R E P O R T   R E S U M E S

ED 014 894   24   EM 004 089

THE RULEG SYSTEM FOR THE CONSTRUCTION OF PROGRAMMED VERBAL LEARNING SEQUENCES.

BY- EVANS, JAMES L.   AND OTHERS
PITTSBURGH UNIV., PA.

REPORT NUMBER CRP-691-3   PUB DATE   AUG 60
EDRS PRICE MF-$0.25 HC-$1.04   24P.

DESCRIPTORS- *PROGRAMED INSTRUCTION, *PROGRAM CONTENT, *PROGRAM DEVELOPMENT, *SYSTEMS APPROACH, *PROGRAM DESIGN

THIS PAPER DESCRIBES THE RULEG PROGRAM SYSTEM, WHICH IS BASED ON THE PREMISE THAT THE VERBAL STATEMENTS OF A PROGRAM CAN BE CLASSIFIED INTO "RU'S" (RULES) AND "EG'S" (EXAMPLES). ALL RU'S ARE STATEMENTS OF SOME GENERALITY, FROM WHICH SUBSTITUTION INSTANCES, OR EG'S, CAN BE OBTAINED. 12 STEPS FOR THE CONSTRUCTION OF RULEG PROGRAMS ARE LISTED--SPECIFY THE CRITERION BEHAVIOR, WRITE DOWN MANY RU'S, COLLECT STIMULUS SUPPORT, ORDER THE RU'S, MAKE A RU MATRIX, CONSTRUCT EG'S, NUMBER THE MATRIX CELLS, ASSEMBLE RU'S AND EG'S INTO FRAMES, AND THE FRAMES INTO A PROGRAM, PRE-TEST AND THEN REVISE THE PROGRAM UNTIL RELIABLE CRITERION BEHAVIOR IS ACHIEVED. (LH)

[redacted]

[redacted]

[redacted]

EM004089

The RULEG System for the Construction of Programmed Verbal Learning Sequences

James L. Evans

Robert Glaser

Lloyd E. Homme

Department of Psychology
University of Pittsburgh

The RULEG System for the Construction of Programmed Verbal Learning Sequences

James L. Evans

Robert Glaser

Lloyd E. Homme

Department of Psychology
University of Pittsburgh
Pittsburgh 13, Pennsylvania

August - 1960

A Report Prepared Under Cooperative Research Project Number 691(9417)

The United States Office of Education
Department of Health, Education, and Welfare

PREFACE

It is now generally realized that the success of a teaching machine depends greatly upon the material used in it and that the development of principles and techniques for programming this material is an important area for investigation. At the present time work in the field. (Teaching Machines and Programmed Learning by A. A. Lumsdaine and R. Glaser, 1960, National Education Association) has suggested certain rudimentary techniques.[1] The eventual goal, however, is the development of a technology of programming based upon the science of learning and upon the results of detailed experimental tryout. The present report describes an approach toward a systematic procedure for the construction of programmed learning sequences for verbal materials. Experimental study with a program prepared according to the system described in this report is presented in a separate project publication (An Investigation of "Teaching Machine" Variables Using Learning Programs in Symbolic Logic, James L. Evans, In Press).

---

[1] A comprehensive review of the present notions underlying programming is also presented in a separate project report. (Principles and Problems in the Preparation of Programmed Learning Sequences, Robert Glaser, In Press).

Introduction to the RULEG System[2]

The RULEG system is based on the premise that the verbal subject material which appears in a program can be classified into two classes of statements which we will call Ru's (for "rules") and EG's (for "examples"). Learning to identify RU's and EG's is itself a problem in concept formation. Therefore, we will deal with it as with any other generalization-and-discrimination problem. That is, we will attempt to give a range of both RU's and EG's together, all properly labeled, so that we can learn to distinguish between the two concepts, although we may find at times that the two concepts are not mutually exclusive. This means that the concept of the RU will, on occasion, shade into the concept of the EG. This "relativity" of RU's and EG's in which the same verbal or symbolic statement may sometimes be both a RU and an EG has so far not caused much difficulty. On the contrary, it has proved useful in the construction of programs. Instances of statements which can be both a RU and an EG will be considered shortly.

We have not yet discovered a completely satisfactory definition of a Ru, nor are we convinced at this stage that we need one. Our strategy will be to exemplify as widely as possible statements which we have classified as RU's. A Ru may be a large number of things. It may be a definition, operational or otherwise. It may be a mathematical formula. It may be an empirical law. It may be a principle, axiom, postulate, or hypothesis from any area of knowledge. But the invariant feature of all RU's is that they are all statements of some <u>generality</u>, from which substitution-instances can be obtained. These substitution-instances are called EG's. An EG may be a large number of things. It may be a description of a physical event. It may be a theorem or deduction of any sort. It may be a statement of a relationship obtaining between specific objects, whether the objects are physical or conceptual. But the invariant feature of all EG's is that they are all statements of some <u>specificity</u>, derived from more generalized RU's.

---

[2]Since any set of principles of programming is itself verbal material, such principles could no doubt be best presented in program form. But to escape getting caught in an infinite regress involving programming programs on how to program, this paper will appear in conventional form.

The clearest instance of RU's and their corresponding EG's come from mathematics. For example, the algebraic " a + b = b ± a", is a RU, which summarizes compactly an (infinite) number of substitution-instances, or EG's, such as "7 + 2 = 2+7" and "3.4 + 8.6 = 8.6 + 3.4". Again, the statement "Unsupported objects will fall toward the earth" is a RU, while "If I release my pencil it will fall" would be an EG, as would "If I release my book, it will fall". From optics we have the RU that "The angle of incidence is equal to the angle of reflection". To exemplify, we create EG's such as "If the angle of incidence is 30°, the angle of reflection is equal to 30°". From psychology we have "Withold reinforcement from an organism and extinction occurs" for the RU, and "If I stop giving the rat pellets, he will extinguish the appropriate response" as an EG for the RU. Of course, all of these RU's (including the mathematical ones which are more precise) would need qualification to become acceptable, but the point should be clear that RU's involve some generality, while EG's involve specificity, albeit a relative specificity. That is, "3 + 2 = 2 + 3" is in turn a RU for such EG's as "3 stones + 2 stones = 2 stones + 3 stones" and "3 mammoths + 2 mammoths = 2 mammoths + 3 mammoths". Indeed, the early man who induced the "3 + 2 = 2 + 3" RU from these EG's for the first time was probably staggered at the profundity of his own insight. Several eons passed before a generalization was made to the "a + b = b + a" RU, making a new EG out of an old RU. And it remained for a modern group theorist to produce that ultimate (?) RU "aob = boa" in which nether the objects nor the operator (o) are specified, and in which "a + b = b + a" becomes a mere EG. Other examples of RU's and EG's could be drawn from the physical sciences, to demonstrate the relativity and inter-dependence of RU's and EG's. With this RU-EG concept in mind, let us now see how it is related to the preparation of programmed learning sequences and proceed to the steps in the construction of a program.

STEP 1. Specify the criterion behavior. At this step the programmer must attempt to outline, as precisely as possible, both the responses he wants from the student at the end of the program, and the stimuli or cures in the presence of which the student will be expected to make these responses. This is the time when questions such as the following must be asked and answered. If the student is studying statistics, will he have to produce the formula for the standard deviation on his own, or will he always have a book available? Is he being prepared to write a short essay comparing two statistical tests or will he be called upon to take a multiple-choice test at the end of the program? In psychology, will we be satisfied if he can identify and label examples of regression or spontaneous recovery, or will he be asked to recommend ways of preventing the occurrence of these phenomena? In language, do we want to build in a complete reading, writing, and speaking knowledge? Or would we be satisfied if the student can merely read the technical literature in his chosen field with the aid of a dictionary? In mathematics, do we want the subject to be able to solve problems, or to prove theorems, or both? With what stimulus-support may the student provide himself while we are assessing his criterion behavior? Can he use a book? Another student? The instructor? His notes? A table of integrals? The construction and form of the program will differ radically as a function of the criterion behavior chosen. Now is the time to be both realistic and specific in stating what the objectives of the programmed course are.

STEP 2. Write down all the RU's you can think of. Do this first in the absence of any support from books, notes, charts, or advice from experts. These will all be available later. At this point you want to come up with as much on your own as you can. If you look at a familiar text first, you may be off on a response chain which will bypass a number of useful rules which you otherwise could have produced. In other words, you may fall too strongly under the control of some previously learned varbal habits in that area. If you are a subject-matter expert, this step may produce all the RU's which you will need for your program; if you are trying to program a new course as you learn it yourself, it may produce only a few RU's. In any event, you should try to get out as much verbal behavior on the topic as you can before you begin to utilize external stimulus supports such as textbooks. Write down each RU on a separate index card.

STEP 3. Collect all the stimulus support you can in the form of texts, notes, advice, and so on. Examine these systematically for the RU's which you wish to build into your students' repertory. Write these RU's on index cards also. If you find EG's only, induce the RU yourself and write it down. (At this step as well as in many of the preceding and following steps, do not be overly critical of your efforts. The ultimate critic is the student; he will criticize your program by his ability or inability to perform as desired. Do not try to guess whether your program is good or bad. Too detailed criticism at this time will slow programmers down and will take longer to get in the hands of a student. Once the program is out, you can get feedback on it and revise it.)

STEP 4. Make some preliminary ordering of RU's. RU's have to be presented in some order, but the conventional order in which they appear in texts is not necessarily the best order. You have put RU's on separate cards to make it easier for you to experiment with orders of presentation. A number of ordering relationships are useful here, such as complexity (introduce simpler RU's first), chronology (ordering RU's in time, as in perhaps a history program), spatiality (ordering RU's in space, as in perhaps a geography program), and, in particular, dependence on other RU's. To illustrate the last relationship you would introduce RU's about the mean in a statistics program before you introduce RU's about the variance, since the definition of the variance involves the definition of the mean. Watch for omitted RU's; don't force your student to deal with unfamiliar terms, or old terms which may need a bit of refreshing. You can anticipate this by a careful consideration of the RU's and the behaviors involved.

STEP 5. Make a RU matrix. List all your RU's vertically like this:

RU 1

RU 2

RU 3

Then, list the same RU's horizontally like this:

	RU 1	RU 2	RU 3	. . .
RU 1				
RU 2				
RU 3				
⋮				

The use of a matrix is based on the fact that the hallmark of the expert in a subject-matter is his ability to interrelate the concepts in his field. The RU matrix permits a systematic method for examining the RU's of a subject singly, in ordered pairs, triplets, and so on. RU's can be checked for similarities, differences, possible confusion, or any of a host of possible intra-verbal connections. The matrix is also not without heuristic value, in that it may prompt comparisons or relationships not previously considered ("How are the laws of thermodynamics related to the laws of optics?" "How is what we know about bread mold related to bacterial growth?").

The upper-left corner of the matrix is reserved for operators for inter-relating the RU's. A very general operator is the relator operator. Consider a matrix like this:

RELATE	RU 1	RU 2	RU 3
RU 1	1	2	3
RU 2	4	5	6
RU 3	7	8	9

As we consider cell #4, for example, we would ask "How is RU 2 related to RU 1?" This might require a different handling than cell #2, which asks, "How is RU 1 related to RU 2?" If the order of relationship is of no consequence, symmetrically placed cells, such as 2 and 4, 3 and 7, 6 and 8 are redundant. If ordering does matter, the complete matrix permits examination of every ordered pair. Such ordered pairs may be added to the axes as new operants (row and column entries) to permit expansion of the matrix to ordered triplets, quadruplets, etc. For example, we could first relate pressure to volume in the "perfect" gas equation, then put this relationship on the axes of the matrix, and have a new cell relating pressure and volume to temperature.

The major diagonal relates each rule to itself, and we have found it useful to reserve this diagonal as definition cells. We "relate a RU to itself" by defining the RU in terms of some previous behavior which we can safely assume exists at sufficient strength in the student to make the definitions meaningful.

If we use another useful operator, which we might term the discriminator operator, we might get a matrix like this:

DISCRIMINATOR	RU 1	RU 2
RU 1	1	2
RU 2	3	4

Cell #2 would permit us to ask "How is RU 1 different from RU 2?" This may set the stage for discrimination training if the RU's have formal or thematic similarities which may later confuse the student.

STEP 6. Use the example operator in the RU matrix to construct examples for the program. This is a critical feature of the RULEG system. The EG's which are constructed must meet a number of criteria. These are:

1. An adequate number of EG's must be generated. It is mainly through EG's that the student will interact with the subject-matter. EG's must be generated not only to exemplify a RU initially, but to provide later practice and review for that RU.

2. The full spectrum of EG's for a given RU must be considered. This means considering special cases, limiting cases, trivial cases, examples with inadequate information, and examples with redundant information. As a good rule of thumb, the first EG for a given RU should be the simplest possible non-trivial example. For instance, for a student's first example of the computation of a variance, use an N of 2, with simple integer scores which produce a mean which is

also an integer. For example, use 3 and 5 which give a mean of 4. Two scores are adequate to illustrate the basic operations involved. An EG with an N of 1 is an important special case, but it can be saved for later. Also, EG's in which all the scores are the same, such as 4, 4, 4, and 4, should be reserved for later. Decimal or fractional scores (e.g., 7.319 or 8 7/16), if used on the initial EG, would divide the student's attention between learning the operation and fussing with the arithmetic at a crucial stage in his learning. Leading off with a simple non-trivial EG for a RU is the sine qua non of good programming. Any temptation to be complex or devious should be resisted at this time; there will be opportunity for this when the student's behavior has been adequately strengthened. Complex "test items" come later.

3. As pointed out before, to insure adequate generalization of a RU the attached EG's must be as diverse as possible and still exemplify that RU. Also, to insure adequate discrimination between RU's, EG's must be selected which resemble each other as nearly as possible while still exemplifying their respective RU's.

STEP 7. Number the cells of the original RU matrix to indicate the order in which the RU's are to be presented and exemplified. It is often useful to start off "down the diagonal", utilizing the major diagonal (the intersection of each RU with itself) as a definition frame. After that, you can decide what RU's you want to compare and contrast, what pairs of RU's you wish to omit (be cautious of labeling RU's as being "unrelated"; like a careful driver, take all "intersections" seriously). Number in sequence all cells with which you wish to deal. This numbered RU matrix will be a useful step in the assembly of the final program.

STEP 8. Begin assembling the RU's and EG's into frames. A frame consists of the stimulus material available at any given moment which demands one or more responses on the part of the student. The student makes his indicated response, and then receives immediate feedback as to the correctness or incorrectness of his response. Frames are constructed by judicious selection and combination of RU's and EG's from the population of RU's and EG's which have been constructed. To provide for responses on the part of the student, words, phrases, numbers, symbols and so on are omitted from certain RU's and certain EG's. Such omissions are indicated with a rectangular box, like this: ☐. For instance, we could give an incomplete RU in this manner: The angle of incidence is equal to the angle of ☐, where "reflection" is to be provided by the student. As an example of an incomplete EG involving numbers, we might have:

$$8 + 3 = 3 + \square$$

with '8' being the appropriate response. The choice of the response to be made by the student is obviously an important matter. It often is easier for stylistic reasons to omit certain words from a frame than others; however, it is necessary to consider just what the response required by the student can accomplish toward producing the criterion behavior. Does the response review or strengthen the responses elicited in part frames? Does it help set him up for the next step? Does it "test" what he should have learned previously?

To designate the incomplete RU's and EG's, we add the symbol "~" (read tilde) over the symbols for rules and examples, like this: $\widetilde{RU}$ (RU tilde) and $\widetilde{EG}$ (EG tilde). For terminal situations in which criterion behavior is being called for in the presence of minimal stimulus support, use two tildes over the RU and EG symbols, like this: $\overset{\approx}{RU}$, $\overset{\approx}{EG}$. An example of a $\overset{\approx}{RU}$ would be:

> The machine formula for the Pearson-product-moment correlation coefficient is ☐.

No other hints or aids would be available to the student. An example of an $\overset{\approx}{EG}$ would be:

> The variance of the following scores: 3, 7, 4, 2, 9 is ☐.

Again, no other RU's or EG's which might serve as prompts must be present, or we would have an $\widetilde{EG}$, not an $\overset{\approx}{EG}$.

We will now present a list of frame-types which we have found to be most useful in program construction, as well as a rationale for the use of each frame-type.

1. <u>RU + EG + $\widetilde{EG}$</u>. We have found this frame-type to be the method par excellence for the economical introduction of a new RU. The RU + EG + $\widetilde{EG}$ can have the student working an example of a brand new rule on the very first frame in which he is exposed to it. By giving him an explicitly stated RU, and one (or more) carefully chosen EG's before calling for a response by means of an $\widetilde{EG}$, a powerful prompt is set up which makes an error in responding most unlikely. To illustrate a RU + EG + $\widetilde{EG}$ frame:

> To multiply by 100, move the decimal point two places to the right. (RU) For example, 100 x 2.843 = 284.3. (EG) Also, 100 x 7.374 = ________. ($\widetilde{EG}$)

This is a critical phase in the development of a program. It seems best not to introduce more than one RU at a time, or to introduce complex RU's too early, or to ignore consideration of the "simplest possible non-trivial example". (The first example in many texts in an effort to be realistic is often much too complicated.) Work the student up to complicated and realistic examples. If 20 redundant frames are needed to present a RU adequately, then use 20 redundant frames, and don't worry about the extra space you use. If someone says, "Can't you omit that example? You have used one very much like it before," let the student's performance decide whether you will delete it or not. The primary consideration is that prerequisite behavior must be adequately strengthened before the student can proceed. (The usual introductory text or lecture may not effectively provide for performing this function.)

We have found then that RU + EG + $\widetilde{EG}$ (in that order) is the most effective method of presenting an initial frame for a new RU. However, other combinations are certainly possible. For example, why not RU + $\widetilde{EG}$? We have found that at least one completely worked-out example is necessary to permit the working of the first EG with ease and efficiency. Why not EG + $\widetilde{EG}$? This would be an <u>analogy</u> frame. The chief characteristic of an analogy is that one must <u>induce</u> the RU from the first EG, and then apply it to the $\widetilde{EG}$. Rather than run the risk of having the student induce an incorrect RU, it seems preferable to state the RU for him explicitly. Incidentally, this same philosophy lead us to reject in general the inductive presentation which we might symbolize as: $EG_1 + EG_2 \ldots EG_n + \widetilde{RU}$. Here a large number of EG's are given and the student is asked to state the RU involved. Such a Socratic technique is inefficient: Humans, with their verbal and symbolic behavior, can be given a rule to follow and it is not necessary to make them guess at it. (And we are trying to bring some specific verbal and symbolic behavior under the control of certain verbal and symbolic stimuli.) If we want to shape up the student's inductive behavior, we would construct an induction program for that purpose.

After the student can recognize and apply a RU with proficiency, then $EG + \widetilde{EG}$ frames are acceptable. But until that time, we feel that it is often hazardous and slow to "sneak up" on a RU through induction and incidental learning. We would prefer that the student adopt the expert programmer's carefully chosen statement of a RU rather than have the student use his own halting induction-derived statement. Examples in text books are typically too few in number or too restricted to indicate the full range of the rule in question. This makes it possible for a student to induce what is essentially an incorrect rule, but one which happens to fit all the examples present. This constitutes another possible source of danger of the induction process.

After initial introduction to a RU through the $RU + EG + \widetilde{EG}$ formula[3], a number of variations have proved useful for the subsequent frames. The following frame-types represent only a small number of the possible RU-EG combinations, but they have proved particularly useful in our work in frame construction.

2. $RU + \widetilde{RU}$. RU's typically contain "technical vocabulary" words and terms, which we call TV words. These are words which may be quite new to a student, e.g., perigee in astronautics, or they are

[3] It is tempting here to write: $RU + EG + EG + \widetilde{EG}$, to indicate that one or more complete examples can be used in the initial frame. Depending on the RU involved, it is often a good idea to provide multiple complete EG's so the student will begin to generalize on the RU and not have the RU attached to only a small number of EG's

familiar words with new meanings to be attached, e.g., population in statistics. Students often are slow in adding such words to their active vocabulary. If a TV word is defined in one frame, and then called for after subsequent frames, it has been our experience that the student will often fail to come up with the word, or will come up with the word only with difficulty. It has proved good technique to force attention to the TV terms by calling for them explicitly soon after they are introduced. To illustrate:

> The Commutation Law permits the interchance of members connected by a "+" sign. (RU) So if you found a case where two numbers had been switched about a "+" sign you could be sure that the ______ had been used. ($\widetilde{RU}$)

3. $\underline{RU + \widetilde{EG}}$. In late frames stimulus support can start to be withdrawn by giving only the RU before giving the EG. Likewise the previously mentioned "analogy" frame (EG + $\widetilde{EG}$) represented a weaker prompt for the EG than does RU + EG + $\widetilde{EG}$.

4. $\underline{EG + \widetilde{RU}}$. An induction frame which can be used more safely when the student has the RU at some strength.

5. $\underline{\widetilde{RU}_1 + \widetilde{RU}_2}$. Such frames can be used to compare and contrast two different RU's as part of the discrimination training involving the RU's. This can obviously be extended to $\widetilde{RU}_1 + \widetilde{RU}_2 + \widetilde{RU}_3$ and so on.*

6. $\underline{\widetilde{EG}_1 + \widetilde{EG}_2}$. For comparing and contrasting examples of two different RU's, or for demonstrating how two RU's can be applied in turn. Again the extension to $\widetilde{EG}_1 + \widetilde{EG}_2 + \widetilde{EG}_3$, etc., is obvious.[4]

7. $\underline{\approx EG}$. The terminal frame of a series in which criterion behavior is performed in the presence of minimal cues. For example, we might ask the student to "Compute an analysis of variance on the following three groups of scores..." or "Describe the acetylcholine cycle."

8. $\underline{\approx RU}$. Although the EG frame is more basic, in the sense that to respond appropriately to an EG, the student must have command of

---

*4 Note that these frame-types would be produced by the matrix technique in the programmer's effort to deal systematically with the important intra-verbal connections which make up the subject-matter.

the RU or RU's involved, on occasion we may wish to call for a statement of a specific RU. To exemplify: "What is the second law of thermodynamics?"

9. $\overline{EG}$. There is another important concept in the RU-EG framework. This is the concept of the "false example" which we symbolize like this: $\overline{EG}$ (read "egg bar"). The rationale for the $\overline{EG}$ is this: Opinion on the role of mistakes and errors in the course of programmed learning is divided. Some hold with Skinner that mistakes are to be avoided and that providing students with opportunities to make errors with a multiple-choice-type format is hazardous. Others, like Pressey and Crowder, feel that making mistakes and dealing with the consequences is a part of the way life is, and their devices, multiple-choice machine and scrambled textbook, reflect this approach. Others have even hypothesized an "optimal error rate" for programmed learning. Our philosophy is this: In addition to the rich intra-verbal associations which the professionally competant person in some fields of learning has, he possesses an additional ability. This consists of his facility in detecting and correcting errors, even though he might never make those errors himself. Such "error-detecting" behavior can be taught like any other behavior, i. e., by giving the student a chance to practice it and get reinforced for performing appropriately. Every field of knowledge has its own peculiar booby-traps and sources of confusion which become painfully well-known to the teacher. Rather than have the student fall into these errors, and rely on uncertain punishment effects to correct his behavior, our technique has been to apprise him (the $\overline{EG}$), and indicate that he is to correct it. By considering error-detecting behavior as a part of the repertory of the expert we are trying to emulate, we can usually get students over rought spots in programs without actually having them emit the incorrect behavior.

Examination of the protocol following a programmed learning session will help point up the sources of error, and subsequent programs can incorporate $\overline{EG}$'s at the appropriate spot. An example of an $\overline{EG}$:

In his homework a student had the following equation: $3(5 + 2) = (3 + 5)(3 + 2)$. But you know this is a mistake. The correct equation is: $3(5 + 2) =$ [   ]
Ans. $(3 \times 5) + (3 \times 2)$

Another use of the $\overline{EG}$ is in getting the student to discriminate where and where not to use TV words. For example:

Names of people and places are termed proper nouns. For example, "Chicago" is the name of a place, so it is a proper noun. "Tree" is not the name of a person or place so it is not a [   ].

To summarize the use of the $\overline{EG}$: When inspection of the RU matrix or program protocols indicate error-likely spots, present the error <u>labeled as such</u> and permit the student to deal with the error while still receiving positive reinforcement (confirming feedback).

As an illustration of varkous frame types, the Appendix to this report presents a revision of Skinner's sample physics program in RULEG terms. This illustrative program is taken from an initial paper by Homme and Glaser (Problems in Programming, Lloyd E. Homme and Robert Glaser, University of Pittsburgh. A paper presented at the meetings of the American Psychological Association in Cincinnati, Ohio, September, 1959.) which involved a brief discussion of the RULEG notion.

STEP 9. <u>Using the numbered RU matrix as a prompt, assemble the frames of various types into a program</u>. Among the problems to be faced at this step are the following: How many frames should be used for each RU? How should these frames be distributed throughout the program? Our procedure is to lead off with a RU + EG + $\widetilde{EG}$ frame, and terminate when the student can deal effectively with $\overset{\approx}{EG}$ frames (and $\overset{\approx}{RU}$ frames). In between, the general principle is to withdraw stimulus support "gradually", with the use of such frame-types as RU + $\widetilde{EG}$, EG + $\widetilde{EG}$, and EG + $\widetilde{RU}$ as mentioned before. Actually, prior to the first administration of the program, the optimal number and distribution of frame-types within a sequence can only be estimated. This leads to the next step.

STEP 10. Give the program to students and "item-analyze" their responses to each frame. Only after a frame-by-frame analysis of students' recorded responses to the program can the effectiveness of a program be assessed. If the students are dealing effectively with EG's, then the program has done its job and further modifications can be in the direction of making it more efficient without interfering with its effectiveness. If the student is not producing the specified criterion behavior reliably, then steps must be taken to add more frames, redistribute the frames, clear up ambiguities, and so on. One of the chief strengths of the programming technique is its provision for analyzing and revising material on a detailed empirical basis for such analysis and revision, i. e., the responses of the learner.

STEP 11. Revise the program on the basis of students' responses and comments. Students should be encouraged to record their comments and criticisms on particular frames as they proceed through the program. The most difficult phase consists of convincing the student that when he makes an error the blame is on the program and not on himself. This often takes some doing, for most students typically enter the programming situation with a long and consistent history of having their faults and inadequacies pointed out to them. Near-errorless learning may come as something of a shock to them. But the student must be convinced that he is the most important participant in the effort for systematic improvement of the program. After initial administration of a program, a large proportion of the corrections will follow quite obviously from the nature of the errors made. Some will be a result of the simple typographical errors; others will be produced by easily-corrected ambiguities in the text. Failure to come up with a correct term or solution implies the necessity of additional prior strengthening of the concept involved.

To pay lip service to the idea of the constantly revised program is one thing; to do the actual dog-work is another. This is why it is risky to use any elaborate duplicating procedure early in the game. Frames written on separate index cards will probably get added to, removed, and edited. Programs prepared in ditto or mimeographed page form may be discarded and replaced by revised programs. But programs on long continuous tapes or nicely bound and printed are probably "locked in", and are quite likely to remain in that form. The point is that you are not writing a textbook; you are preparing a sequence which will teach a specified behavior, and you don't want to go to press until this program has demonstrated that it can do its job.

Precautions. In the process of program revision, there are a number of booby-traps. The first is the dager of over-prompting. If a student cannot come up with the word "entropy", the solution is not to give him "entrop_" and have him guess the letter "y". With adequate prompting, a jungle savage can get through a course in differential equations. It is easy to correct a program by adding prompts, re-administering it, and finding to one's delight that the errors drop practically to zero in the corrected frames. But the proof of the pudding is in the $\overset{\approx}{Eg}$'s; only if the student can produce criterion behavior with minimal cueing has a revision succeeded.

A second booby-trap consists of assuming that the student will respond appropriately to supplemental instructions, comments, and material which appears on the frame but is not involved in the actual production of the response to that frame. For example, do not present the quadratic formula and say "learn this". Do not print "read each frame carefully" on each item. Such techniques are probably not without effect, but they have a hard-to-control differential effect on students of varying compulsivity. A more realistic rule-of-thumb is: The student will attend only to those stimuli which are necessary to produce the correct response on any given frame. Exhortations to learn well and not to make errors are tenuous sources of control of student behaviors; behavior important enough to be strengthened at all should be dealt with by evocation and reinforcement.

The third booby-trap is somewhat allied to booby-trap number two, and is intimately related to the RULEG system itself. It stems from the fact that as we formalize a system of program writing, as has been done in RULEG, the responses of the student may fall under the control of the wrong set of stimuli. Many of us in our childhood had experience with a "teaching machine" in which the learner used a wire to connect a question, e.g., What is the capital of Arizona?, to possible answers. By joining the correct question and answer, an electrical circuit was made which rang a bell. The difficulty with such a device was that the learner soon learned the relative position between questions and answers, and was able to give the correct answer independently of the verbal material which was on the device. An analogous danger exists in formalizations such as RULEG. For example, if we underlined all TV words (technical vocabulary) the first time they were presented and always called for the TV word later on the same frame, the production of this word might soon fall under control of the underline, rather than under control of the textual material present.

On the other hand, formalization is not without its benefits. We have some observational evidence that some sort of a "learning how-to-learn" phenomenon can take place. That is, once a student gets into the swing of the RU + EG + $\widetilde{\text{EG}}$ formula, he begins to grasp new RU's very quickly because he can begin to anticipate how the presentation of the rule will be developed. If he finds that he is always required to state each RU at some later frame (as in a $\widetilde{\text{RU}}$), this may lead him to attend more carefully to the RU as it is first presented.

STEP 12. Repeat the administration and revision procedures until the program is producing criterion behavior reliably and efficiently. This last step concludes the presentation of the RULEG system of program construction. We trust that this early attempt to formalize and make explicit some of the rules of programming will be judged by the same criterion we use to ascertain the effectiveness of a program, namely, by results. Like a program, we hope that it will be edited, revised, supplemented, or discarded on the basis of the results it produces. In particular we would like to dispel any impression that we are irrevocably committed to a RULEG approach, or that we feel RULEG will solve all programming problems. The system contains a large number of gaps and guesses. But in our own work we have found RULEG to be an enormously helpful prompting system. Previous to a semi-formalization of RULEG, our programming behavior could best be described as sort of an artistic manipulation of the subject matter. Also, we found it almost impossible to be of any help to people who wished to learn to program, beyond such nebulous advice as "Proceed carefully in small steps." After we began to be more explicit (and dogmatic) in the rules for program construction, we noticed that our own frame production went up tremendously, and we found that for the first time we could make some sensible and helpful suggestions to beginning programmers. We have found the RULEG system flexible enough to be applied to such diverse topics as statistics, Hebrew, Hungarian, physics, symbolic logic, and investment banking; and communicable enough to be grasped and employed by bright undergraduates. Finally, our chief claim for the RULEG system is that it helps produce programs. Once a program has been constructed, then it can be evaluated and modified on the basis of the responses of the learner and his attainment of the criterion behavior.

APPENDIX

This appendix contains an illustrative revision of Skinner's sample physics program rewritten in Ruleg terms. The original program excerpt appeared in Science, 24 October 1958, Vol. 128, 3330, 969-977. The Ruleg revision appearing here was first described in a paper by Homme and Glaser which appears in the book Teaching Machines and Programmed Learning by A. A. Lumsdaine and R. Glaser, 1960, National Education Association.

Skinner's High School Physics Program Reconstructed
According to the Ruleg System

Class		Sentence to be completed	Word to be supplied
ru + $\widetilde{eg}$	1.	To "emit" light means to "send out" light. For example, the sun, a fluorescent tube, and a bonfire have in common that they all send out or ________ light.	emit
$\widetilde{eg}$	2.	A firefly and an electric light bulb are alike in that they both send out or ________ light.	emit
ru + $\widetilde{eg}$	3.	Any object which gives off light because it is hot is called an <u>incandescent</u> light source. Thus, a candle flame and the sun are alike in that they both are ________ sources of light.	incandescent
$\widetilde{eg}$	4.	When a blacksmith heats a bar of iron until it glows and emits light, the iron bar has become a(n) ________ source of light.	incandescent
$\overline{eg}$	5.	A neon tube emits light but remains cool. Unlike the ordinary electric light bulb, then, it is not an ________ ________ of light.	incandescent source
$\widetilde{ru}$	6.	An object is called incandescent when ________.	it emits light because it is hot
ru + $\widetilde{ru}$	7.	It has been found that an object, an iron bar, for example, will emit light if its temperature is raised above <u>800 degrees Celsius</u>. Therefore, we say that above ________ ________ (temperature) objects will become ________.	800° Celsius incandescent
ru + $\widetilde{ru}$	8.	An electric light bulb produces light when the fine wire, technically called a <u>filament</u>, inside the glass is heated to incandescence. This means, then, that the fine wire or ________ must exceed a temperature of about ________° Celsius to emit light.	filament 800

$\widetilde{eg}$	9.	In an electric light bulb when an electric current is passed through the fine wire or _____, it becomes _____, because it is heated to a temperature above 800° Celsius.	filament incandescent
ru + eg + $\widetilde{ru}$	10.	The hotter an incandescent light source becomes, the greater the amount of light it emits. For example, an object heated to 900° Celsius would emit more light than the same object heated to 800° Celsius because the _____ of light emitted depends on the _____ of the object.	amount temperature
$\widetilde{eg}$	11.	A nearly "dead" battery in a flashlight produces only a feeble light in the bulb, because the filament fails to get _____ enough to produce much light.	hot heated
$\widetilde{ru}$	12.	An incandescent light source can be made non-incandescent by cooling it, since the amount of light emitted depends on the _____ of the source.	temperature
ru + $\widetilde{eg}$	13.	The color of the light an incandescent object emits changes from red to white as the object's temperature increases. Thus, when a blacksmith heats a piece of iron it first glows _____ (color), and then as it gets hotter turns _____.	red white
$\widetilde{eg}$	14.	A weak flashlight battery produces only a dull _____ (color) glow in the bulb's filament and little light is emitted because the _____ of the filament is relatively low.	red temperature
$\widetilde{ru}$	15.	Both the _____ and _____ of light emitted by an incandescent object depends on the temperature.	amount color
ru + $\widetilde{eg}$	16.	In "emitting light" an object changes or, technically, "converts" one kind of energy to another. For example, in a flashlight the electrical _____ supplied by the battery is changed or _____ed into heat and light.	energy converted

eg	17.	The light from a candle flame comes from the __________ released by chemical changes as the candle burns.	energy
eg	18.	One "turns on" an incandescent light bulb by closing a switch so that __________ energy can be converted to __________ and __________.	electrical heat, light
eg	19.	The hot wick of a lighted candle gives off small pieces or particles of carbon. These __________s are heated to incandescence, so that we know their temperature must exceed __________ __________.	particles 800° Celsius
eg	20.	Any incandescent object can be made non-incandescent by cooling it below 800° Celsius. For example, if one places a piece of metal in a candle flame the carbon __________s will be __________ below incandescence, and will collect on the metal as soot.	particles cooled
eg	21.	Smoke from a candle is comprised of carbon particles which did not burn and cooled below __________ (temperature) after they left the heat of the flame.	800° Celsius