AN ALGORITHMIC APPROACH TO ASSESSING BEHAVIOR POTENTIAL: COMPARISON WITH ITEM FORMS AND HIERARCHICAL TECHNOLOGIES

JOHN DURNIN
Villanova University

JOSEPH M. SCANDURA
University of Pennsylvania
AN ALGORITHMIC APPROACH TO ASSESSING
BEHAVIOR POTENTIAL:
COMPARISON WITH ITEM FORMS AND HIERARCHICAL TECHNOLOGIES

JOHN DURNIN
Villanova University

JOSEPH M. SCANDURA
University of Pennsylvania

An algorithm-based technology for assessing behavior potential was compared with two item form technologies. Bases for comparison were (a) relative effectiveness in predicting performance on individual test items, based on performance on items identified according to respective technologies; (b) relative power (generalizability); (c) relative efficiency (number of items); and (d) relative validity of item hierarchies. Two parallel tests on column subtraction were administered to 25 subjects. Test performance was analyzed according to each technology. Algorithmic technology (a) better predicted individual subjects failure on individual second test items, (b) had higher generalizability levels, (c) was more efficient, and (d) had higher validity indices on hierarchical ordering of tasks than item form technologies. Implications for diagnostic testing and remediation were discussed.

Recent research in individualized and computer-assisted instruction has led to an increasing awareness of the inadequacies of norm-referenced testing and the need for testing procedures that determine each individual's mastery on specific types of tasks (e.g., Coulson & Cogswell, 1965). Knowing how well a student has performed relative to some peer group, for example, says relatively little about the kinds of decisions that must be made if instruction is to be totally individualized. Ideally, mastery testing should (a) provide a sound basis for diagnosing individual strengths and weaknesses on each type of task, (b) require as few items as possible, and (c) provide a basis for generalizing from overall test performance to behavior on a clearly defined universe or domain of tasks. If, in addition, items can be ordered hierarchically to allow for conditional (sequential) testing, efficiency could be further increased.

Fortunately, a number of new technologies have recently been developed for constructing tests that have the above characteristics (e.g., Ferguson, 1969; Hively, Patterson, & Page, 1968; Scandura, 1971). The purpose of this study was to compare, with respect to these characteristics, three of the technologies; the item forms technology (domain-referenced testing) of Hively et al. (1968), the hierarchical or stratified item forms technology of Ferguson (1969), and the algorithmic technology of Scandura (1971, 1973).

In domain-referenced testing, a defined universe or domain of items (e.g., column subtraction problems) is subdivided into classes of items or item forms on the basis of observable properties the items in each class have in common. Osburn (1968) characterized an item form as having a fixed syntactical structure (e.g., $x - y$), one or more elements (e.g., $42 - 21, 28 - 16$), and explicit criteria for specifying which elements belong to the form (e.g., $x = x_1 x_2$; $y = y_1 y_2$; $y_1 < x_1$; $y_2 < x_2$; $x_1, x_2, y_1, y_2$ ∈

---

1 This study, based on a doctoral dissertation submitted by the first author under the second author's chairmanship, was supported by U.S. Office of Education Grant 3-71-0136 and, in part, by the National Science Foundation Grant GW 6796, both to the second author. The authors thank Alfonzo Georenoc and David Shore for their cooperation in providing subjects.

2 Requests for reprints and for a complete report should be sent to J. M. Scandura, Graduate School of Education, University of Pennsylvania, Philadelphia, Pennsylvania 19174.
To assess pupil performance on a given domain of problems, a test is constructed by randomly selecting one item from each of the identified forms. It was felt by Hively et al. (1968) that item forms might be used not only to assess a pupil's overall performance on the domain of problems but also to predict his behavior on specific problems in the domain. That is, if a subject were successful on one problem belonging to an item form, then he would be successful on any other problem of the same form, and similarly if he were unsuccessful on a problem belonging to an item form, he would be unsuccessful on any other problem of the same form. Although Hively et al. obtained high coefficients of generalizability (Cronbach, Rajaratnam, & Gleser, 1963) for tests based on the item forms technology, their item forms, in general, were not homogeneous in the sense that the problems associated with individual item forms were not behaviorally equivalent (regarding success-failure).

One criticism of the item forms technology has been that the hierarchical relationships among item forms have not been taken into account in testing. In a recent study by Ferguson (1969), these relationships were dealt with explicitly. In this study, item forms were generated for both terminal and prerequisite instructional objectives in a way analogous to task analysis (e.g., Gagné, 1962). Starting with a terminal item form, corresponding to a terminal instructional objective, subitem forms (i.e., subobjectives) that were considered prerequisite to the terminal item form were identified. The item forms so identified were then ordered according to the hypothesized hierarchical structure, and a computer was programmed to make branching decisions based on probabilistic evaluations of student performance on each of the forms. Clearly, a conditional testing procedure of this sort could conceivably provide a highly efficient basis for assessing the behavior potential of individual subjects.

Although the technologies for assessing mastery developed by Hively et al. (1968) and Ferguson (1969) appear to be major steps toward improved mastery and diagnostic testing, they are subject to serious criticism if one adopts a cognitive, information processing view. The technologies are based primarily on extensional analyses of item domains (i.e., analyses in terms of observable properties of items). With the possible exception of Ferguson’s hierarchical ordering of forms, which is based essentially on task analysis, there appears to be little basis other than (possible) sound intuitive judgment as to how items should be categorized. As a result, both technologies can be criticized on a priori grounds. For example, the item forms identified for subtraction by Hively et al. and those identified by Ferguson both failed to partition the domain of subtraction problems into mutually exclusive and exhaustive classes (i.e., equivalence classes). More specifically, some problems were associated with more than one item form and some problems were not associated with any item form. For example, the problem 153 - 92 cannot be uniquely classified in terms of the item forms of Hively et al. This problem is compatible with both the “borrow” and “no borrow” item forms. This lack of strict partition in the mathematical sense may very well have contributed to the Hively et al. finding that item forms did not represent homogeneous classes of items. In general, it is not an easy task to generate item forms which will partition a domain.

Furthermore, neither technology specifically takes into account the knowledge which makes it possible to solve problems belonging to a given domain. This is an important limitation because there can be any number of ways of solving problems within a domain. For example, there are several common rules a pupil may use to solve subtraction problems. His performance on such problems could be due to his mastery of any one of these rules. (Identifying what rules may be used on a domain of problems also has important implications for providing remediation, and more is said on this below.)

Scandura’s (1971, 1973) theory of structural learning provides a theoretical basis for an algorithmic technology for assessing behavior potential that deals directly with
the above problems. This theory consists of three hierarchically related partial theories: a theory of knowledge, a memory-free theory of learning and performance, and a theory of memory. For present purposes, two basic assumptions of the memory-free theory suffice. Stated simply, they are that people use rules to solve problems, and if an individual has learned a rule for solving a given problem or task, then he will use it.

To see how these assumptions are involved, notice that if an observer knows what rule or rules a subject has available for solving a given domain of problems, then he can predict perfectly the subject’s performance on problems in that domain. Unfortunately, the observer generally has no a priori way of knowing this. Nonetheless, with many familiar tasks (e.g., ordinary subtraction) there are a limited number of rules that subjects in a given population are most likely to use (e.g., the “borrowing” and “equal addition” methods for subtraction), and the first step in assessing behavior potential is for the observer-theorist to identify them.

It does not necessarily follow, of course, that every subject (or even any subject) will know any one of these rules completely. Rules consist of operations and branching decisions (i.e., subrules), which are performed in certain specified orders (see Scandura, 1973). The branching decisions of the rule serve to combine the operations in different ways for solving different kinds of problems. Thus a subject may know part of a rule or parts of several rules, and hence, may solve certain tasks governed by the rule(s) but not others. The object of testing is to determine from a subject’s performance on a limited number of problems what parts of the rule or rules he knows and what parts he does not know.

The operations and branching decisions of a rule (algorithm) can be described or listed in much the same way that one constructs a computer program, flow chart, or directed graph. From the flow chart one can see that there are a finite number of ways in which the subrules may be combined or sequenced to solve problems. These sequences of subrules, ignoring repetitions, are called paths and partition the domain of tasks governed by an algorithm into equivalence classes.

Consider, for example, the domain described by “Find sums (less than 100) for column addition using two or more addends of one digit.” An algorithm governing this domain may be characterized by the following program: (a) Add the top two addends; (b) if there are no other addends, go to c; otherwise go to d; (c) write the sum and stop; (d) add the units digit of the obtained sum to the next addend; (e) if the sum is greater than 10, go to f; otherwise go to g; (f) add 1 to whatever is in the tens place and return to b; and (g) return to b.

This program can be represented by a directed graph (see Figure 1) in which the numbered arcs correspond to the subrules and points to branching decisions (i.e., “if” statements).

From the graph it can be determined that there are four paths through the algorithm:

1. Path 1 is used to solve problems having only two addends.
2. Path 2 is used to solve problems having more than two addends but with intermediate sums less than 10 and the final sum less than 19.
3. Path 3 is used to solve problems having more than two addends, where successive sums increment the tens place.
4. Path 4 is used to solve problems having more than two addends, where the successive sums may or may not increment the tens place.

Each problem in the domain of an algorithm is solvable via exactly one of its paths. Hence, the paths partition the domain into equivalence classes (i.e., problems are equivalent if and only if they are solvable by the same path).

If the constituent subrules of an algorithm are atomic for a subject (i.e., if each is either “known” perfectly or not at all), then it follows logically that the paths of the algorithm will also act in atomic fashion. That is, if a subject is successful on any one item in an equivalence class, then he should also be successful on any other and similarly for failure. In effect, according to this theory, only one item is needed
from each equivalence class in order to test for mastery of the algorithm (i.e., assess a subject's behavior potential).

There may, of course, be more than one feasible algorithm underlying a domain of tasks, each of which in general will partition the domain differently. This slight complication can be easily handled by forming what we shall call an intersection partition on the given domain of tasks. The intersection partition is formed by selecting equivalence classes from one partition and taking their intersections with equivalence classes of other partitions. The collection of all possible nonempty intersections formed in this way generates the intersection partition. Generally speaking, the intersection partition is a finer partition of the domain than the partition associated with any one algorithm. Thus, to assess behavior potential simultaneously with respect to all of the identified algorithms, it will suffice in theory to (randomly) select just one item from each equivalence class belonging to the intersection partition.

In order for this assessment procedure to be applicable to a given population of subjects, the observer must assume that he has refined the algorithms to a point where the subrules are atomic for most of the subjects. According to the theory, this is always possible in principle because the subrules of an algorithm may be decomposed into ever finer subrules (i.e., refined). Indeed, rules can be reduced to associations such as \( 3 + 2 = 5 \) (Arbib, in press; Seandura, 1970, in press; Suppes, in press), which under the present memory-free conditions are necessarily atomic. It should be kept in mind, however, that unnecessary refinement of an algorithm results in a loss of testing efficiency. More test items are needed. In practice, the goal is to find some optimal level of refinement.

The algorithmic technology also provides a basis for ordering equivalence classes of
problems. Certain paths in an algorithm are superordinate to other paths in that they contain all of the atomic rules of the subordinate path plus some of their own. For example, Path 4 of the above algorithm is superordinate to Paths 1, 2, and 3, and both Path 2 and Path 3 are superordinate to Path 1. Since the branching decisions associated with a superordinate path encompass those associated with its subordinate paths, it follows that if a student can use a superordinate path, he should be able to use the subordinate paths. Hence, success on problems associated with a superordinate path should imply success on all problems associated with relatively subordinate paths.

Empirical support for the above analysis was obtained by Scandura and Durnin (reported in Scandura, 1971, 1973). In that study a variety of tasks was used and the subjects ranged in ability from preschool to graduate level. The atomic rules of each algorithm were "built in" (taught) to each subject and he was provided an opportunity to put the rules together to solve problems belonging to the domain of the algorithm. (The theory of structural learning accounts for the combining of subrules through the use of higher order rules, see Scandura, 1971.) Each subject was then tested on one item from each equivalence class associated with a path of the algorithm. Based on first test performance, predictions were made concerning performance on individual second test items. The results of the study showed that prediction of combined success—failure on second test items was possible with 96% accuracy ($\phi = .92$). Furthermore, it was found that in 95% of the cases where a subject was successful on a superordinate path, he was also successful on all subordinate paths.

In summary, the algorithmic approach deals directly with all of the questions raised earlier. It provides a theoretical basis for categorizing classes of problems and assures that this categorization partitions the domain of problems into equivalence classes. It also provides a theoretical basis for the hierarchical relationship between tasks and takes into account the different ways in which a domain of tasks may be solved. (The implication of this for task analysis is that there can be more than one way of hierarchically ordering problems within a given domain of tasks. In fact, there is a different hierarchy for each rule governing the domain.)

Granting the more rigorous theoretical foundations for the algorithmic technology, its pragmatic value relative to other existing technologies was still an open question. The objective of this study was to help clarify this issue. Specifically, we wanted to determine whether or not the algorithmic approach to partitioning, which is based on an intuitive analysis of intensity (i.e., analysis in terms of underlying rules or procedures), provides any practical advantage for mastery testing, relative to the extentional technologies of Hively et al. (1968) and Ferguson (1969). The domain of column subtraction problems was chosen for the comparison.

In effect, the different way in which each technology subdivided the domain of subtraction problems was the independent variable. Item forms are based on extentional analysis of the kinds of problems involved. Ferguson's approach is just another version of item forms which leaves some of the complex problems out of the analysis. The algorithmic analysis, on the other hand, is in terms of what the subject has to do, that is, the processes he must go through in order to solve the problems. This form of intensional analysis frequently combines classes that may on the surface appear to be different. For example, the path of the subtraction algorithm in Figure 3 governs both problems shown in the figure, but according to item form analysis the problems belong to two different classes.

For the purposes of this study, practical advantage meant one or more of the following criteria were met: (a) an improvement in predictions concerning the performance of individual subjects on particular kinds of test items, (b) an improvement in the degree of generalizability (from test items to a clearly specified domain), (c) a reduction in the number of test instances required to determine behavior potential, and (d) an improvement in the hierarchical ordering of
ALGORITHMIC APPROACH TO BEHAVIOR POTENTIAL

Method

The algorithmic technology was used to construct four algorithms for column subtraction. Two algorithms were based on a borrowing procedure for subtraction and consisted of six and five paths, respectively. The other two algorithms were based on an equal additions procedure and consisted of four and eight paths, respectively. The intersection partition with respect to all four algorithms was then constructed. It contained 12 equivalence classes. The flow chart of the subtraction algorithm shown in Figure 2 was designed explicitly to have a path corresponding to each and every equivalence class in the intersection partition. For example, the path of the algorithm in Figure 3 determines correct solutions for the class of subtraction problems having more than one column and using facts \( \leq 9 \).

Hively et al. (1963) used an item forms analysis of subtraction problems to identify 28 subclasses of problems. Of these 28 subclasses, 22 pertained to column subtraction. With the exception of "Large numbers," which was omitted from consideration because it included several of the other categories (e.g., "Borrow one digit from large number," "Repeated borrows," "Separated borrows," etc.), the item forms were interpreted to represent mutually exclusive classes of problems.

By taking intersections of the 21 item forms with the 12 equivalence classes generated by the algorithmic approach, 37 new classes of subtraction problems were obtained. Prediction and criterion tests, parallel Tests A and B, respectively, were constructed by generating two arbitrary items for each of the 37 classes in the intersection set obtained from item forms and equivalence classes, one for each test. The order of items was randomized in each test.

Fig. 2. Subtraction algorithm.

Fig. 3. Sample path through subtraction algorithm and two stimulus instances from the corresponding equivalence class.
Subjects and Procedures

The subjects were 34 ninth-grade general mathematics students attending summer school at Shaw Junior High School, Philadelphia. Tests A and B were administered to the subjects in their classrooms on consecutive days. The order in which the tests were given was counterbalanced over subjects. Of the 34 subjects, 25 were in attendance both days and received both Tests A and B.

Analysis

Since Ferguson (1969) in his analysis identified hierarchical forms involving only simple subtraction problems (i.e., numbers with three or fewer digits), comparison of the assessment procedures was done in two parts: (a) for the entire domain of column subtraction problems and (b) for a restricted domain of subtraction problems, comparable to Ferguson's hierarchical forms. The restricted domain consisted of classes of problems in the intersection set associated with 7 of the 12 equivalence classes and 13 of the 22 item forms.

In order to compare the item forms and algorithmic approaches on the unrestricted domain of subtraction problems, two imbedded subtests were designated for each technology: one from Test A and one from Test B. With the algorithmic technology, for example, this was done by randomly selecting one test item for each equivalence class from those items in Tests A and B, respectively, that belonged to that equivalence class. This was possible because the 12 equivalence classes partitioned the 37 items in each test into 12 mutually exclusive and exhaustive classes. The other A and B tests were constructed similarly using item forms.

To compare performance on the restricted domain, a pair of similar, imbedded subtests was designated for each technology (restricted intersection, algorithmic, hierarchical forms, and item forms) from problems belonging only to the restricted classes of items in Tests A and B.

Performance on the unrestricted subtests provided the basic data for comparison of the algorithmic and item forms technologies for the unrestricted domain of subtraction problems. Performance on the restricted subtests provided the basic data for comparison of the algorithmic, item forms, and hierarchical forms technologies on the restricted domain of subtraction problems.

RESULTS AND DISCUSSION

Levels of Predictability

Table 1 shows the levels of predictability and correlation between items belonging to the same class for each of the various types of tests on the unrestricted domain of subtraction problems.

In regard to the first criterion, the overall levels of predictability on individual items were approximately the same for all unrestricted tests. However, the correlation (phi coefficient) of .53 between corresponding Test A and Test B items, involving the $2 \times 2$ success-failure matrix for the 300 equivalence class instances, was significantly greater ($p < .05$) than the correlation of .39 between corresponding items for item form instances. This correlation for equivalence classes was also higher, although not significantly so, than that for the intersection of equivalence classes and item forms (.40).

The difference in correlations between equivalence classes and item forms was due to the significantly higher ($p < .05$) levels of predictability for equivalence classes for those Test A items on which subjects were not successful. Furthermore, the level of predictability for those Test A items on which subjects were not successful was also significantly greater ($p < .05$) for equivalence classes than for the intersection of item forms and equivalence classes. This

<table>
<thead>
<tr>
<th>TABLE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numbers of Items, Percentage of Correct Predictions, and Correlations Between Corresponding Items</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test</th>
<th>Number of items</th>
<th>Number of Test A instances on which subjects were successful</th>
<th>Percentage of correct predictions</th>
<th>Number of Test A instances on which subjects were not successful</th>
<th>Percentage of correct predictions</th>
<th>Total number of Test A instances</th>
<th>Percentage of correct predictions</th>
<th>Correlation between corresponding A and B test instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intersection</td>
<td>37</td>
<td>690</td>
<td>91%</td>
<td>226</td>
<td>55%</td>
<td>925</td>
<td>82%</td>
<td>.49</td>
</tr>
<tr>
<td>Item forms</td>
<td>21</td>
<td>444</td>
<td>89%</td>
<td>233</td>
<td>51%</td>
<td>552</td>
<td>83%</td>
<td>.39</td>
</tr>
<tr>
<td>Equivalence</td>
<td>12</td>
<td>225</td>
<td>85%</td>
<td>75</td>
<td>71%</td>
<td>300</td>
<td>82%</td>
<td>.53</td>
</tr>
</tbody>
</table>

*$p < .05$ for bracketed pairs of numbers.
latter result must be tempered, however, because the difference in levels of predictability between the intersection and equivalence classes for those Test A items on which subjects were successful was also significant \((p < .05)\). (The corresponding difference between equivalence classes and item forms was not significant.)

In effect, the test constructed on the basis of the algorithmic technology with approximately 57% as many items (12 as compared to 21) gave better predictions on individual items than the corresponding test for item forms. Furthermore, tests formed from the two algorithms based on borrowing had 65% and 75% levels of prediction, when subjects were unsuccessful on Test A items with overall levels of predictability at 78%. These levels of prediction were obtained with only 6 and 5 items for the respective tests. Hence, with considerably fewer items these tests were not only as effective in overall predictability as the intersection and item forms tests but also had higher (and for the 5-item test, significantly higher, \(p < .05\)) levels of predictability than the item forms test for those Test A items on which subjects were unsuccessful.

It is also worth noting, that of the four algorithms originally identified, the two based on borrowing had significantly higher \((p < .05)\) levels of prediction than the two algorithms based on equal additions on which subjects were unsuccessful on Test A items (65% and 75% as compared to 29% and 32%). The implication of this, of course, is that for these subjects the tests formed from algorithms based on borrowing were better predictors than the tests formed from algorithms based on equal additions. This difference between the two types of subtraction appears to reflect the fact that borrowing is the more common procedure taught in American schools.

Components of variance analysis (Winer, 1962, pp. 184–191) are also relevant to criterion one. For example, although the interaction of subjects by items within classes contributed most of the variance for each of the three types of tests on the unrestricted domain, the contribution was lowest for equivalence classes (40% vs. 63% for item forms). Furthermore, the sources of variance due to classes and subjects by classes were greater for equivalence classes than item forms (22% vs. 15% and 27% vs. 15%, respectively). These results tend to confirm the previous finding that even with fewer items, the algorithmic approach was more sensitive than the item forms technology in pinpointing strengths and weaknesses of individual students.

As regards the restricted domain, none of the obtained results concerning the levels of predictability or correlation was significantly different. Restricting the domain, however, had the effect of increasing overall predictability for each technology. The obtained levels of predictability were practically identical, ranging only 89%–91%. Since most of the problems in the restricted domain appeared to be relatively easy for the subjects, the levels of predictability for “success” items were even higher (93%–95%). The relatively small number of errors involved overall suggests that the low levels of predictability for items on which subjects were not successful (22%–41%) may have been due to careless mistakes.

Components of variance could not be obtained for most of the tests in regard to the restricted domain because estimates of variance due to items within classes were negative for all restricted tests except item forms. In that case, the contribution of variance due to persons by items within item forms was 77%.

**Generalizability Results**

In regard to the second criterion, Table 2 shows the coefficients of generalizability \(\alpha\) and \(\alpha^*\) for each type of test. The coefficient \(\alpha^*\) is a lower bound estimate of how well one can generalize from a subject’s obtained score on a test to his performance on the stated domain of items (Cronbach et al., 1963), in this case column subtraction problems. It is also an intraclass correlation coefficient for estimating reliability (Winer, 1962, pp. 124–132). The coefficient \(\alpha^*\) (Rajaratnam, Cronbach, & Gleser, 1965) is an estimate of generalizability for stratified parallel tests, tests for which the domain of
items is divided into different classes, as was the case in this study.

The top half of Table 2 shows the coefficients of generalizability for the unrestricted domain of subtraction problems. Of these, the intersection test provided the highest estimates of generalizability; those for equivalence classes were next; and item forms last. Again, it is of interest to note that the two subtests formed from borrowing algorithms had levels of generalizability as high as the subtest formed from item forms. For the test with 6 items, \( \alpha' = .75, \alpha'_g = .60 \) and for the test with 5 items, \( \alpha' = .84, \alpha'_g = .62 \).

On the restricted domain of subtraction problems, the coefficients shown in the lower half of Table 2 for the restricted intersection, restricted item forms, and restricted equivalence classes were greater than the coefficients for hierarchical forms.

The values of \( \alpha' \) and \( \alpha'_g \) obtained for the restricted tests were not the same as those obtained for the unrestricted tests \( (\chi^2 = 20.6, df = 6, p < .01; \chi^2 = 26.19, df = 6, p < .01) \). In effect, a subject's score on a restricted test, and in particular on the test generated by hierarchical forms, could not viably be generalized to the entire domain of column subtraction problems. Hence, although the overall levels of predictability for these tests were higher than those generated from the unrestricted domain, the above results indicate that this was accompanied by a significant loss in generalizability.

Efficiency Criterion

The data clearly show that the algorithmic approach was more efficient than the item forms technology. Only 12, as compared to 21, items were required to achieve about the same overall level of predictability and somewhat better levels of generalizability. The increase in efficiency evident with the tests formed from the two borrowing algorithms is even more striking. With only 6 and 5 items, respectively, they had essentially the same levels of predictability and generalizability as the item forms test with 21 items.

Furthermore, although it seems reasonable to suppose that the intersection test with 37 items would produce the highest levels of predictability and generalizability, in general this was not the case. With a third (12 as compared to 37) as many items, the algorithmic approach maintained as high a level of overall predictability and only slightly (nonsignificantly) lower levels of generalizability. The item forms test, which had slightly more than half the number of items as the intersection test, also obtained as high a level of predictability, although somewhat lower levels of generalizability. Overall, these results lead one to suspect that under the testing conditions used, the algorithmic approach for assessing mastery approaches asymptote. Further improvement would almost necessarily require more rigorous testing conditions (cf. Scandura, 1973).

Even on the restricted domain the equivalence classes test appeared to be the most efficient. Overall levels of predictability were the same for all tests, while generalizability coefficients were somewhat higher for the equivalence class and item forms tests. These higher levels of generalizability, however, were obtained with half as many items in the case of the equivalence classes test.

Hierarchical Analyses

The fourth criterion is concerned with the fact that efficiency may sometimes be increased through the use of conditional testing procedures, at least where the various items lend themselves to Guttman-type (1947) scaling. In the present study, however, it must be noted that each of the technologies compared provides an explicit basis
for ordering items that is independent of empirical data (cf. Ferguson, 1969; Hively et al., 1968).

The method of analysis used to determine the relative validity of the three hierarchies was similar to that used by Gagné (1962) to confirm relationships between higher and lower levels in task analysis. Cases in which success on a pair of problems associated with a superordinate class implies uniform success on all problems associated with relatively subordinate classes and cases in which failure on at least one of the problems in a superordinate class is implied by failure on at least one of the relatively subordinate classes validate a hierarchy. Cases in which success on a superordinate class fails to indicate success on all relatively subordinate classes contradict a hierarchy. The proportion of verifying cases to the number of verifying plus contradictory cases was .82 for the equivalence classes hierarchy as compared to .74 for the item forms hierarchy (p < .01). None of the differences on the restricted domain were significant.

To summarize, then, the algorithmic approach not only provided the best and most efficient method for assessing behavior potential, but the hierarchy induced by the approach could be used to increase this efficiency even more through the use of conditional testing procedures which involve branching (with or without computer assistance).

**Implications**

On almost all measures obtained, the algorithmic approach to assessing behavior potential proved to be either better or at least as good as the technologies based on item forms or hierarchical analysis. Nonetheless, at first thought the item forms technology might appear to have a certain advantage over the algorithmic approach. Given an item form, it is a routine matter to generate an instance of that item form. This could be particularly useful in computer-assisted testing, since the computer could be programmed to randomly generate test items within forms. (The item forms themselves, however, must be determined directly by the test constructor.)

In the algorithmic approach this would have to be done indirectly. Nonetheless, the computer, once given an algorithm, could be programmed to automatically trace out the paths, identify the equivalence classes, and order the items for testing. That is, the computer should be able to generate not only the items but also the item forms (i.e., equivalence classes) themselves.

Moreover, on further reflection, it becomes apparent that the more circuitous route required for generating test items via the algorithmic approach has a further major advantage. It provides an explicit basis for remedial instruction. To see this, we assume, in accordance with Scandura's theory, that subjects actually use rules (algorithms) to generate their behavior. Then, because each equivalence class of items corresponds to a unique path of a rule and because the steps in each such path are known explicitly to the instructor (or computer), each pupil can be given specific instruction to overcome his inadequacies. Put succinctly, he can be taught the needed paths. These ideas constitute the theoretical basis for a series of self-diagnostic and remedial workbooks developed by the Mathematics Education Research Group (e.g., Scandura, 1972) and could be extended for use in computer-assisted testing and instruction.  

1 After this article was prepared, the second author had the pleasure of reading John Bormuth's (1970) fine book on test construction and was delighted to find so many compatible ideas in it. Bormuth argued that traditional methods of test construction suffer serious limitations and proposed a new approach in which test items are operationally defined in terms of instructional material. For example, an instruction like, "George Washington was the first president of the United States," can be converted into the "who" item, "who was the first president of the United States?"

Bormuth's method of operationally defining test construction is in some ways similar to that proposed by Scandura (1972) and developed here. Both methods provide a systematic way to construct (identify) test items, the former from verbal instruction and the latter from procedures. An even closer relationship becomes apparent if one accepts the view that the meanings of verbal instructions can be represented as to-be-learned pro-
REFERENCES


(Received for early publication October 10, 1972)