DOCUMENT RESUME

ED 064 896 EM 009 965

AUTHOR	Durnin, John H.; Scandura, Joseph M.
TITLE	An Algorithmic Approach to Assessing Behavior Potential: Comparison With Item Forms and Hierarchical Technologies.
SPONS AGENCY	National Science Foundation, Washington, D.C.; Office of Education (DHEW), Washington, D.C.
PUB DATE	72
GRANT	OEG-3-71-0136
NOTE	47p.; Based on thesis submitted to the University of Pennsylvania
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	Branching; *Computer Assisted Instruction; *Criterion Referenced Tests; *Individualized Instruction; Programed Materials; *Test Construction

ABSTRACT

For individualized or computer assisted instruction, norm referenced testing is inadequate to determine each individual's mastery on specific kinds of tasks. Hively's item forms and Ferguson's stratified item forms, both based on observable characteristics of the problems, and Scandura's algorithmic technology, positing that persons use rules to solve problems and thus that problems should be partitioned on the basis of rules needed to solve them, have been developed to measure individual mastery. This study was designed to compare their effectiveness and efficiency in assessing mastery of column subtraction problems. All three methods were essentially equal in predicting mastery of individual items, but the algorithmic method used far fewer items and thus was more efficient. The item forms technology would seem to have a slight advantage in the ease with which a computer could randomly generate test items, but even items for the algorithmic form can be computer generated, although slightly indirectly. (RH)

TO: LEASCO

FROM: ERIC/EM

This document was transferred from TM to CG who, in turn, transferred it to EM. The OE funding was provided for the thesis on which this work is based, therefore we included the OEG number on the cataloging. If you think it should be deleted, please do so.

Compliments of

ERIC

Clearinghouse on Educational Media and Technology

at the Institute
for Communication Research
Stanford University
Stanford, California 94305

CG
N-NPO

An Algorithmic Approach to Assessing Behavior Potential: Comparison with Item Forms and Hierarchical Technologies[1]

John H. Durnin and Joseph M. Scandura

University of Pennsylvania

Recent research in individualized (e.g., Lipson, 1967) and computer assisted (e.g., Suppes, 1966) instruction has led to an increasing awareness of the inadequacies of norm referenced testing and the need for testing procedures which 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, in mastery testing the procedures used should 1) provide a sound basis for diagnosing individual strengths and weaknesses on each type of task, 2) require as few items as possible, and 3) provide a basis for generalizing from overall test performance to behavior on a clearly defined universe or domain of tasks.

---

[1]This article is based on a Ph.D. dissertation submitted by the first author under the second author's chairmanship to the University of Pennsylvania. This study was supported by U.S. Office of Education Grant 3-71-0136 and, in part, by National Science Foundation Grant GW6796, both to the second author. continued

If, in addition, items can be ordered according to difficulty 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; Johnson, 1970; Nitko, 1970; Osburn, 1968; Rabehl, 1970; Roudabush & Green, 1971; Scandura, 1971a, 1972). 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 (1971a, 1972).

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., $\frac{x}{-y}$), one or more elements (e.g., $\frac{42}{-21}$, $\frac{28}{-16}$), and explicit criteria for specifying which elements belong to the form (e.g., $x = x_1 x_2$; $y = y_1 y_2$; $y_1 < x_1$; $y_2 < x_2$; $x_1, x_2, y_1, y_2 \in \{0,1,2,\ldots, 9\}$). 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

---

The authors thank Alfonso Georeno and David Shore for their cooperation in providing subjects. The authors would also like to thank Frederick Davis, James Diamond, Zoltan Domotor and Albert Oliver for helpful comments on an earlier version of this paper.

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. (1968) were able to obtain high coefficients of generalizability (Cronbach, Rajaratnam, & Gleser, 1963; Rajaratnam, Cronbach, & Gleser, 1965) for tests based on the item forms technology, they did not find that item forms, in general, represented homogeneous categories of problems of the type described above.

One criticism of the item forms technology has been that the hierarchical relationships among item forms have not been taken into account in testing (e.g., Nitko, 1970). 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., Gagne, 1962). Starting with a terminal item form, corresponding to a terminal instructional objective, sub-item forms (i.e., subobjectives) were identified which were considered prerequisite to the terminal item form. 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 one fundamental criticism. There is no real theoretical basis for either technology. With the possible exception of Ferguson's hierarchical ordering of forms, which is based essentially on task analysis, there is 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). This lack of partition may very well have contributed to Hively *et al.*'s 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. Also, once a set of item forms has been generated, it is difficult to determine whether or not the item forms do indeed form a partition.

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 (1971a, 1972) theory of structural learning provides a theoretical basis for an algorithmic technology to assessing behavior potential which 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 that 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 is 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, 1970b, 1971a). 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.

Now the operations and branching decisions of a rule can be described or listed in much the same way that one constructs a computer program. (An alternative description is a flow chart. When discussing rules in which the operations and branching decisions are made explicit in either of these two ways, the term algorithm is used.) From the list or program one can see that there are a finite number of ways in which the subrules may be combined or sequenced to solve problems.[2] These sequences of subrules, called paths, 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."[3] An algorithm governing this domain may be characterized by the following program:

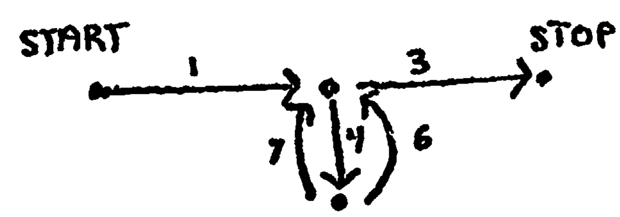
---

[2]Some of the sequences involve cycles or loops in which the same subrules may be repeated indefinitely. Each traversal through a loop, of course, generates a new extended sequence of the same subrules. However, because no new subrules are added or deleted, these sequences are considered equivalent.

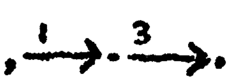
[3]This description of a class of tasks was adapted from a list of objectives for the Individualized Prescribed Instruction Program at the University of Pittsburgh's Learning Research and Development Center, September, 1965.

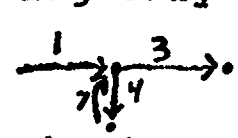
1. Add the top two addends.
2. If there are no other addends, go to 3; otherwise go to 4.
3. Write the sum and stop.
4. Add the units digit of the obtained sum to the next addend.
5. If the sum is greater than 10, go to 6; otherwise go to 7.
6. Add 1 to whatever is in the tens place and return to 2.
7. Return to 2.

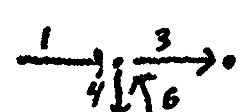
This algorithm can be represented by a directed graph in which the numbered arcs correspond to subrules and points to branching decisions (i.e., "if" statements) as follows:

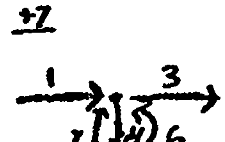


From the graph it can be determined that there are four paths (i.e., sequences of subrules) through the algorithm.

a. Path 1,  , is used to solve problems having only two addends (e.g., $\begin{array}{r}2\\+6\end{array}$).

b. Path 2,  , is used to solve problems having more than two addends but with intermediate sums less than ten and the final sum less than nineteen (e.g., $\begin{array}{r}2\\3\\4\\+9\end{array}$).

c. Path 3,  , is used to solve problems having more than two addends where successive sums increment the tens place (e.g., $\begin{array}{r}6\\9\\8\\+7\end{array}$).

d. Path 4,  , is used to solve problems having more than two addends where the successive sums may or may not increment the tens place (e.g., $\begin{array}{r}8\\5\\3\\+9\end{array}$).

It is easy to see from this example, then, that paths partition the domain governed by an algorithm into equivalence classes. That is, two problems are equivalent if and only if they are solvable by the same path through the algorithm.

If the constituent subrules of an algorithm are atomic (i.e., a subrule can be used by a subject on all or none of its instances) for any given subject, then it follows logically that the paths of the algorithm will also be atomic. This implies that if the subject is successful on any one item of an equivalence class, then he should be successful on any other and similarly for failure. Hence, to assess his behavior potential all that is needed is one item from each equivalence class.

As was mentioned earlier, of course, there may be more than one feasible algorithm underlying a domain of tasks. If several algorithms are identified, then it is likely that some of these algorithms will partition the domain differently. This slight complication can be easily handled, however, by forming what we shall call an <u>intersection partition</u> on the given domain of tasks. The intersection partition is formed by selecting one equivalence class from each partition and taking their intersection. The collection of all possible non-empty intersections[4] formed in this way generates the intersection partition. Generally,

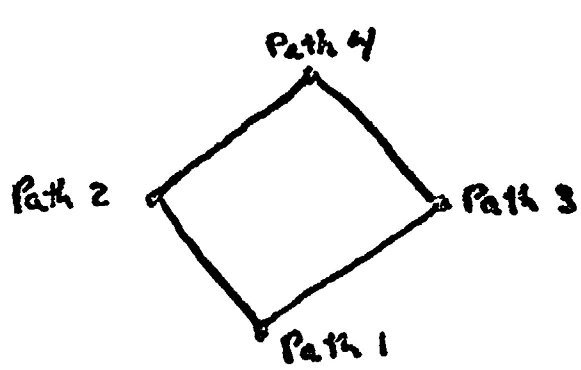
---

[4]To see in more detail how these intersections may be obtained, let A_{i_k} represent an equivalence class associated with path i of algorithm k. The collection of intersection sets for n algorithms can be generated by taking $A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_k} \cap \cdots \cap A_{i_n}$ where the i_k vary over all paths of the algorithms. If there are m_k paths per algorithm, then there can be at most $\prod_{k=1}^{n} m_k$ non-empty intersections.

the intersection partition is a finer partition of the domain than the partition associated with any one algorithm. To assess behavior potential simultaneously with respect to all of the identified algorithms, one item from each equivalence class belonging to the intersection partition is randomly selected for testing.

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. Indeed, rules can be reduced to associations (Arbib, 1969; Scandura, 1970a, 1970b, 1972; Suppes, 1969), which under memory-free conditions are necessarily atomic. Although this can always be done for a given population, what is gained at this level of atomicity is lost in 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 classes of problems according to difficulty. 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 (e.g., path 4 of the above algorithm is superordinate to paths 1, 2, and 3). Since the superordinate path is more difficult (on the basis of having more constituent rules) than a subordinate path, and since the branching decisions in the superordinate path account for all performance capable by means of the subordinate path, it follows that if a subject can use the superordinate path, he should also be able to use the subordinate path. Hence, success on problems associated with a superordinate path should imply success on all problems associated with relatively subordinate paths. An example of this partial hierarchical ordering is the following lattice representing the ordering of paths for the above algorithm.



Empirical support for the above analysis was obtained by Scandura and Durnin (reported in Scandura, 1971a, 1972). In that study a variety of tasks were used and the subjects ranged in ability from preschool to graduate level. The atomic rules of an algorithm were given or "built into" 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, 1970a).] 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 and failure on second test items was possible with 96% accuracy.[5] 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.

To determine the accuracy of the above analyses under classroom conditions an exploratory study was conducted in which the atomic rules of the algorithms were assumed rather than "built into" the subjects.

Forty four subjects in two first year highschool

---

[5] The correlation between corresponding items was .92.

algebra classes were given two tests on factoring monic trinomials shortly after they had completed a unit on that topic. The tests were devised by first identifying the procedure used in the text and determining those rules which the author of the text assumed the students knew (i.e., that were atomic) and, then, constructing two sets of test items corresponding to each path in the procedure.

As in the previous study first test performance was used to predict second test performance. The results of the study showed that prediction on individual second test items was possible with 86% accuracy.[6] And in 87% of the cases where a subject was successful on a superordinate path he also was successful on all subordinate paths.

By way of summary, it is important to notice that the algorithmic approach to assessing behavior potential 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, of course, is that there can be more than one way of hierarchically ordering problems within a given domain of tasks.

[6] The correlation between corresponding items was .60.

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 assessing behavior potential was an improvement over the technologies developed by Hively _et al_. (1968) and Ferguson (1969). The domain of column subtraction problems was chosen for the comparison because of the availability in the literature of relevant information (i.e., Hively _et al_., 1968; Ferguson, 1969).

For the purposes of this study, improvement meant one or more of the following:

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 tasks (with its important implications for conditional testing).

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 6 and 5 paths, respectively. The other two algorithms were based on an "equal additions" procedure and consisted of 4 and 8 paths, respectively. The intersection partition with respect to all four algorithms was then constructed (see footnote 4). It contained 12 equivalence classes. The flow chart of the subtraction algorithm shown in Figure 1 was designed explicitly to have a path corresponding to each and every equivalence class in the intersection partition.

Insert Figure 1 about here

The directed graph, the twelve possible paths, and items from corresponding equivalence classes of the subtraction algorithm of Figure 1 are shown in Figure 2. The numbered arcs in the graph and paths correspond to rules in the flow chart and the points to the initial (START), terminal (STOP) and branching rules of the flow chart.

Insert Figure 2 about here

Hively _et al_. (1968) used an item forms analysis of subtraction problems to identify 28 subclasses of problems. Of these 28 subclasses, the following 22 pertained to column subtraction:

1. Basic fact; minuend $\leq$ 10
2. Subtract 0

3. Answer = 0
4. Basic fact; minuend > 10
5. No borrow; no 0 in answer or problem
6. No borrow; x-0 fact in problem
7. No borrow; 0-0 fact in problem
8. No borrow; x-x fact in problem
9. No borrow; small; unequal lengths
10. No borrow; large; unequal lengths
11. Simple borrow
12. Simple borrow; one digit subtrahend
13. Simple borrow; one digit answer
14. Simple borrow; medium
15. Borrow; one digit from large number
16. Borrow; medium; subtrahend one digit short
17. Borrow; medium; unequal lengths
18. Separated borrows
19. Repeated borrows.
20. Borrow across 0
21. Borrow across two (or more) 0's
22. Large numbers

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 in the above list were interpreted so as to represent mutually exclusive classes of problems.[7]

By taking intersections of the 21 item forms with the 12 equivalence classes generated by the algorithmic approach, 37 new classes of subtraction problems, shown in Table 1, were obtained.

Insert Table 1 about here

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.

Subjects and Procedures. The subjects were 34 ninth grade general mathematics students attending summer school at Shaw Junior High School in 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 of Results. Since Ferguson (1969) in his analysis on-

---

[7]There was one ambiguous class of problems (e.g., $\frac{153}{-92}$) which may be interpreted as borrow or no borrow depending upon how one considers the problem. Also, some of the item forms (i.e., classes of problems defined by the item forms) are properly contained in other item forms. For example, "Borrow; medium; subtrahend one digit short" is properly contained in "Borrow; medium; unequal lengths." In this case, unequal lengths was taken to mean that the minuend contained two or more digits more than the subtrahend.

In effect, using mutually exclusive item forms had the effect of improving the level of item forms predictions by 1% so the present study provides a more conservative comparison as regards the algorithmic approach.

ly identified hierarchical forms (see Fig. 3) involving three or fewer digit numbers, comparison of the assessment procedures was done in two parts (1) for the entire domain of column subtraction problems and (2) for a restricted domain of subtraction problems, comparable to Ferguson's hierarchical forms. The restricted domain consisted of classes of problems (marked by * in Table 1) in the intersection set associated with the first seven equivalence classes and the thirteen item forms, 1-9, 11-13, and 19, pertaining to basic facts and no borrow (minus large lengths), simple borrow, and repeated borrow, respectively. Parallel tests, A' and B', were constructed for the restricted domain by deleting from tests A and B items from those classes of problems not marked by an asterisk.

In order to compare the item forms and algorithmic approaches on the unrestricted domain of subtraction problems, two subtests were constructed for each technology, one from test A and the other from test B. This was done for each technology by randomly taking one test item from each class of items associated with an item form or equivalence class.

To compare performance on the restricted domain, a pair of similar subtests was constructed from the restricted tests A' and B' for each technology (algorithmic, hierarchical forms, and item forms).

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 2 shows the levels of predictability and correlation between items belonging to the same class for each of the various types of tests. The top half of Table 2 shows the levels of predictability for tests measuring performance on the unrestricted domain of subtraction problems.

Insert Table 2 about here

In regard to the first criterion (p. 14), the overall levels of predictability on individual items were approximately the same for all unrestricted tests. However, the correlation between corresponding test A and test B items for equivalence classes, .53, was significantly greater ($p < .05$, Edwards, 1966, p. 82) than the correlation, .39, between corresponding items for item forms. This correlation for equivalence classes was also higher, although not significantly so, than that for the intersection of equivalence classes and item form (.49).

The difference in correlations between equivalence classes and item forms was due to the significantly higher ($p < .05$, Edwards, 1966, p. 53) 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 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" (see p. 16) had 65% and 75% levels of prediction where 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 where subjects were unsuccessful.

It is also worth noting that of the four algorithms (see p. 16) originally identified, the two based on "borrowing" had significantly higher ($p < .05$) levels of prediction than the two algorithms based on "equal additions" where 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.

The components of variance (Winer, 1962, pp. 184-191) shown in Table 3 are also relevant to criterion one (p. 14). Consider the contribution of variance due to the interaction of subjects by items within classes. Although this source contributed most of the variance for each of the three types of test on the unrestricted domain, the contribution was lowest for equivalence classes. Furthermore, the sources of variance due to classes and subjects by classes were greater for equivalence classes than item forms. 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.

Insert Table 3 about here

The levels of predictability and correlation associated with the restricted domain are shown in the lower half of Table 2. None of the obtained results was significantly different. Restricting the domain, however, had the effect of increasing overall predictability for each technology. 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 quite high. The relatively small number of errors involved overall suggests that the low levels of predictability for items on which subjects were not successful 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 (p. 14), Table 4 shows the coefficients of generalizability α' and α'_s for each type of test.[8] The coefficient α' 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 α'_s (Rajaratnam, et al., 1965) is an estimate of generalizability for stratified parallel tests, tests for which the domain of items

---

[8] α' and α'_s are estimates of generalizability from a single test to a well-defined domain of items and correspond to Cronbach's (1951) α and Rajaratnam et al.'s (1965) α_s, respectively, which are estimates of generalizability from the mean of two or more parallel tests (to a well-defined domain).

is divided into different classes as was the case in this study.

Insert Table 4 about here

The top half of Table 4 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 $\alpha' = .75$; $\alpha'_s = .60$, and for the test with 5 items $\alpha' = .64$; $\alpha'_s = .62$.

On the restricted domain of subtraction problems, the coefficients shown in the lower half of Table 4 for the restricted intersection, restricted item forms, and restricted equivalence classes were greater than the coefficients for hierarchical forms.

The values of α' and α'_s obtained for the restricted tests were not the same as those obtained for the unrestricted tests ($\chi^2 = 20.6$, 6df, $p < .01$; $\chi^2 = 26.19$, 6df, $p < .01$, Edwards, 1966, p. 83). 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.

<u>Efficiency Criterion</u>. 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 & Durnin in Scandura, 1972).

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 (p. 14) 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 (1947) type 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.

Figures 3, 4 and 5, respectively, show the various hierarchies (partial orderings) proposed for hierarchical forms (Ferguson, 1969), item forms (Hively et al., 1968), and the algorithm of Figure 1.

Insert Figures 3, 4 and 5 about here

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.

In Table 5, the positive-positive (++) superordinate-subordinate relationship shows for each hierarchy the number of cases where uniform success on the two superordinate problems associated with a class implied uniform success on all problems associated with relatively subordinate classes. The (--) superordinate-subordinate relationship shows the number of cases where failure on at least one of the superordinate problems in a superordinate class implied failure on at least one of the relatively subordinate classes. The (+-) superordinate-subordinate relationship shows the number of cases where success on a superordinate class failed to indicate success on all relatively subordinate classes. The (-+) superordinate-subordinate relationship shows the number of cases where there was uniform success on all subordinate classes but not on the relatively superordinate class.

Insert Table 5 about here

The ++ and -- relations, therefore, validate an ordering whereas the +- relation contradicts one. The -+ relation is considered neutral.

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 (e.g., Shoemaker and Osburn, 1969; Ferguson, 1969), 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 of problems, randomly generate test items in 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 (1971a, 1971b, 1972) 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 tapes and workbooks developed by the Mathematics Education

Research Group (e.g., Scandura, 1970c; Scandura, Gramick & Durnin, 1971) and could be extended for use in computer assisted testing and instruction.

References

Arbib, M.A. Memory limitations of stimulus-response models. Psychological Review, 1969, 76, 507-510.

Coulson, J.E. & Cogswell, J.F. Effects of individualized instruction on testing. Journal of Educational Measurement, 1965, 2, 59-64.

Cronbach, L.J. Coefficient alpha and the internal structure of tests. Psychometrika, 1951, 16, 297-334.

Cronbach, L.J., Rajaratnam, N., & Gleser, G.C. Theory of generalizability: A liberalization of reliability theory. The British Journal of Statistical Psychology, 1963, 16, 137-163.

Davis, F.B. Educational measurements and their interpretation. Belmont, Calif.: Wadsworth, 1964.

Edwards, A.L. Experimental design in psychological research. New York: Holt, Rinehart, & Winston, 1966.

Ferguson, R.L. Computer-assisted criterion-referenced measurement. Unpublished manuscript, Learning Research and Development Center, University of Pittsburgh, 1969.

Gagné, R.M. The acquisition of knowledge. Psychological Review, 1962, 59, 355-365.

Guttman, L. The Cornell technique for scale and intensity analysis. Educational and Psychological Measurement, 1947, 7, 247-280.

Hively, W. II, Patterson, H.L., & Page, S. A "universe defined" system of arithmetic achievement tests. Journal of Educational Measurement, 1968, 5, 275-290.

Johnson, P.E. The origin of item forms. Paper presented at the Annual Meeting of the American Educational Research Association, Minneapolis, March, 1970.

Lipson, J.I. Individualized instruction in elementary mathematics. In J.M. Scandura (Ed.) Research in mathematics education. Washington: National Council of Teachers of Mathematics, 1967. Pp. 70-79.

Nitko, A.J. Some considerations when using a domain-referenced system of achievement tests in instructional situations. Paper presented at the Annual Meeting of the American Educational Research Association, Minneapolis, March, 1970.

Osburn, H. G. Item sampling for achievement testing. Educational and Psychological Measurement, 1968, 28, 95-104.

Rabehl, G. J. The MINNEMAST experiment with domain referenced achievement testing. Paper presented at the Annual Meeting of the American Educational Research Association, Minneapolis, March, 1970.

Rajaratnam, N., Cronbach, L. J., & Gleser, G. C. Generalizability of stratified-parallel tests. Psychometrika, 1965, 30, 39-56.

Roudabush, G. & Green, D. R. Some reliability problems in a criterion referenced test. Paper presented at the Annual Meeting of the American Educational Research Association, New York, February, 1971.

Scandura, J. M. Role of rules in behavior: Toward an operational definition of what (rule) is learned. Psychological Review, 1970, 7[illegible] 516-533. (a)

Scandura, J. M. Theoretical note: S-R theory or automata? ([illegible] Suppes' reaction). Report No. 58, September 10, 1970, St[illegible] Learning Series. Mathematics Education Research Group, Graduate School of Education, University of Pennsylvania, Philadelphia, Pennsylvania. (b)

Scandura, J. M. A plan for the development of conceptually based mathematics curriculum for disadvantaged children. Report No. 59, October 15, 1970, Structural Learning Series. Mathematics Education Research Group, Graduate School of Education, University of Pennsylvania, Philadelphia, Pennsylvania. (c)

Scandura, J. M. Deterministic theorizing in structural learning. Journal for Structural Learning, 1971, 3, 21-53. (a)

Scandura, J. M. A theory of mathematical knowledge: Can rules account for creative behavior? Journal for Research in Mathematics Education, 1971, 2, 183-196. (b)

Scandura, J. M. Mathematics and structural learning. New York: Gordon & Breach Science Publishers, 1972, in press.

Scandura, J. M., Gramick, J., & Durnin, J. The arithmetic skills project. Paper presented at the Annual Meeting of the American Educational Research Association, New York, February, 1971.

Shoemaker, D.M. & Osburn, H.G. Computerized item sampling for achievement testing: A description of a computer program implementing the universe defined test concept. *Educational and Psychological Measurement*, 1969, *29*, 165-172.

Suppes, P. The uses of computers in education. *Scientific American*, 1966, *215*, 207-220.

Suppes, P. Stimulus-response theory of finite automata. *Journal of Mathematical Psychology*, 1969, *6*, 327-355.

Winer, B.J. *Statistical principles in experimental design*. New York: McGraw-Hill, 1962.

Table 1

Equivalence Classes	Item Forms	Stimulus Instances from Classes in the Intersection
1.	Basic facts; minuend < 10	9 -7
	Subtract 0	4 -0
	Answer 0	8 -8
2.	Basic facts; minuend > 10	13 -6
	Basic fact; minuend = 10	10 -3
3.	No borrow; no 0 in answer or problem	45 -23
	" ; x-0 fact in problem	36 -10
	" ; 0-0 fact in problem	802 -301
	" ; x-x fact in problem	342 -321
	" ; small unequal lengths	268 -24
	" ; large unequal lengths	28759643 -427102
4.	No description	153 -92
5.	Simple borrow	35 -17
	Simple borrow; one digit answer	68 -59
	Repeated borrow	811 -623

Table 1 cont.

Equivalence Classes	Item Forms	Stimulus Instances from Classes in the Intersection
	Simple borrow; 1 digit subtrahend	38 -9
6.	Repeated borrow	1563 -875
7.	Simple borrow	352 -216
	Simple borrow; 1 digit answer	723 -716
	Simple borrow; 1 digit subtrahend	5673 -8
	Simple borrow; medium	68423 -51712
	Borrow; 1 digit from large number	9463217 -9
	Borrow; medium; unequal lengths	85463 -392
	Repeated borrows	4223 -1332
	Separated borrows	98542 -4617
	Borrow; medium; subtrahend 1 digit short	74918 -4622
8.	Borrow; medium; subtrahend 1 digit short	15362 -8071
	Repeated borrows	12459 -6990
	Separated borrows	186421 -98371
9.	Borrow across 0	603 -578
	Borrow across two (or more) 0's	5002 -2138

Table 1 cont.

Equivalence Classes	Item Forms	Stimulus Instances from Classes in the Intersection
10.	Borrow across 0	4029 -3642
	Borrow across two (or more) 0's	70035 -41362
11.	Borrow across 0	1500 -877
	Borrow across two (or more) 0's	14003 -9678
12.	Borrow across 0	11029 -8437
	Borrow across two (or more) 0's	160018 -76325

Table 2

Numbers of Items, Percent Correct Predictions, and Correlations between Corresponding Items

Tests	Number of Items	Number of Test A(A') instances on which Ss were successful	Percent correct predictions	Number of Test A(A') instances on which Ss were not successful	Percent correct predictions	Total number of Test A(A') instances	Percent correct predictions	Correlation between corresponding A(A') & B(B') test instances
Intersection	37	699	91%	226	55%	925	82%	.49
Item Forms	21	444	89%	81	51%	525	83%	.39
Equivalence Classes	12	225	85% a	75	71% a	300	82%	.53 a
Restricted Intersection	18	420	94%	30	37%	450	91%	.29
Hierarchical Forms	6	133	94%	17	41%	150	89%	.36
Restricted Item Forms	13	302	95%	23	22%	325	90%	.19
Restricted Equivalence Classes	7	165	93%	10	30%	175	89%	.19

a: Differences significant at the .05 level

Table 3

Components of Variance in Item Scores

Source	TESTS: Intersection	Item Forms	Equivalence Classes
Subjects			
MS	1.144	.420	.539
σ^2	.014	.008	.019
%	8	6	9
Classes			
MS	1.887	1.443	2.525
σ^2	.033	.020	.045
%	19	15	22
Items (within classes)			
MS	.157	.106	.182
σ^2	.003	.001	.004
%	2	1	2
Subjects by Classes			
MS	.163	.124	.194
σ^2	.037	.020	.055
%	21	15	27
Subjects by Items (within classes)			
MS	.089	.084	.083
σ^2	.089	.084	.083
%	51	63	40

Table 4

Coefficients of Generalizability α' and α'_s
for each Test

Tests	α'	α'_s
Intersection	.85	.87
Item Forms	.62	.66
Equivalence Classes	.71	.74
Restricted Intersection	.39	.46
Hierarchical Forms	.15	.14
Restricted Item Forms	.29	.25
Restricted Equivalence Classes	.30	.21

Note: $\alpha' = \dfrac{\text{MS between people} - \text{MS people x tests}}{\text{MS between people} + \text{MS people x tests}}$

$$\alpha'_s = \frac{S_t^2 - (2 \sum_c \sum_{i_c} S_{i_c}^2 - \sum_c S_c^2}{S_t^2 + (2 \sum_c \sum_{i_c} S_{i_c}^2 - \sum_c S_c^2)}$$

where S_t^2 is test variance, $S_{i_c}^2$ is item variance within a class and S_c^2 is class variance.

Table 5

Pass(+)-Fail(-) Relationship Between Superordinate Problems and Relatively Subordinate Problems

Hierarchies	Number of Cases for each Relationship Between Superordinate Problems and Relatively Subordinate Problems				Test for Verifying Hierarchies	
	1. Super.+ Sub.+	2. Super.- Sub.-	3. Super.+ Sub.-	4. Super.- Sub.+	N 1+2+3	Proportion $\frac{1+2}{1+2+3}$
Item Forms	219	79	103	49	401	.74
Equivalence Classes	109	53	35	53	197	.82
Hierarchical Forms	64	3	13	20	80	.84
Restricted Item Forms	181	13	34	22	228	.85
Restricted Equivalence Classes	92	2	13	18	107	.88

Figure Captions

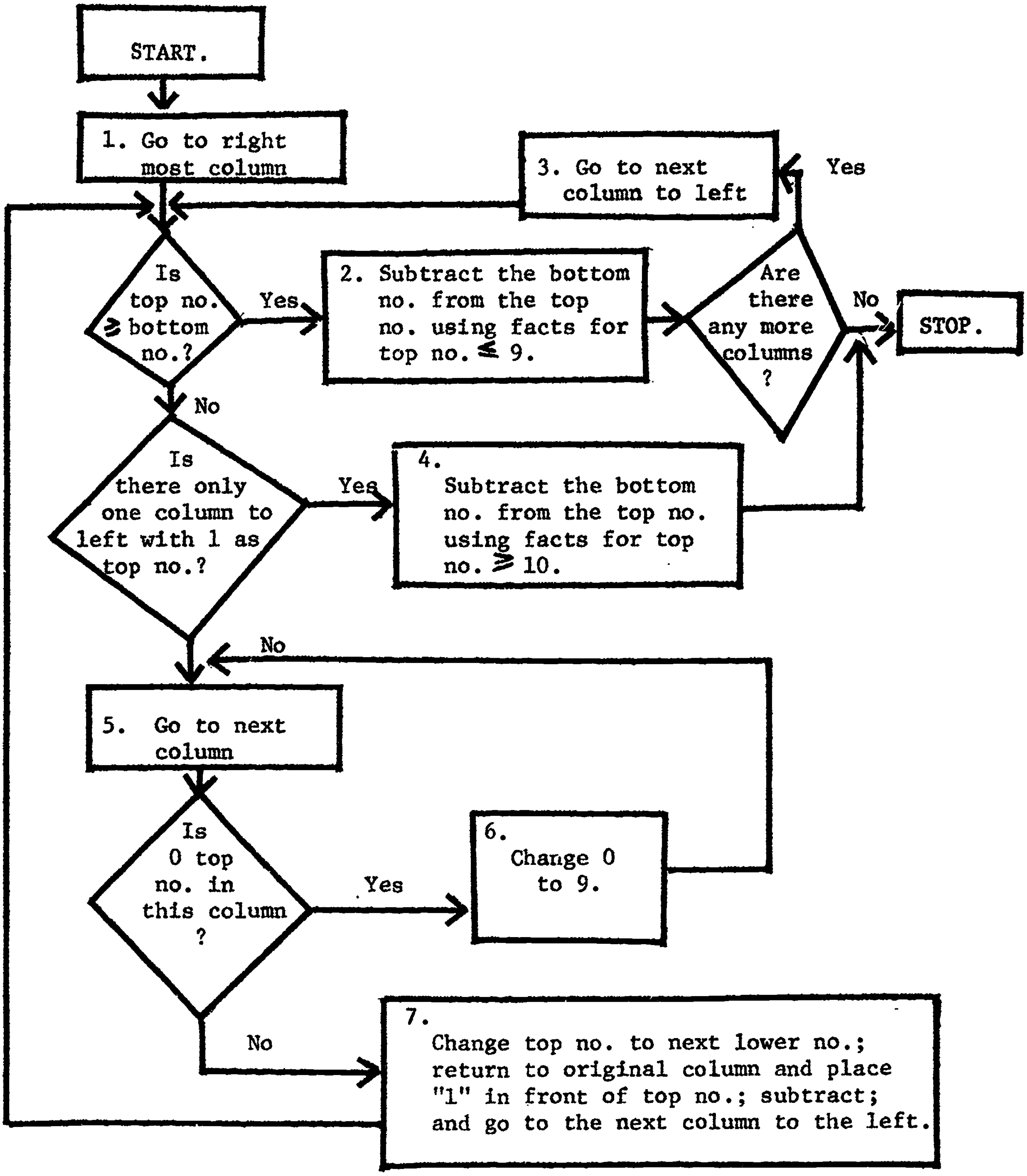
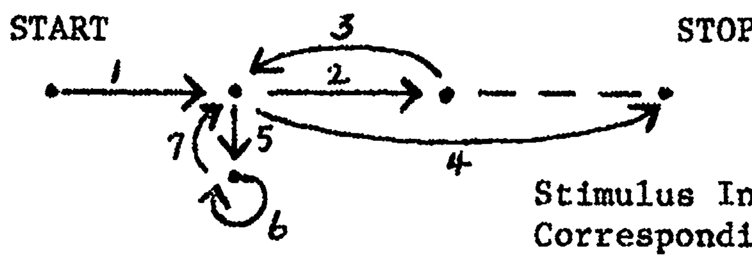
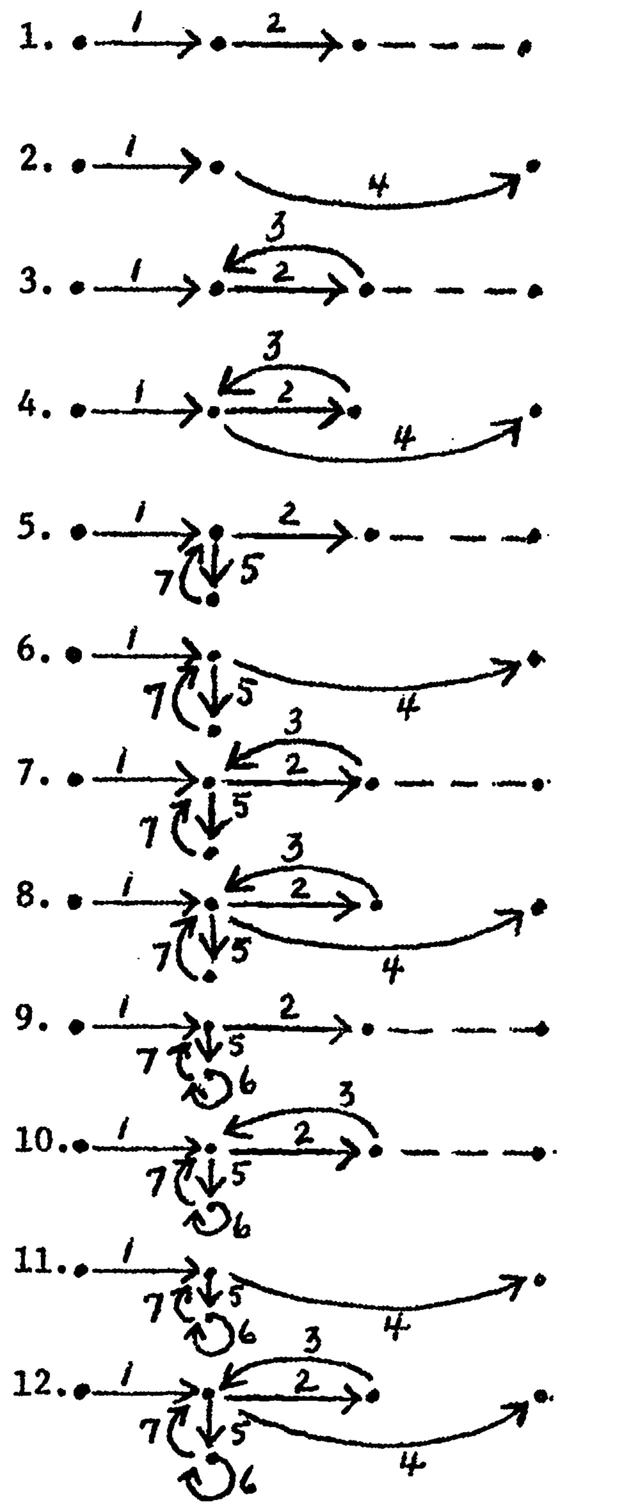


Figure 1: Subtraction Algorithm

Directed Graph



Paths



Path	Stimulus Instances from Corresponding Equivalence Classes
1.	7 -3
2.	13 -6
3.	258 -13
4.	153 -92
5.	54 -27
6.	1563 -875
7.	268 -97
8.	1663 -824
9.	603 -578
10.	4029 -3642
11.	1300 -423
12.	16059 -8797

Figure 2: Directed graph and paths of subtraction algorithm

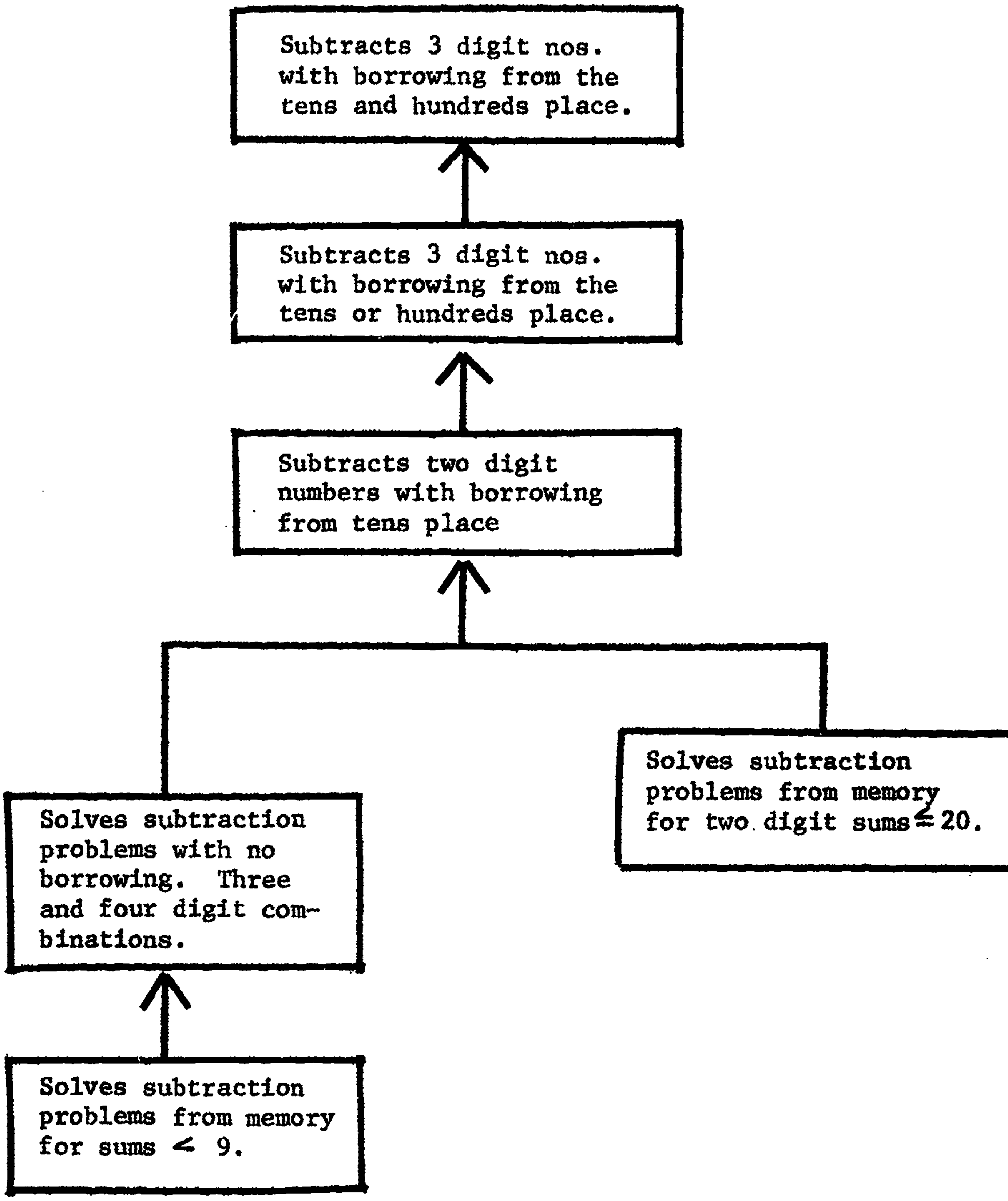


Figure 3: Hierarchical Forms adapted from Ferguson (1969)

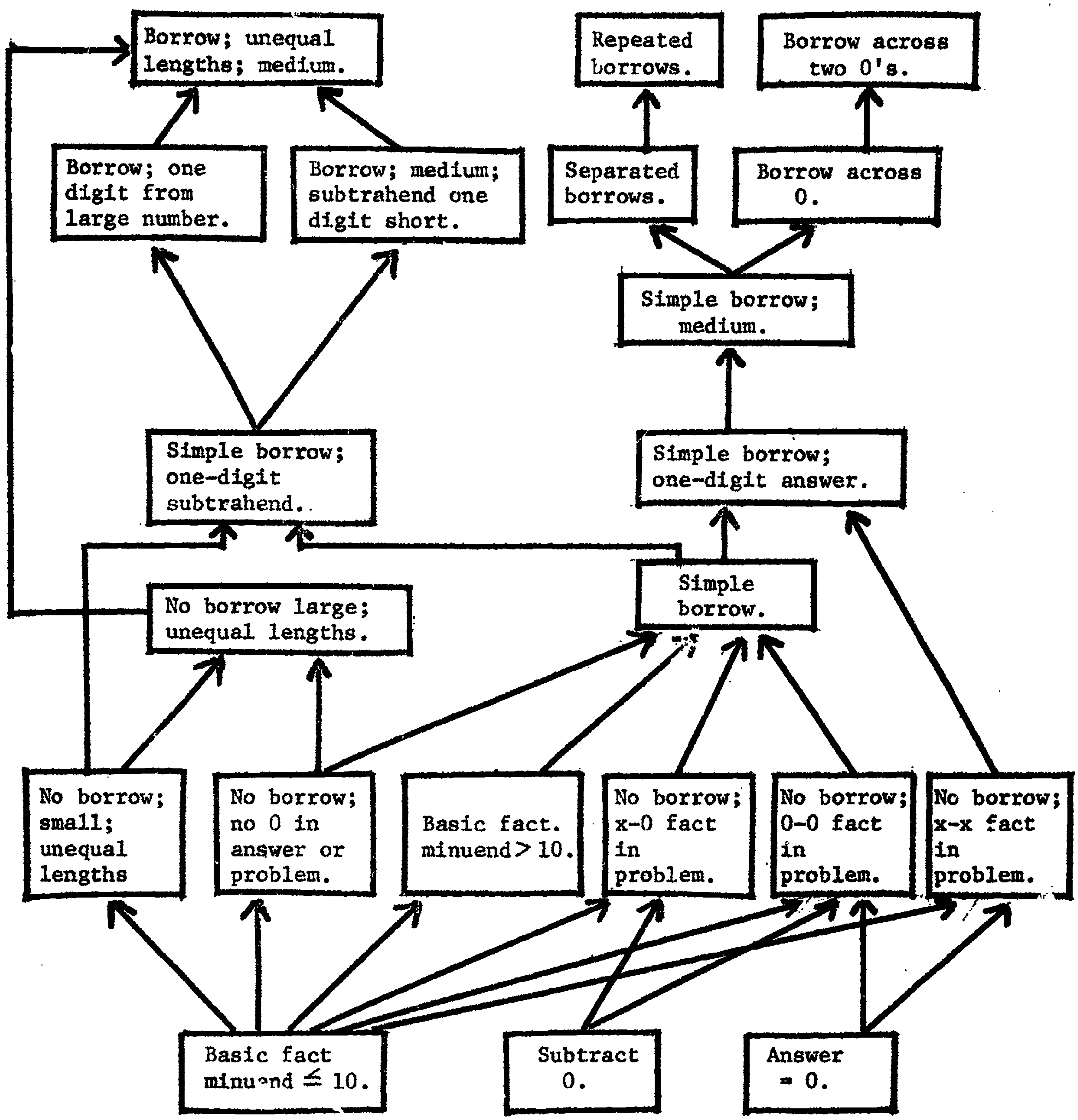


Figure 4: Hypothesized hierarchy for subtraction item forms adapted from Hively, Patterson, & Page (1968)

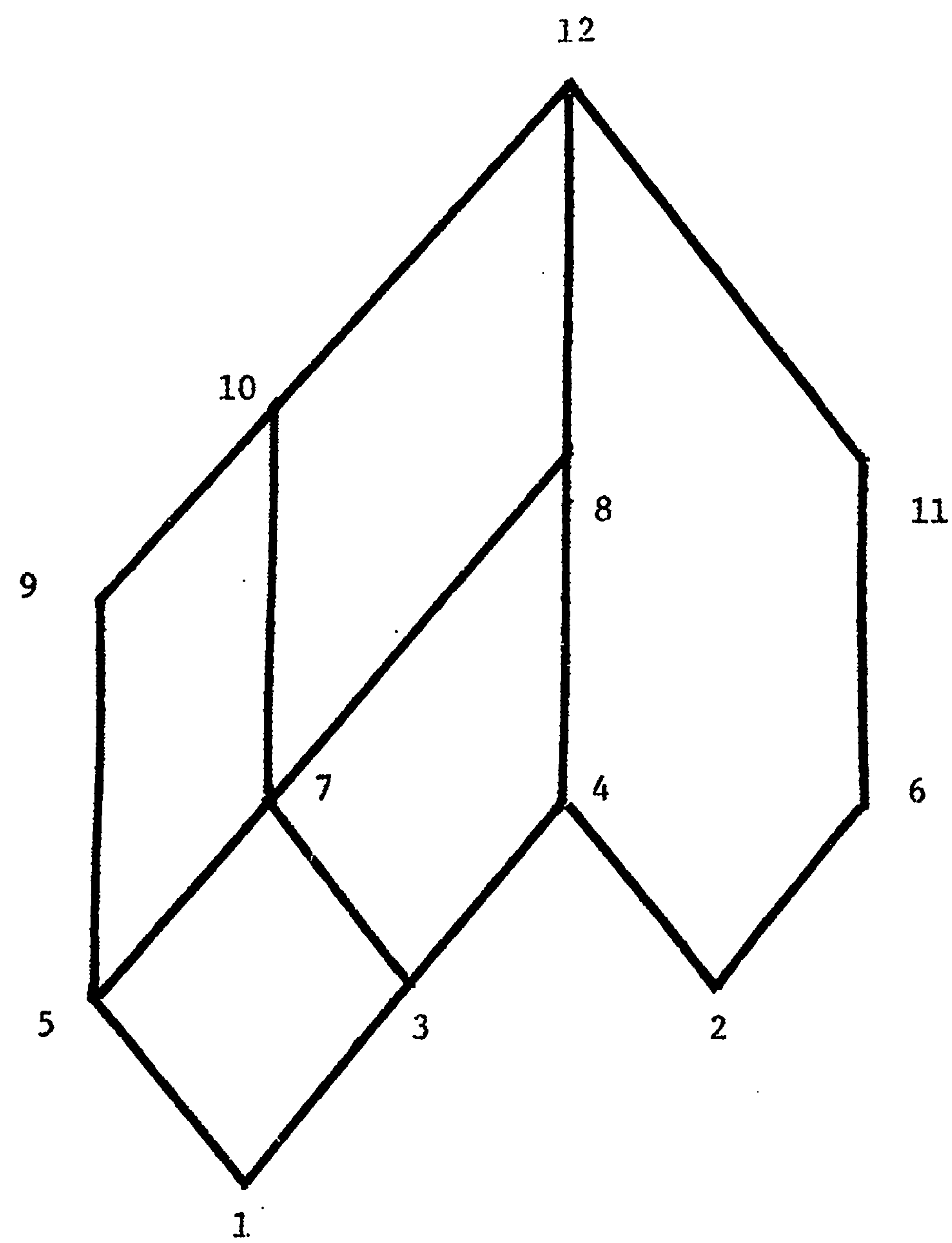


Figure 5: Hierarchy of Paths based on Subtraction Algorithm