Construction and Evaluation of Programmed Materials in Biology

• *Chester A. Lawson, Department of Natural Science, Michigan State University, East Lansing*

Programmed learning was a topic for one of the NABT meetings with the AAAS in Philadelphia, December, 1962, and this paper was one of those given at that time. Prof. Lawson is Head of the Department of Natural Science at Michigan State.

Two forms of programs have been developed so far. One is called linear programming and the other branching programming.

The linear type of program was invented by Professor B. F. Skinner on the basis of a conditioned reflex theory of learning. In linear programs the student is given small bits of information, is asked to respond to this information by filling in a blank in a sentence, or giving a one word response to a question and then is immediately informed of the correct response. This is followed by another small bit of information, another statement requiring a response by the student, and again the student is informed of the correct response. This pattern is continued over and over again.

The reason for feeding the information in small bits is to make it almost impossible for a student to make an error. The reason for the blank in a statement is to require the student to make an active response to the information, and the reason for informing him immediately of the correct response is to reinforce the student's behavior.

According to Skinner one learns by becoming conditioned, and conditioning occurs most effectively when small segments of behavior are immediately rewarded or reinforced. The reward in this case is the correct answer. Also because of the assumption that learning is conditioning few or no errors should be permitted. According to Skinner if a student makes an error, the error may be learned rather than the correct response.

The branching type of programming was invented by Norman A. Crowder. In this type information is presented to the student in larger segments, a question is asked, and two or more answers are given from which the student makes a choice.

If he chooses correctly he is informed that he is correct and then is given more information, another question and another battery of choices.

If the student selects an incorrect answer he is informed that he is in error and also is given an explanation of why his selection was wrong. The student is then sent back to try again.

The information, the question and the battery of possible answers are usually presented on one page. The explanations of wrong answers and the new information, following a correct response, are on other pages. The student is directed to these other pages by page numbers that appear after each answer. Thus a program of this type cannot be read as one would read a normal book. Instead of going from page 1 to page 2 to page 3, etc., one might go from page 1 to page 7, to page 15, to page 2. For this reason this type of program has been called a "scrambled book."

Crowder presumably had no particular learning theory in mind when he designed the first scrambled book. He operated on the assumption that one can learn by reading and that it would be helpful to have frequent checks on what was learned. The questions do this, and if errors are made they are corrected and the student is given another chance.

Is one type of program better than the other? In general the answer is no. Students learn about as effectively with one type as with the other. However, we have found that if the subject matter is primarily descriptive information, that is new to the student, the linear program is easier to write. On the contrary if the subject is more complex and involves the student in reasoning, then the branching type is better.

The branching type also permits greater variability, such as side tracks, for some students who need additional information. Thus a single program can be written for students of different ability and background and satisfy the needs of all. It permits one to require greater discrimination on the part of the student. For example a series of answers can be given all of which are partially correct, but only one of which is completely correct.

Also the branching type of program permits error and takes advantage of the error. As one programmer remarked, "If we lived in a one choice world linear programming would suffice, but the world we live in is a multiple choice world and we all make errors." We have to learn to learn from our errors.

In writing programs of either type there are several rules that must be followed.

> "Before any writing is done the author should define the goals of the program as precisely as possible, in terms of the final behavior the students are to be able to display. . . . Failure to define the purpose and specific goals is a little like setting out on a long journey without deciding on a destination. It might prove to be an interesting trip but you might travel for a long time without getting far from your starting point and without ever reaching the distant places you wanted to visit." (*A Guide for Wiley Authors—in preparation of linear auto-instructional programs.* 1961)

As an example of this—we wrote a program on the relation of probability to the behavior of genes. After the students had worked through the program we gave them a short quiz which consisted of several problems to be solved. The students failed completely. The reason for this was that we had programmed the information, but had not programmed how to use the information in problem solving. You can't program for one type of behavior and expect to achieve another kind of behavior.

The program must be orderly. For any system of ideas that is to be learned certain concepts must be presented before others. Furthermore concepts must be broken down to subconcepts, so that there are no gaps and the student can move easily from one concept to another. Beginning programmers almost always move too fast. They present the information in pieces that are too large, and contain too many ideas. The size of the steps that a student can take will depend on the intellectual capacity of the student plus the knowledge he already has, so that for some students larger steps are possible. However, the basic rule still holds. Move by small related steps that leave no conceptual gaps.

Each frame, unless it is a review frame, should be limited to one concept, or one relation between concepts. And each frame should be related to the previous frame and to the following frame so clearly that no student, suddenly confronted with a fact or idea, wonders how he got there.

Another rule is repetition of concepts. Do not expect that because the student has responded correctly once to an idea, that he has mastered that concept. The same concept must be repeated. According to the literature three to five repetitions are usually necessary.

A device somewhat related to repetition but primarily intended to help the student develop a concept of the integration of the bits and pieces of information into a pattern is the worksheet. This is a separate sheet that is torn out of the program and on which the student records pertinent information as he works through the program. The worksheet when completed summarizes on one page the essential elements of the system of ideas and shows their relationship.

Another useful method for tying the subject matter together is the review and summary. In both the linear type of program and the branching type we have used, at natural breaks in the sequence, a series of sentences with blanks to be filled in as a method of review. Following the review is a short summary-discourse of the traditional type to further emphasize the integration of the concepts into a meaningful pattern.

After the program is written it must be tested. We have used two methods. One is to have large groups of students work through the program and to mark each response. By collecting and tabulating the marked responses we can discover where the error rate was high. A high error rate, say 10% or more, indicates that something is wrong with that frame or some related previous frame. Usually an inspection of the frame gives a clue to what is wrong.

The frame is then rewritten. After rewriting the faulty frame or frames the entire program is tried again on a new group of students.

Another method of testing a program is to have individual students work through the program in the presence of the programmer. When the student makes an error he tells the programmer, who then can discover from the student comments why he made the error. This gives a clue to the necessary correction, which may be needed on some previous frame rather than the one on which the student made the error.

We have found that mass testing is best for branching programs, but that individual student testing is best for linear programs.

As a result of mass testing we accidentally discovered a useful revision of the scrambled form. One very laborious aspect of preparing a scrambled book is the scrambling itself. We had to work out a system of form-numbering so we could keep track of the sequence, and so we did not place two or more items from the same battery of answers on the same page or on facing pages. To eliminate this labor for trial runs of programs we simply put the book in the traditional page sequence. Thus on page one was placed the initial information, the question and the possible answers. At the top of page 2 we placed the explanations of the incorrect responses; at the bottom of the same page we placed the correct response, the new information and the second question and possible answers.

This type of program has all the elements of the branching type, but is not scrambled. To our surprise we discovered that the students preferred this form by 7 to 1. Their reasons were that they didn't lose their places as easily, they did not have to turn so many pages back and forth, and they always had some idea of how far they had progressed. In the scrambled form the fact that a student is on the next to last page does not mean that he has only one more page to go. He has to be told when he has finished the last frame, because it could be in the middle of the book. Another advantage is in review. It is easier to go through a program rapidly for review purposes if it is paged in sequence.

There are several disadvantages to the nonscrambled form. One is that a student can make only one mistake per frame, because whatever selection he makes sends him to the next page where he sees the correct response. Another disadvantage is that it is easier for a student to cheat the program. Thus instead of making a serious effort to understand the material and to select the appropriate response, the student may simply make a half-hearted attempt and then turn the page, or he may make no attempt at all, but simply read through the program.

Because of the students' pronounced preference for the unscrambled form we decided to compare the effectiveness of both types on large numbers of students. We used about 500 students and five teachers. Half of them used the scrambled form and half the unscrambled form. The experiment ran for about five weeks. We gave four one hour examinations at intervals during the five weeks and a final examination at the end. There were no significant differences in the means of the scores of any of the examinations.

Experiments of this sort are fraught with loose variables, but if we can assume that one uncontrolled variable cancelled out another uncontrolled variable then the unscrambled form is just as effective as a learning tool as the scrambled form.

One additional point, that any of you who contemplate writing programs should know, is that it is difficult and time consuming. When you begin you should know that it will take you longer than it would take you to write the same material in the traditional textbook form. Then after you have made some judgment as to the length of time it will take you, multiply that time by five and you will reach a more accurate estimate.