theoretical hypothesis upon which it is based. With experimental confirmation of this simple hypothesis in the assessment situation, research can proceed in several directions. First of all, it would be important to determine the value of this procedure in assessing the behavior potential of individuals at different developmental levels and with a variety of different kinds of meaningful tasks. This could be particularly important for research in various subject matter areas such as mathematics. Second, it might be used to determine individual differences in computational ability so that this might be taken into account in studies of problem solving in which \( S \)s have to compute. If the procedure can also be made to work with such tasks as number conservation and the like, developmental psychologists could have a valuable new tool for their research.

Insofar as practice is concerned, there are also some fairly immediate implications of such a procedure for diagnostic testing, sequential testing and computer assisted instruction. Research on some of these implications will be given in future reports.
order path. From this it follows (on largely logical grounds) that if S is successful on an instance associated with a higher order path, then S should also be successful on items associated with all (relatively) lower order paths. This was the second hypothesis to be tested experimentally.

METHOD AND RESULTS

A number of experiments designed to test the above hypotheses have been completed.

The basic approach used was as follows:

1) A suitable class of rule governed behaviors (i.e., S–R pairs) was selected, together with an algorithm for generating each behavior in this class. That is, given any stimulus in the class of stimuli, the corresponding response could be generated by following a path of this algorithm. It was of particular interest to choose tasks which were not so easy that the Ss were uniformly successful on the task, nor so hard that they were uniformly unsuccessful.

2) Each of the atomic rules included in this algorithm was then built directly into the S in question in the sense that he could correctly apply each of the constituent atomic rules uniformly well. Where the number of instances of an atomic rule was infinite, enough varied instances of the rule were chosen to assure that the S did have complete mastery.

3) Armed with these atomic rules as prerequisite knowledge, the S was presented with stimulus instances of the given rule-governed class and was required to generate, if he could, the corresponding responses. After the S gave his response or indicated that he did not know the answer, knowledge of results was given in either one of two forms: self-reinforcement or extrinsic reinforcement provided by the experimenter. This procedure was continued with new instances until the S seemed not to be making any further progress on the learning trials. During the learning period, care was taken to insure that the S had ample opportunity with instances corresponding to each path of the algorithm.

4) The S was then tested on one arbitrarily selected instance for each path. More accurately, the instances were selected from the equivalence classes corresponding to the various paths through the algorithm.

5) Predictions were made about the S's performance on future instances associated with these equivalence classes (paths of the algorithm) on the basis of how the S had performed on the test instances. These predictions were checked in the obvious way by simply testing the subject on another set of representative instances.

‡A detailed explanation of this point would require more attention to the nature of the operating and decision making rules used in constructing procedures than can be given here. The basic idea, however, seems reasonably clear.
According to previous analyses, confirmation was expected with all Ss on all tasks to the extent that the presumed boundary conditions were met—that is, that the S was actually trying to solve the problems, that the presumed atomic rules were indeed atomic, and that the S was unencumbered by memory or his limited capacity to process information. Hence the aim was to run this basic experiment with a wide variety of different tasks and with a wide variety of different Ss at different developmental levels. In particular, Ss were sampled from graduate school down to the pre-school level. Notice that each combination of task and S constitutes a replication of the basic experiment.

**EXPERIMENTS 1–5**

The task used in the first set of experiments was one in which each S could provide his own reinforcement. That is, the S had available an independent means for checking his answers during the learning.

**Task I**

1) The class of rule-governed behaviors for these experiments involved finding multiplicative inverses of four-tuples (e.g., \((0, 1, 2, 0)\)) of integers modulo 3. The system of integers mod 3 (which consists of elements 0, 1, and 2) may be defined by the addition and multiplication tables:

<table>
<thead>
<tr>
<th>+</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>·</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

The S was shown how to multiply two four-tuples by using the definition \((a, b, c, d) \times (e, f, g, h) = (ae + bg, af + bh, ce + dg, cf + dh)\) where + and · are the operations of addition and multiplication, respectively, in mod 3. For example, \((1, 0, 2, 1) \times (2, 0, 1, 2) = (1 \cdot 2 + 0 \cdot 1, 1 \cdot 0 + 0 \cdot 2, 2 \cdot 2 + 1 \cdot 1, 2 \cdot 0 + 1 \cdot 2) = (2, 0, 2, 2)\).

A four-tuple is said to have a multiplicative inverse if there exists a second four-tuple (possibly the same) such that their product is \((1, 0, 0, 1)\). For example, since \((1, 0, 2, 1) \times (1, 0, 1, 1) = (1, 0, 0, 1)\), \((1, 0, 1, 1)\) is said to be the multiplicative inverse of \((1, 0, 2, 1)\) and vice-versa. The S's goal in each of the following experiments was to find inverses (responses) of certain given (stimulus) four-tuples. (This task was modeled after finding inverses of \(2 \times 2\) matrices.)

If a four-tuple \((a, b, c, d)\) has an inverse, then the following flow diagram presents one algorithmic procedure for finding it.
START

Compute a\cdot d and b\cdot c

\text{Compute } a\cdot d + *(b\cdot c)

\begin{align*}
\text{Is } b\cdot c = 0? & \Rightarrow \text{no} \\
\text{Is } b = c? & \Rightarrow \text{yes}
\end{align*}

\begin{align*}
\text{Does } a\cdot d = 2 \text{ with } b\cdot c = 0 \text{ or } a\cdot d = *(b\cdot c) = 2? & \Rightarrow \text{yes}
\end{align*}

\begin{align*}
\text{Is } a = d? & \Rightarrow \text{no}
\end{align*}

Interchange a and d and then compute (2\cdot d, 2\cdot b, 2\cdot c, 2\cdot a)

Compute (2\cdot a, 2\cdot b, 2\cdot c, 2\cdot d)

\begin{align*}
\text{Is } a = d? & \Rightarrow \text{no}
\end{align*}

Interchange a and d

\begin{align*}
\text{Is } a = d? & \Rightarrow \text{yes}
\end{align*}

For b \neq 0 change b to *(b) and for c \neq 0 change c to *(c)

STOP

\text{FIGURE 1}
The diagram indicates that there are several other paths through the procedure (e.g., ). There are, however, no instances in the given rule-governed class which require the use of these (vacuous) paths and so they may be discarded. The numbers associated with the various arrows specify which atomic rules they represent.

This flow diagram can be represented more simply as a directed graph where lines represent atomic rules and points, the branching decisions which need to be made in carrying out the algorithm. This graph, together with the six possible paths through it and illustrative stimuli associated with these paths is given in Figure 2.

2) During the preliminary training phase, the subjects were first taught how to add and multiply integers mod 3 and an operation * defined by *(2) = 1 and
*(1) = 2. They were also given the definition for multiplying four-tuples and practice in applying the definition. The Ss were also shown the identity (1, 0, 0, 1) and told what an inverse was. This information made it possible for the Ss to check.

Then each S was taught the following six atomic rules:

1) Find the products \(a \cdot d\) and \(b \cdot c\) (i.e., multiply \(a, d\) and \(b, c\)).
   
   \textit{Example:} (1, 0, 2, 1) \quad a \cdot d = 1, b \cdot c = 0.

2) Find \(* (b \cdot c)\) and the sum \(a \cdot d + *(b \cdot c)\)
   
   \textit{Example:} \quad b \cdot c = 2, a \cdot d = 1 \quad * (b \cdot c) = 1.
   
   \quad a \cdot d + *(b \cdot c) = 2.

3) \(a\) interchange \(a\) and \(d\); i.e., \((a, b, c, d) \rightarrow (d, b, c, a)\)
   
   \(b\) multiply \(d, b, c,\) and \(a\) by 2; i.e. \((2d, 2b, 2c, 2a)\)
   
   \(c\) for \(2b \neq 0\) change \(2b\) to \(* (2b)\) and for \(2c \neq 0\) change \(2c\) to \(* (2c)\)
   
   \textit{Example:} \quad (2, 2, 1, 0) \rightarrow (0, 2, 1, 2) \rightarrow (0, 1, 2, 1) \rightarrow (0, 2, 1, 1).

4) \(a\) multiply \(a, b, c,\) and \(d\) by 2; i.e., \((2a, 2b, 2c, 2d)\)
   
   \(b\) for \(2b \neq 0\) change \(2b\) to \(* (2b)\) and for \(2c \neq 0\) change \(2c\) to \(* (2c)\).
   
   \textit{Example:} \quad (2, 1, 0, 1) \rightarrow (1, 2, 0, 2) \rightarrow (1, 1, 0, 2).

5) \(a\) interchange \(a\) and \(d\); i.e., \((a, b, c, d) \rightarrow (d, b, c, a)\).
   
   \(b\) for \(b \neq 0\) change \(b\) to \(* (b)\) and for \(c \neq 0\) change \(c\) to \(* (c)\).
   
   \textit{Example:} \quad (2, 1, 0, 1) \rightarrow (1, 1, 0, 2) \rightarrow (1, 2, 0, 2).

6) For \(b \neq 0\) change \(b\) to \(* (b)\) and for \(c \neq 0\) change \(c\) to \(* (c)\).
   
   \textit{Example:} \quad (1, 1, 0, 1) \rightarrow (1, 2, 0, 1)

The S was trained on each rule to a criterion of at least three correct responses in a row and was allowed as much time in working with the rules as he required. The order of presenting these rules was randomized.

3) After S had learned the six atomic rules to criterion, he was given a practice sheet consisting of 24 stimulus instances from the rule-governed class. The problems presented were divided into four sets of six instances each, one instance from each of the six equivalence classes. Within each set, the instances were randomized. The S was then told to find the inverse of each four-tuple, using the (atomic) rules he had just learned. He was told to do as many of the problems as he could and to check his answers by multiplying the four-tuple he derived with the one given. (In order to be correct, the product had to be (1, 0, 0, 1).) A printed statement of the rule for multiplying four-tuples was available to him at all times so that he did not need to commit the rule to memory. The S was allowed as much time as he needed to complete the problems.
4) After S had completed as many problems as he could, E collected the problems of the same type. There were no time limits on either test, so the S would no longer be able to check his answers. This was done to prevent the S from learning while being tested, and corresponds to Levine's (1966) non-reinforcement trials.

The first set of test problems consisted of six new stimulus instances, one instance from each equivalence class. During the testing, as well as the pretraining, the S had statements of the atomic rules available in case he forgot any of them.

5) Immediately following this test, the S was given a second set of test problems of the same type. There were no time limits on either test.

The results of five experiments (five Ss using this task) are given in Table 1.

| TABLE 1 |
| --- | --- | --- | --- | --- | --- |
|   | 1st test | 2nd test | 1st test | 2nd test | 1st test | 2nd test |
| 1 | + | + | + | + | + | + |
| 2 | + | + | + | + | − | − |
| 3 | + | + | + | + | + | + |
| 4 | + | + | + | + | − | − |
| 5 | + | + | + | + | + | − |
| 6 | + | + | + | + | − | − |

"+" indicates a correct response while "−" indicates an incorrect response. Encircled pairs indicate results which went contrary to prediction. In this case, there were 29 correct predictions and one incorrect.

**EXPERIMENTS 6–21**

The second set of experiments involved tasks in which the reinforcement was external. Only on the practice trials did E tell the S whether or not he was correct.
Task II

1) The artificial class of rule-governed behaviors used in experiments 6–9 consisted of strings of symbols from which the S was required to obtain a certain number by using the atomic rules given him. The following is a sample list of some of the instances in the rule-governed class. (This task was modeled after Polish notation.)

**Stimulus**

| M A 5,6, A 2,1 | 33 |
| IM 5,1 A 3,2 | 1 |
| IM A 4,2,13 | \(\frac{1}{2}\) |
| 8 | 8 |

**Response**

START

Read each number

Is any no. immediately preceded by an I?

- yes → Change Ix to 1/x

- no

Is there a pair of numbers preceded by an M or A?

- yes → Read the pair of numbers

  - Is the pair preceded by an A or H?

    - yes → Change Mx,y to the product of x and y

    - no → A

  - no

- no → STOP

FIGURE 3
An algorithmic procedure for generating this class of behaviors is described by the flow diagram of Figure 3.

A directed graph of this procedure is given below with its paths and representative stimuli from the corresponding equivalence classes.

2) The procedure and criterion used in training the S on the atomic rules were essentially the same as those used in the previous set of experiments. The S was given the following set of atomic rules:

1) Read each number.
2) Change Ix to 1/x.

Example: Stimulus Response
I 3 1/3

3) Read a pair of numbers.

START \[
\frac{i^2}{4(\frac{3}{5})}
\]
\[
\implies
\]
STOP

Paths

1. 1 \[\rightarrow \]
2. 1 \[\rightarrow \]
3. 1 \[\rightarrow \]
4. 1 \[\rightarrow \]
5. 1 \[\rightarrow \]
6. 1 \[\rightarrow \]
7. 1 \[\rightarrow \]
8. 1 \[\rightarrow \]

Stimulus instances from corresponding Equivalence Classes

3
I 4
I M 4, I 3
M M 4, 3, M 2, 1
A A 4, 3, A 1, 5
I A 4, I 2
M A 3, 4, A 2, 1
M A 2, 1, I A 3, 3

FIGURE 4 Directed graph.
4) Change $M \times y$ to the product of $x$ and $y$.

*Example:*  
**Stimulus**  
**Response**

$M 4, 1/2$  
$2$

5) Change $A, x, y$ to the sum of $x$ and $y$.

*Example:*  
**Stimulus**  
**Response**

$A 4, 5$  
$9$

3) After the $S$ had learned the five atomic rules to criterion, he was presented with a practice sheet of 32 stimulus instances, four instances from each of the eight equivalence classes. The arrangement of these instances was the same as in the above experiments.

The $S$s were told to try to solve each problem using the atomic rules they had just learned. After each answer was given, the $E$ informed the $S$ whether his response was correct or incorrect and then told him to go on to the next problem. This procedure was continued until the $S$ used up the four available instances or performed uniformly well or poorly on *at least* two instances in a row from each of the eight equivalence classes.

4–5) As before, each $S$ was tested in turn on two sets of eight instances, one instance in each set from each equivalence class. No reinforcement was given during the testing.

The results of these tests for each experiment are given in Table 2.

**Task III**

1) This stimuli in experiments 10 and 11 consisted of arrays of squares with column numbers along the top and row numbers along the right hand side of each array (e.g.,

```
  4  5  6
  2  3
```

The task was to fill in the array and find appropriate numbers corresponding to each square along the bottom and left hand side of the array (e.g.,

```
  1  2  1  5  1  8
  0  0  1  1  2
```

(This task is an algorithm for computing products of whole numbers.)
TABLE 2
Experiments

<table>
<thead>
<tr>
<th>Equivalence Classes</th>
<th>6 College Student</th>
<th>7 College Student</th>
<th>8 H.S. Student</th>
<th>9 H.S. Student</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st test instance</td>
<td>2nd test instance</td>
<td>1st test instance</td>
<td>2nd test instance</td>
</tr>
<tr>
<td>1</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>5</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>6</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>7</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>8</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

"+" indicates a correct response.
"-" indicates an incorrect response.

A procedure governing the desired class of behavior is represented in Figures 5 and 6.

2) The atomic rules were:

1) Start with any empty square in the array. Multiply the column number of the square with its row number and put the units digit of the product below the diagonal in the square.

*Example:*

```
     6
  2  7
```

2) Put 0 above the diagonal in the square of the product.

*Example:*

```
  4
0  2
```
3) Put the tens digit of the product above the diagonal in the square.

Example:

![Example Diagram]

4) Write the number in the "bottom" diagonal at the end of the diagonal outside the array.
Example:

5) Add the numbers in the diagonal and write the units digit of the sum at the end of the diagonal outside the array.

Example:

6) Add the tens digit from the sum of the lower diagonal to the start of the next diagonal. Add the numbers in that diagonal and write the units of the sum at the end of the diagonal outside of the array.

Example:

3) After training the S was presented with 32 practice instances from the rule-governed class.

4–5) Table 3 gives the results of the tests following practice.

Task IV

1) The class of rule governed behaviors for experiments 12 through 15 was designed for use with preschoolers. The task was to cross out squares, circles and combinations thereof from a display of three circles and three squares (e.g., ○ ○ □ ○ □ □). The S was given a stack of 3" × 5" stimulus cards with a colored figure drawn on one side of each card. The figures varied along two
dimensions: color (red, blue and yellow) and shape (square and circle), with the relevant dimension being shape. The sides with figures on them were turned face down. The S was to turn each card over and cross out on the display sheet whatever figure was shown on the card.

A flowchart and directed graph of a procedure governing this behavior are shown in Figures 7 and 8.

**TABLE 3**

*Experiments*

<table>
<thead>
<tr>
<th>Equivalence Classes</th>
<th>10 6th grader</th>
<th>11 6th grader</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st test</td>
<td>2nd test</td>
</tr>
<tr>
<td></td>
<td>instance</td>
<td>instance</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

"-" indicates a correct response.
"+" indicates an incorrect response.

2) The S was trained on atomic rules 1, 2, and 3 shown in the flow diagram.

3) After this initial training the S was given 20 practice instances from the rule governed class.

4–5) Each S was then tested on two sets of five instances. The results of the tests are shown in Table 4.

In conducting experiments 12 and 13 we observed that the two older Ss who could count used a slightly different procedure than that shown in the preceding flow diagram. A flow diagram which approximates the procedure they used is given in Figure 9.
The paths through this procedure partition the class of rule governed behaviors into the same equivalence classes as the paths of the previous procedure. Hence, the same items could be used for assessing behavior potential regardless of which procedure the subject used.

FIGURE 7
FIGURE 8 Directed graph.

TABLE 4
Experiments

<table>
<thead>
<tr>
<th>Equivalence Classes</th>
<th>12 5 year old</th>
<th>13 4 year old</th>
<th>14 3 year old</th>
<th>15 2½ year old</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st test instance</td>
<td>2nd test instance</td>
<td>1st test instance</td>
<td>2nd test instance</td>
</tr>
<tr>
<td>1</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>5</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

"+" indicates a correct response.
"-" indicates an incorrect response.
START with the top card

☐ ×
x is a ○ or a □

Is there another card to turn over?

yes

no

Are there □ 's?

yes → Count the no. of □ 's → Apply ○ ⇒ × that many times

no → Are there ○ 's?

yes → Count the no. of ○ 's → Apply ○ ⇒ × that many times

no → STOP

FIGURE 9
Task V

1) A sample list of the desired behaviors from the rule-governed class used in the experiments 16 through 19 is given below.

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐☐☐☐☐</td>
<td>☐☐☐☐☐☐☐☐☐☐</td>
</tr>
<tr>
<td>☐☐☐☐☐☐☐☐☐☐</td>
<td>☐☐☐☐☐☐☐☐☐☐</td>
</tr>
<tr>
<td>☐☐☐☐☐☐☐☐☐☐</td>
<td>☐☐☐☐☐☐☐☐☐☐</td>
</tr>
</tbody>
</table>

The flow diagram and directed graph in Figures 10 and 11, respectively, represent a procedure for generating this behavior.

2) Each S was trained on rules 1 and 2 in the procedure.

3) The practice conditions in these experiments were the same as those in experiments 6–15.

4–5) The results of the tests are shown in Table 5.
FIGURE 11 Directed graph.

TABLE 5
Experiments

<table>
<thead>
<tr>
<th>Equivalence Classes</th>
<th>16 7 year old</th>
<th>17 6 year old</th>
<th>18 4 year old</th>
<th>19 4 year old</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st test</td>
<td>2nd test</td>
<td>1st test</td>
<td>2nd test</td>
</tr>
<tr>
<td>1</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

+ + indicates correct response.
- - indicates incorrect response.
Notice that in experiment 17, the S missed alternate items from equivalence classes 2 and 3. The first and second items in each test differed in the number of times the S was required to apply the particular atomic rule involved. The instances on which the S was successful required only one application of the atomic rule, whereas the missed instances required repeated application of the same rule.

As was noted earlier, the atomic rules in this case would have to be refined so as to distinguish between only one application of the rules and more than one application.

Task VI

1) In experiments 20 and 21, each S was shown six 3" × 5" response cards. Three of the cards had one of the numerals “1,” “2,” or “3” along with one, two, or three dots, respectively, written on them. Each of the remaining three cards had a circle, triangle or square drawn on it. The S was then shown two stimulus cards, placed above the six response cards, containing figures varying in shape (circle, triangle, and square) and number (one, two, and three). An example of a stimulus display is shown below.

\[
\begin{align*}
\text{Stimulus cards} \\
\begin{array}{c}
\bullet\bullet \\
\Delta\Delta
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{cccc}
1 & 2 & 3 & \text{Response cards} \\
\bullet & \bullet & \bullet & \circ \\
& & \Delta & \square
\end{array}
\end{align*}
\]

The task was to turn over the response card(s) which showed which values of the number and shape dimensions were the same. For example, the correct response to the above display

\[
\begin{align*}
\begin{array}{c}
\bullet\bullet \\
\Delta\Delta
\end{array}
\end{align*}
\]

is:

\[
\begin{align*}
\begin{array}{cccc}
1 & 3 & \circ & \Delta & \square
\end{array}
\end{align*}
\]

One procedure for generating this behavior, together with the various paths associated with it, is represented in Figures 12 and 13.
2) Since the Ss in these experiments were unable to read, the atomic rules were presented verbally.

1) "Turn over the card which has the same shape on it as both of the above two cards."

Example:  

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Stimulus Card 1]</td>
<td>![Response Card 1]</td>
</tr>
<tr>
<td>![Stimulus Card 2]</td>
<td>![Response Card 2]</td>
</tr>
<tr>
<td>![Stimulus Card 3]</td>
<td>![Response Card 3]</td>
</tr>
<tr>
<td>![Stimulus Card 4]</td>
<td>![Response Card 4]</td>
</tr>
<tr>
<td>![Stimulus Card 5]</td>
<td>![Response Card 5]</td>
</tr>
<tr>
<td>![Stimulus Card 6]</td>
<td>![Response Card 6]</td>
</tr>
<tr>
<td>![Stimulus Card 7]</td>
<td>![Response Card 7]</td>
</tr>
<tr>
<td>![Stimulus Card 8]</td>
<td>![Response Card 8]</td>
</tr>
<tr>
<td>![Stimulus Card 9]</td>
<td>![Response Card 9]</td>
</tr>
<tr>
<td>![Stimulus Card 10]</td>
<td>![Response Card 10]</td>
</tr>
</tbody>
</table>

FIGURE 12
FIGURE 13 Directed graph.

2) “Turn over the card which has the same number of dots as figures in both of the above two cards.”

Example:  

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Card 1" /></td>
<td><img src="image2" alt="Card 2" /></td>
</tr>
</tbody>
</table>

After the S had learned each rule to criterion, he was required to state in his own words as accurately as possible the rule he had learned. This was done to help insure that the S had a correct interpretation of each rule.
3) On each trial in the practice session the S was presented with two stimulus cards. He was told to turn over those response cards which showed how the two stimulus cards were the same. During the practice session the rules were repeated each time two new stimulus cards were shown. Whenever the S asked what he was to do, the E restated both rules.

4-5) The results of the tests are shown in Table 6.

<table>
<thead>
<tr>
<th></th>
<th>20 5 year old</th>
<th>21 6 year old</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equivalence Classes</td>
<td>1st test instance</td>
<td>2nd test instance</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

"+" indicates correct response.
"-" indicates incorrect response.

EXPERIMENTS 22–30

In the third set of experiments the Ss had no way of knowing during practice whether their answers were correct or not. They were neither told when they were correct nor given an independent means for checking their answers. The main point here was to determine whether the intrinsic structure of procedures may itself provide sufficient intrinsic reinforcement for learning.

Task VII

1) The class of rule governed behaviors used in experiments 22–26 consisted of combining strings of pairs of symbols by using certain rules. The following is a sample list of some of the instances in the rule-governed class.

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 2) (2, 4)</td>
<td>(1, 4, 2)</td>
</tr>
<tr>
<td>(Δ, □) (□, ○)</td>
<td>(Δ) (□, ○)</td>
</tr>
<tr>
<td>(a, 2) (□, 2)</td>
<td>(a) (2, Δ, □)</td>
</tr>
<tr>
<td>(b, b)</td>
<td>(b)</td>
</tr>
</tbody>
</table>

(This task was modeled after composition of two-cycles in permutations.)
An algorithmic procedure for generating this class of behaviors is represented in Figures 14 and 15.

2) The atomic rule statements used in experiments with this task differed from those in the previously described tasks in that branching instructions were included in the rule statements. For each of the following atomic rules, the S was informed that “y” was a variable.

1) Select the left most element, y, in the left most pair and write “(y,).” Shift to the other element in that pair (making it y).

Example:

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, a) (3, b) (4, 3) (c, l)</td>
<td>Written: (1,</td>
</tr>
<tr>
<td>y</td>
<td>Verbal: “a is y”</td>
</tr>
</tbody>
</table>

2) If y appears again in another pair to the right, go to the closest pair to the right of which y is an element and shift to the other element in that pair (making it y).

Example:

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>(∆, o) (□, *) (□, o) (o, △)</td>
<td>&quot;□ is y&quot;</td>
</tr>
<tr>
<td>y</td>
<td></td>
</tr>
</tbody>
</table>

3) If y does not appear in a pair to the right and y has not been written in the answer before, write y in the answer and start at the left most pair containing y and shift to the other element in the pair (making it y).

Example:

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>(∆, o) (#, 1) (1, o) (o, #)</td>
<td>(∆, 1,</td>
</tr>
<tr>
<td>y</td>
<td>Written: (∆, 1, o</td>
</tr>
<tr>
<td></td>
<td>Verbal: &quot;∆ is y&quot;</td>
</tr>
</tbody>
</table>

4) If y has been written in the answer before, put a close parentheses after the answer.

Example:

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a, b) (c, a) (b, c)</td>
<td>(a, c</td>
</tr>
<tr>
<td>y</td>
<td>Written: (a, c)</td>
</tr>
</tbody>
</table>
5) If the number of distinct elements in the answer does not equal the number of distinct elements in the problem, select a \( y \) which is not in the answer but in the problem; write \("(y,\)"; start at the left most pair containing \( y \) and shift to the other element in the pair (making it \( y \)).

Example:

\[ \begin{align*}
  \text{Stimulus} & \quad \text{Response} \\
  (a, b) (b, a) (c, d) (e, d) & \quad (a, b) \quad \text{Written: (a, b)} (c, \\
  y & \quad \text{Verbal: \("d is y\")} 
\end{align*} \]
6) If the number of distinct elements in the answer equals the number of distinct elements in the problem, then stop.

Example:

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 3) (4, 5) (6, 5) (1, 2)</td>
<td>Verbal: “Stop”</td>
</tr>
<tr>
<td>(1, 2, 3) (4, 5, 6)</td>
<td></td>
</tr>
</tbody>
</table>

3) After training, the S was presented with a list of 32 strings of pairs and told to combine each string using the rules. *The S received no reinforcement during this practice session.*
## TABLE 7
Experiments

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st test instance</td>
<td>2nd test instance</td>
<td>1st test instance</td>
<td>2nd test instance</td>
<td>1st test instance</td>
</tr>
<tr>
<td>1</td>
<td>+ + + + + + + + + +</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+ + + + + + + + + +</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>+ + + + + + + + + +</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>+ + + + + + + + + +</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>+ + + + + + + + + +</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>- - + + + + + + + +</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>- - - - - - + + + +</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>- - - - - - + + + +</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*"+" indicates correct response.  
*"-" indicates incorrect response.

4–5) The results of the tests are given in Table 7.

### Task VIII

1) The following is a partial list of instances from the class of rule governed behaviors used in experiment 27.

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Stimulus" /></td>
<td><img src="image" alt="Response" /></td>
</tr>
</tbody>
</table>

A procedure governing this behavior is represented in Figures 16 and 17.
2) The atomic rules the S was trained on were:

1) If four squares contain an identical figure, cross them out and draw a new square containing the given figure. Then draw a triangle containing the new square.
   e.g., \[\begin{array}{c}
   \times \\
   \times \\
   \bigcirc & \bigcirc
   \end{array}\] and write \[\bigtriangleup\]

2) If three triangles contain an identical figure, cross them out and draw a new triangle containing the given figure. Then draw a circle containing the new triangle.
   e.g., \[\begin{array}{c}
   \bigtriangleup \\
   \bigtriangleup \\
   \bigtriangleup
   \end{array}\] and write \[\bigcirc\]

3) If two circles contain an identical figure, cross them out and draw a new circle containing the figure. Then draw a square containing the new circle.
   e.g., \[\begin{array}{c}
   \bigcirc & \bigcirc
   \end{array}\] and write \[\bigbox\]
3) The practice problems consisted of 32 strings. The S received no reinforcement during this practice session.

4–5) The results of this experiment are given in Table 8.

**Task IX**

1) The class of rule governed behaviors used in experiments 28 to 30 involved reducing strings of A's and B's, where \( n = 0, 1, 2, \ldots \). A sample list of instances is given below.

**Stimulus**

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>A A B^0 B^1 B^1 B^1 B^0 B^0 B^1</td>
<td>A A B^0 B^2 B^2</td>
</tr>
<tr>
<td>A A A A</td>
<td>B^0</td>
</tr>
<tr>
<td>B^0 A</td>
<td>B^0 A</td>
</tr>
</tbody>
</table>
A procedure for generating this class of behaviors is represented in Figures 18 and 19.

2) The atomic rules each S was trained on were:

1) Count the number of A’s.

   Example:

   **Stimulus**
   A A B\(^0\) A A B\(^1\) B\(^2\) A B\(^3\)
   **Response**
   “Five A’s”

2) Cross out 4 A’s and write B\(^0\) at the end of the string. Count the number of remaining A’s.

   Example:

   **Stimulus**
   A B\(^0\) B\(^1\) B\(^2\) A B\(^0\) A B\(^1\) B\(^3\) A
   **Response**
   A B\(^0\) B\(^1\) B\(^2\) A B\(^0\) A B\(^1\) B\(^3\) A B\(^0\)
   “zero A’s”

3) Let \( n = 0 \). Count the number of B\(^n\)’s for \( n = 0 \).

   Example:

   **Stimulus**
   A B\(^0\) B\(^1\) B\(^0\) A B\(^3\) B\(^2\) B\(^4\) B\(^7\) B\(^7\)
   **Response**
   “Two B\(^0\)’s”
4) Cross out 3 B\textsuperscript{n}'s and write B\textsuperscript{n+1} at the end of the string. Count the number of remaining B\textsuperscript{n}'s.

**Example:**

**Stimulus**

for \( n = 3 \)

\[ A \ B^0 \ B^3 \ A \ B^3 \ B^0 \ B^0 \ B^3 \ B^4 \ B^3 \]

**Response**

\[ A \ B^0 \ B^3 \ A \ B^3 \ B^0 \ B^0 \ B^3 \ B^4 \ B^3 \ B^4 \]

"One B\textsuperscript{3}"
5) Increase n by 1 and count the number of B^n's for the new n.

Example:

Stimulus

for n = 2

B^0 B^1 B^3 B^2 B^3 B^4 B^3
A B^0 B^3

Response

"4 B^3's"

3, 4–5) The practice and testing conditions for the Ss were essentially the same as those for Experiment 27. The results are shown in Table 9.
DISCUSSION

The results provide strong support for the basic hypothesis. In a total of 194 cases, there were 187 or 96% correct predictions. The 95% confidence interval for this proportion (96%) is between 93% and 99%.

The obtained results also provide evidence in favor of the second hypothesis. Not only does success on one instance of a given equivalence class imply success on any other instance, but success with one equivalence class frequently implies success on certain others. More generally, the equivalence classes associated with a given partition may be partially ordered as to difficulty.

Recall that the basis for this partial ordering resides in the nature of the corresponding paths. Thus, certain paths sometimes include others in the sense that the former include all of the atomic rules of the latter and in the same order, plus some additional steps. Consider, for example, Task IV. In this case, paths 5, 4, and 3 are superordinate to paths (4, 3, 2, 1), (1), and (2) respectively.
Notice too that in all 12 cases where a subject was consistently successful on a superordinate path, he was also successful on the subordinate paths. The converse is not always true. Of the 8 cases where the subjects were successful on all of the subordinate tasks, they were also successful on the corresponding superordinate tasks 6 times.

There were 205 cases in which one path was superordinate to another and in all but 7 or 3\% of these cases success on a superordinate task implied success on the subordinate ones. The 95\% confidence interval for the obtained 97\% level of prediction is between 95\% and 99\%. It may also be noted that 4 of the 7 exceptions can be attributed to the (strong) possibility that two of the Ss combined the given atomic rules of Task VI so as to form procedures which differed from those on which the E based his analysis. This possibility had no effect on the within equivalence class analysis, however, since the procedures actually devised by the Ss resulted in the same partition as did the experimenter-used-procedure.

Conversely, Ss were consistently successful on superordinate tasks in 90 of 117 cases in which they were also successful on all subordinate tasks (at the next lower level). These results are directly comparable to those obtained in using the now familiar task analysis technique pioneered in education by Gagné (1962) and his collaborators (e.g., Gagné, Mayor and Garstens, 1962). There too, successful performance on superordinate tasks is almost always indicative of success on subordinate tasks and success on all the subordinates frequently (about 80\% of the time) implies success on corresponding superordinate tasks.

This should not come as a surprise since the present form of analysis in terms of procedures parallels task analysis directly. The major difference is that where the intentional analysis allows for denumerably many different ways of solving a given class of tasks: task analysis implicitly assumes that there is just one. Analysis in terms of procedures is also more precise and explicit in the sense that the analyst is forced to make his intuitions public. It is interesting to note in this regard that some task analysts have recently begun to move in this direction (Resnick, 1970)

As suggestive as they might be, the experimental results deal only indirectly with the more practical problem of assessing the behavior potential of given Ss with respect to given classes of rule-governed behavior. In these experiments, the atomic rules were built directly into the Ss. Practice, however, requires that the observer make intelligent guesses as to which procedures a given S (or class of Ss) might reasonably use, including judgments concerning the inclusiveness of the component rules (i.e., what are the atomic rules).

The introduction of procedures in this way makes it possible, theoretically at least, to conceive of highly efficient testing techniques. Ideally, such techniques would capitalize not only on the identification of a finite number of
equivalence classes, but on the partial ordering imposed on these equivalence classes by the procedures introduced (by the experimenter). It would be possible, for example, to assess behavior potential through various forms of conditional or sequential testing. Such techniques could play a crucial role in computer assisted testing (and teaching) and in other forms of automated or semi-automated testing and instructional devices.

Another major practical advantage of this approach over traditional forms of testing is that it is self-correcting. If a particular analysis is in error, one knows exactly where to look for difficulties. When a probabilistic model is used, the test constructor simply knows how adequate or inadequate the predictions are. There are no explicit guidelines as to where to look in order to improve the level of prediction.

In the procedural approach, poor prediction may result, for example, where what are presumed to be atomic rules do not turn out to be atomic. In this case, more detailed analysis is called for. The class of behaviors associated with each such atomic rule can be analyzed in exactly the same way as with any other class of rule governed behaviors. The process involved would be analogous to analyzing a subroutine of a computer program. For example, consider the rule of step 1 in the illustrative algorithm given in the Introduction for generating next numerals in base three: "Read (encode) the last digit of a given (base-three) numeral." Anyone who had mastered this rule would certainly be able to "read" the last (circled) digits in "102," "13," "221" "20121" or in any other base-three numeral. While it would be difficult to find an adult for whom this ability did not act in atomic fashion, the situation with a young child just learning to read might be quite different. It might be necessary, for example, to replace step 1 by another (equivalent) procedure composed of less molar (atomic) rules. An obvious possibility might be to distinguish one-digit numerals from the rest. In this case, the new procedure might be represented graphically by

![Diagram](circle)

where one arc corresponds to reading one-digit numerals and the other to multi-digit numerals.

After breaking down each presumed atomic rule into more suitable sub-atomic rules, the test constructor could then construct and try out a new test in the same manner as before. This procedure, of course, could be repeated any number of times, but practically speaking, it is unlikely that an experienced test constructor would be off by very many levels of analysis.
Furthermore, it is always possible in principle to determine *a priori* how adequate the atomic assumptions are. Thus, one can always test directly to determine whether those rules, which are assumed to be atomic, are indeed atomic. In general, one simply looks to see whether success on one test instance implies success on any other test instance, and similarly for failure. If so, the rule is said to be atomic; otherwise, not. With respect to the ability to decode, for example, one might give the S a particular stimulus to see if his ability to decode the appropriate property in that stimulus necessarily implies his ability to determine the corresponding property of any such stimulus in the class.

As described, of course, the proposed approach applies only to given rule-governed classes and individual Ss. If one is concerned with a more diverse population of problems or with predicting the behavior of groups, as in ordinary testing, a few modifications need to be made in the basic approach.

In so far as the former problem is concerned, one cannot normally expect to find any single procedure which can generate all of the responses to a diverse population of problems. It may be possible, however, to identify a number of different generative procedures which come fairly close to matching the variety of problems in the population. In this case, test instances might be selected for each procedure in pretty much the same way as before. Of course, the number of procedures involved in such a situation could become extremely large. Nonetheless, this does not imply that the basic approach is in error. By following a modified form of this approach one could certainly do as good a job in constructing, say, standardized tests as is presently accomplished by random sampling from a hypothetical and ill-defined population of problems.

Conceptually speaking, extension of the proposed approach from individuals to groups is straightforward. It is sufficient to simply point out that, in general, it will be necessary to break down the underlying procedures more finely than with individuals since the atomic assumptions must be commensurate with the prerequisites had by the weaker members of the group. The important point is that when in doubt one can always analyze any given part into smaller constituents. Particularly in testing groups it would be better to go too far in analyzing a procedure than not far enough.

The only disadvantage of this approach with the more gifted members is that it makes assessing behavior potential less efficient than it might be. To the extent that the paths can be ordered as to "difficulty," of course, this limitation might tend to be counterbalanced. (Introducing additional test items would in any case increase reliability.)
References