6-1989

# Design of a Program of Instruction to Teach State Notation

Esther Hannah Shafer
*Western Michigan University*

# DESIGN OF A PROGRAM OF INSTRUCTION TO TEACH STATE NOTATION

by

Esther Hannah Shafer

A Dissertation
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Doctor of Philosophy
Department of Psychology

Western Michigan University
Kalamazoo, Michigan
June 1989

Order Number 9000833

# Design of a program of instruction to teach state notation

Shafer, Esther Hannah, Ph.D.

Western Michigan University, 1989



U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

ACKNOWLEDGEMENTS

I would like to express my gratitude to the members of my doctoral committee, Drs. Jack Michael, Dale Brethower, Paul Mountjoy, and Arthur Falk, for their helpful guidance throughout this project. I would particularly like to express my appreciation to my advisor, Jack Michael, for his support throughout my graduate training. Special thanks go to Chuck Butler for his willing assistance with many aspects of this project, as well as his constant encouragement.

Esther Hannah Shafer

TABLE OF CONTENTS

Table of Contents--Continued

CHAPTER

Table of Contents--Continued

Table of Contents--Continued

CHAPTER

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER I

## INTRODUCTION

The field of programmed instruction is broadly defined by Markle (1978, p. 1) as "the ... attempt to apply psychological principles to instructional practice." Beginning with the teaching machine described by Pressey in 1927, principles of learning have been applied to instruction in a variety of ways over the last 60 years. Although a variety of methods of programming have since been developed and advances in technology have made possible more sophisticated versions of teaching machines, the same learning principles which were utilized in the first attempts at programmed instruction are still the basis for effective programmed instruction today.

Although Pressey described his first teaching machine in 1927, the educational community did not recognize the potential of the teaching machine until Skinner's (1954/1968 and 1958/1968) renewed interest in the topic. In 1954, Skinner proposed that if the knowledge about basic principles of behavior acquired during the 1940's and 1950's were applied to Pressey's teaching machines, a powerful technology of learning could develop, and in 1958, Skinner discussed how principles derived from the operant laboratory could be applied to the design of programmed instruction. These principles include the shaping of complex behavior through use of successive approximations, chaining behaviors together to comprise one unit of behavior, the effective use of priming, prompting, and fading of prompts, bringing behavior under the control of both verbal and non-verbal classes of stimuli, following a desired response with intermittent reinforcement, and the scheduling of intermittent reinforcement so that persistence develops. Holland (1960) described several additional areas of knowledge which were directly transferable from the laboratory to the development of programmed instruction. These include control of the learner's observing behavior, and discrimination training procedures. Mechner (1967) notes that the principles of discrimination, generalization, and chaining are the basis for an analysis of any subject matter which is to be programmed.

As a result of the knowledge gained from the operant laboratory, Skinner developed programmed instruction presented in a teaching machine format which had a number of distinctive features. A high rate of overt responses were made as the learner worked through the program, and the learner received immediate feedback after each response. The construction of the program was critical; the learner had to respond to the appropriate stimuli so that control would develop. The terminal behavior which the learner was to achieve was defined, and successive approximations to that behavior were arranged through sequencing of the material. The first response sometimes had to be primed; later responses sometimes needed to be prompted with eventual vanishing of all prompts so that the learner was responding to similar stimuli that were to be encountered outside of the learning setting. Ideally, much of the reinforcement that a learner obtained was an automatic consequence of working through the program successfully.

After Skinner's elucidation of the features of programmed instruction and the success of his programming method, known as linear programming (DeCecco, 1964), the area of programmed instruction grew rapidly, and other programming methods were developed. All programming methods are applications of principles derived from the operant laboratory, and to a large extent, share many of the same features as Skinner's linear programming. Mathetics, a programming method designed by Gilbert (1962a,b), is distinguished from linear programming primarily by the rejection of a small step size. Gilbert's rationale for use of a large step size has been discussed in his two papers on mathetics, as well as by other instructional designers.

## The Role of Step Size in Mathetics

Before mathetics was proposed, the use of a small step size was widely accepted by instructional designers as necessary for learning to occur, and a small step size is an integral part of linear programming (Markle, 1978). Markle notes that the frame used in early programmed instruction was often held to 30 words or less, and that subject matter was automatically fractionated according to this limit. Gilbert (1962a) objects to this pre-conceived notion of step size, and instead, prefers to organize material in terms of the largest amount of material that can be presented at one time with resulting progress toward mastery of the final goal. Gilbert contrasts the traditional frame and the unit of instruction used in mathetics, called the exercise. He defines a frame as "a physical unit that refers to the space in which materials are exposed to the student," while an exercise "is defined as all the material designed to establish a single new operant in the chain of mastery" (p. 24). With this statement, Gilbert emphasizes that in mathetics, the material to be taught dictates the physical structure of the program, rather than the programmer beginning with a preconceived structure of how the material should be presented. Like Skinner, Gilbert is interested in establishing units of behavior in the learner's repertoire, each unit consisting of a stimulus and response. Each of these units is defined as an operant. However, according to Gilbert, the size of an operant is somewhat arbitrary. For example, in long division, one can feasibly conceptualize each single arithmetic operation as a single operant, or one can view the entire process of long division as a single operant. Gilbert's definition of an operant, therefore, frees the programmer to conceptualize the behavior to be taught in terms of larger steps if the programmer deems larger steps to be appropriate. Since one kind of behavior change may require more material than an equal behavior change of another kind, the physical size of the exercise is free to vary depending on the amount of material needed to establish that operant. Finally, the operant span is defined as the basic element of behavior change in mathetics, and is the largest gain towards mastery that can be produced in a single exercise. The programmer's task in mathetics, therefore, is to first identify all of the operants that comprise a task, conceptualizing these operants in terms of the largest units that can be mastered by a learner, or operant span. Presentation of the material on the printed page, or the exercise, can be arranged in any format which will accommodate the operant span.

## Validation of Mathetics

### Claims of Effectiveness

Gilbert (1962a) has claimed that "mathetics produces teaching exercises that exceed the efficiency of lessons produced by any known method" (p. 8). The efficiency of mathetically designed materials may be attributed to Gilbert's definition of the operant span. Since the operant span is determined by learners' ability to negotiate the teaching material in initial tryout, only material that is necessary to learning is included in the program. Gilbert has also claimed that "the more difficult the material, the greater the advantage gained by mathetics," and that "mathetics is applicable to all subject matters" (p. 8). He goes on to say that "we have compared mathetical exercises with the best available programmed learning materials and found that these (available) programs require twice to ten times as much learning time, five to thirty times as many exercises to cover the same subject matter, and the (longer) programs result in poorer recall" (p. 8).

Experimental Validation

Few investigators have attempted to directly investigate Gilbert's proposals contained in his 1962 papers, particularly his proposition that a large step size is more effective than a small step size. In one of the only published studies concerned with validating the procedures and theoretical bases of mathetics, Balson (1971) compared the effectiveness of the mathetics approach with linear programming in the teaching of mathematics to Grade 4 children. Three varieties of the same program were compared: (1) a mathetics program organized by backward chaining, (2) a mathetics program organized by forward chaining, and (3) a linear program. The amount of learning by all three groups was similar, but the time needed to complete the program was significantly less for both mathetically designed programs. There was also a significant difference between the forward and backward chaining programs in terms of time to complete the program, with a shorter time required for the forward chaining program.

A similar study by Hunt (1972) compared the effectiveness of linear and mathetics programming techniques to teach service writing to military cadets. Both linear and mathetically designed texts were constructed to teach the subject matter, and a total of 92 military cadets were randomly assigned to one of two groups. Amount of learning was measured by a posttest and efficiency of learning was measured by a comparison of time to complete each program. An analysis of covariance revealed that the mathetics technique was more effective than the linear approach in producing the required learning, and 39% less time was required to complete the mathetics program than the linear program.

Some evidence exists that at least for some tasks, teaching the task as a whole is more effective than teaching the task in parts. Cox and Boren (1965) evaluated this proposition with Army trainees learning to prepare missiles for firing. The trainees were divided into three groups: backward chaining, learning the task as a whole, and learning the task when divided into seven parts or operant spans. No differences were found between groups in terms of time required to produce the first perfect performance. Cox and Boren conclude that generalities cannot easily be drawn from this study because each task has unique characteristics, some of which are difficulty for the learners, degree of organization of the task, and amount and spacing of practice provided during learning of the task. They conclude that each task will vary with respect to the above characteristics and therefore, each task must be evaluated for the most effective teaching method.

In a similar study, Wilcox (1974) investigated the effect of three chain lengths (short, medium, and large), and three types of teaching strategies (backward chaining, forward chaining, and whole method). Each subject was instructed in two chains: one was a motor task in which they folded sheets of paper in specific ways; the second chain involved completing a number calculation comprised of a number of steps. Acquisition under whole method was found to be more rapid than under the forward chaining method as measured by total time to reach criterion, total prompts needed to reach criterion, and total spans practiced to reach criterion. Wilcox concludes that task characteristics should determine step size, and no one method should be recommended for all tasks. He notes that the relative advantages of different chaining methods may be a function of task characteristics.

## Part-Whole Research

An area within the human learning literature which has some relationship to Gilbert's proposal of large step size is referred to as part-whole research. In a review of the area, Stammers (1982) notes that the part-whole debate is well-established, with the first published study in the area appearing in 1900. In spite of the long-standing interest in whether training is best accomplished using part or whole methods, Stammers notes that the research still has not answered several important questions. First, there are no clear guidelines as to how task characteristics can be related to training methods. Despite the body of research, a technology has not developed which can assist programmers. Second, if part training is done, the research does not provide clear answers as to how the parts can best be combined to create the whole task. Third, even though the determination of the optimum size of the learning unit has been the central focus of research, no definite answers have emerged.

Stammers points out that most of the experimental work has involved tasks which are described as serial or procedural, in which the tasks require the learner to recall and perform a chain of operations, usually in an invariant order. Much of the research has been guided by theories proposed by Naylor (1962) and Annett and Kay (1956). These researchers have attempted to predict training methods in terms of task demands.

Naylor (1962), in a review of the literature on part and whole training methods, indicated that no general rule is possible for employing part or whole training methods, since the most efficient method appears to vary as a function of four variables. These are: (1) task variables, which are difficulty and organization, (2) subject variables, such as age and intelligence, (3) the learning situation, such as conditions of practice and amount of prior learning, and (4) criterion variables, which are time or error scores. Naylor acknowledges that difficulty and organization of tasks are arbitrarily defined, but possess some general properties. Organization refers to the extent to which task components are meaningfully related to one another in terms of type of response, total task performance, and response probability. Task difficulty is determined primarily in terms of the amount of practice needed for reaching a skilled level of performance. Low difficulty was regarded as requiring one hour or less, medium difficulty was regarded as requiring one hour to one week, and high difficulty was regarded as requiring more than one week to master the task. Naylor has proposed that several task variables may interact, and therefore a decision about whether to use part or whole training must depend on the particular task being considered. In general, he reports that the literature indicates a number of consistent findings. First, the more highly organized the task, the more efficient is the whole method. Second, tasks which are of moderate or high difficulty are best learned by the whole method. As task difficulty decreases, the whole method appears to be less effective. Third, there is probably an interaction between task difficulty and task organization, such that for highly organized tasks, whole training is increasingly more beneficial as difficulty increases. However, with tasks of low organization, part training is relatively more efficient than whole training for tasks of greater difficulty. Fourth, for tasks of low organization, there is evidence that part practice on the least proficient components results in the greatest improvement, while for tasks of high organization, whole task practice is necessary.

Stammers (1982) notes that Naylor's (1962) discussion of task difficulty and organization results in some ambiguities, since task characteristics are often not identifiable enough to easily categorize a task in this manner. He also points out that Naylor's concept of organization is from the expert's point of view rather than the naive learner.

In contrast to Naylor, Annett and Kay (1956) have attempted to analyze tasks from the viewpoint of a naive learner. They suggest that skill acquisition can be thought of as the informational value of signals during the learning process. Tasks are of two main types: (1) those in which one action produces a signal which clearly leads to the next action, and (2) those in which an action does not result in a signal which clearly leads to the next action. When actions do result in signals for subsequent responses, or are interdependent, part training is recommended. When actions do not result in signals for subsequent responses, or are not interdependent, whole training is recommended. Stammers suggests that Naylor's and Annett and Kay's training principles are complementary. When evaluating a task, he recommends that the organization of the task should be evaluated from the viewpoint of the naive learner. If the task has a high degree of organization to the naive learner, part training would seem appropriate. If the task does not contain a high degree of organization for the naive learner but does for the sophisticated learner, than whole training is recommended. Both Naylor and Annett and Kay determine task characteristics such as organization and complexity by an analysis of the task from the instructional designer's point of view.

Sheffield (1961) has asked a different question from Naylor and Annett and Kay, focusing on the best way to subdivide a task into parts when a part method is deemed most appropriate. Sheffield has proposed that tasks contain "natural units" which are readily integrated into a whole after being learned separately. A natural unit may often consist of the amount of content which can be learned with minimal loss due to interference from other units, but the natural unit can also exceed the amount of material that can be learned at once. Stammers concludes that experiments which have attempted to examine Sheffield's natural units have failed to demonstrate the usefulness of natural units for division of tasks. Again, research is difficult to interpret due to large differences that existed between tasks that were compared. Also, interpreting the research on part methods is difficult because of varying amounts of practice. In some experiments, practice was given after each part, while in other experiments, practice was given only after the entire task had been taught.

A summary of the research that has compared part and whole methods shows that part training does not necessarily have advantages over whole training, even for tasks that Sheffield might predict would be best taught by a part method. However, tasks used in these experiments have varied in terms of complexity, organization, and training methods used. Stammers (1982) notes that because the research has left questions unanswered, the part-whole research has not yielded a technology of training. No set of rules or principles of training exist which relate task characteristics to training methods. However, some general guidelines are suggested. Stammers recommends that the programmer may be able to decide whether a part or whole method should be used simply by examining the characteristics of the task. Tasks which are simple or short would not require a part approach, and therefore a whole approach can be used. Conversely, some tasks are too complex to present as a whole. Tasks which consist of unrelated parts cannot logically be presented as a whole. Stammers recommends that when either whole or part training is being considered, whole training has advantages in terms of efficiency because it places less demands on both learner and trainer time. The question of which part method should be used has not been resolved. The part-whole research generally substantiates Gilbert's (1962a) theory, in that for some tasks, whole methods result in learning that is more efficient than when part methods are used. Sheffield's hypothesis of natural units resembles Gilbert's notion of the operant span. Unfortunately, the research concerning natural units did not yield conclusive results which can support Gilbert's theory.

Practical Validation

Several instructional designers have proposed that programs with large steps are more efficient than programs with small steps, citing the popularity of such programs in practical settings as proof of their effectiveness. Lean programming is a series of steps which programmers can follow to produce programs that result in efficient learning by using the largest steps that the learner can take without failure (Rummler, 1965), and incorporates the principles that Gilbert detailed in his 1962 articles. Lean programming has been widely used to develop programmed learning for business and industry, and is reported to be more cost-effective than programs which use a small frame size (Rummler, 1965). MacDonald-Ross (1969) reports that in the United Kingdom, the number of industrial users of programmed learning increased greatly during the 1960's, with an increased use of mathetically designed programs.

## Factors Which Influence the Choice of a Programming Method

Stammers (1982) notes that at present, a set of rules or principles of training relating task characteristics to training methods does not exist. He recommends that the characteristics of each task should be analyzed individually in order to determine the most appropriate programming method. Before deciding on a particular programming method, the instructional designer may wish to consider several factors which can influence choice of a programming method.

### Type of Behavior Required of the Learner

Taxonomies of learning or classification systems enable instructional designers to consider the type of behavior required of the learner at mastery level (Duncan, 1972). Taxonomies provide guidelines to instructional designers as to how instruction should be designed for that particular task, given the nature of that task and the demands on learners. Taxonomies of learning originated in educational approaches to task analysis, of which Gagne's (1965) taxonomy is one of the most widely used. Gagne describes how the selection of the most appropriate programming method is determined by the behavior(s) that the learner should be expected to emit upon completion of the program, and describes seven categories of behavior. For example, the task may be one primarily of simple association learning (emitting only one response in the presence of a stimulus); discrimination training, in which many similar stimuli are presented and the learner must differentially respond to each; chaining, in which the task has a predetermined order of steps and the learner must emit these in sequence; concept formation, in which the learner's behavior must simultaneously be controlled by several stimulus features, and strategies, in which many concepts are relevant to a particular task. Although Duncan notes that tasks can seldom be neatly classified into one level of a taxonomy, the use of a taxonomy provides the programmer with a guide with which to make decisions about specific programming procedures. Duncan recommends Gagne's (1965) taxonomy as the most applicable to programming, since the levels of the taxonomy are classified according to preconditions of the learner, conditions of the instructional situation, and a description of the behavior which the learner must emit. Tiemann and Markle (1978) have organized a manual of task analysis for the programmer which is based on the use of a taxonomy as a starting point for selecting the most appropriate programming method or methods.

Amount of Theory to be Taught

When designing instruction, the programmer must decide the amount of theory to be taught directly. The terminal goal of instruction may be to verbalize theory about a particular subject matter, or it may be to perform a task or to learn a skill. If the goal of instruction is for the learner to perform a task or learn a skill, the instructional designer must determine whether directly teaching theory will help the learner perform the task.

If some amount of theory is deemed necessary for successful execution of the task, the programmer must then decide how much theory should be taught and what type of theory should be included in the program. Mechner (1965) recommends that when faced with decisions of this type, the programmer should do two types of analysis: (1) an analysis of the subject matter to determine all the possible types of theory that could be taught for that subject matter, and (2) an analysis of the behaviors that the learner will need to perform, given the occupation or settings in which the learner will be involved. Gilbert (1962a) also addresses this issue. He calls the behaviors that are involved in performing the task the synthetic repertory, and the amount of theory needed to perform the task as the analytic repertory. As a general rule, Gilbert recommends that if the goal of instruction for the learner is to perform a task or learn a skill, only the amount of theory necessary to enable the learner to perform at mastery level should be taught, and calls this the domain theory. By teaching the appropriate amount of theory, both savings in learning time and improvements in retention should be evident. Gilbert recommends that the programmer should include theory at the same time as the student is learning to perform new behaviors, so that the theory will assist the student in performing the behaviors in the presence of the desired stimuli.

## Programming For Specific Types of Tasks

Stammers (1982) notes that "any particular task will have its own characteristics and these may well determine the most efficient training method for it" (p. 186). Stammers notes that although a technology of training has existed for decades, no set of rules or principles exist which relate task characteristics to training methods. Lacking specific guidelines for matching method to task, a programmer may examine the instances in which a method has been used successfully with a particular type of task, noting the characteristics of that task. Linear programming has been widely used for developing a complex verbal repertoire, and may be most applicable to this type of subject matter. Holland and Skinner's (1961) programmed text to teach principles of behavior analysis is perhaps one of the most widely known examples of the use of linear programming to develop a verbal repertoire about a subject matter. Mathetics has been widely applied to the teaching of behavior chains in which all the component operants exist at considerable strength, and backward chaining has been thought of as synonymous with mathetics itself (Espich & Williams, 1967). Hunt (1972) has offered the opinion that mathetical programming would appear to be most successful in those situations involving clearly defined behavioral sequences, such as operating a machine. He questions whether mathetical programming would be appropriate or feasible in abstract, non-sequential tasks. However, Gilbert (1962a) notes that mathetical programming is by no means restricted to the establishment of behavior chains, and suggests that mathetical programming could be used to teach such subject matters as contract law. For subject matters which require problem-solving skills, Gilbert (1962a) states that a mathetical analysis will clarify the points of difficulty peculiar to the subject matter, and will guide teaching strategies that are best suited to these difficulties.

Other instructional designers have noted the unique requirements of programming for complex skills and subject matters, particularly when many decisions must be made or a complex procedure is to be followed. These instructional designers have not recommended a specific programming method for complex subject matters, but rather have suggested a variety of aids that the learner can use to clarify decision-making processes while working the program. These methods, which are often overlapping, have been called flowcharts (Tiemann & Markle, 1978; a decision-tree (Davies, 1967, 1972; Tiemann & Markle, 1978), a linked statement, or a series of numbered questions which show the relations between relevant conditions (Davies, 1967, 1972); a visual diagram or WHIF, used to illustrate a causal chain of events (Davies, 1967, 1972), and algorithms (Lewis & Horabin, 1977; Tiemann, 1977). Gilbert (1962b), in one of his original papers on mathetics, suggested making a checklist available to the learner, to be used at the learner's discretion. The checklist contains the operants of the prescription in the form of answers and in the correct sequence. Gilbert (1978), elaborating on his previously stated idea, categorized job aids in three ways: (1) directories, which help people make well defined discriminations, (2) ensamplers, which aid people in making well defined generalizations in addition to discriminations, and (3) queries, which help people consider all of the information available and make judgments. The job aids that Gilbert describes can be of many formats, including flowcharts, pictorial diagrams, checklists, charts which illustrate cause and effect situations, etc.

## Design Process in Mathetical Programming

A general interest in the process by which instruction is designed became evident during the late 1960's (Hartley, 1972). Duncan (1972) proposed that a concern with task analysis, or with the initial stages of defining and ordering what the learner must master, has been the major contribution of programmed instruction to learning. Gilbert (1962a) outlined the process by which mathetical programming is to be designed. The design process is an integral part of mathetical programming, and is comprised of four stages. The first three stages describe the task analysis of the subject matter, and the fourth step involves the actual construction of the program. The first stage, development of the prescription, includes specification of the behaviors that are necessary to demonstrate mastery of the subject matter. These behaviors are specified in units of the operant span, and includes specification of discriminations, generalizations, and chains of behavior inherent in the subject matter. A prescription is developed in a series of approximations, in which materials are constructed based on the programmer's estimate of the operant spans, and testing these materials with learners. During this testing phase, the programmer is advised to err on the side of overestimating operant spans, because an error of overestimation is easier to correct than an error of underestimation. The second stage involves the identification of the theory needed to execute the behavior successfully; only that amount of theory should be taught which is essential to execution of the behavior. Properly selected theory is essential in fostering generalization, increasing retention of the overt behaviors which the learner performs, and increasing the intrinsic reinforcing properties of the behaviors performed in the program. In the third stage, called characterization, the programmer analyzes the behavior to be taught in terms of critical discriminations and generalizations. The programmer must also be aware of existing behavior which threatens to compete with the learning and retention of the new repertoire, and any operants at strength in the learner's repertoire before the learner begins the program. These operants may either facilitate learning or compete with learning; the programmer can anticipate these events occurring and plan instruction accordingly. This analysis results in a detailed lesson plan that serves as a guide to the programmer for writing the exercises. The fourth stage is concerned with the specifics of lesson design; these are elaborated in another paper (1962b). Generally, the lesson plan includes the sequence in which the exercises will occur and details the specific stimuli and responses that are to be included in each exercise.

As previously mentioned, a method of programming entitled lean programming (Rummler, 1965) was developed which allowed instructional designers to develop programs with large steps, and shows many similarities to the principles of programming proposed by Gilbert. Lean program design consists of a sequence of distinct stages, and these stages have been described in detail by several instructional designers (Brethower, Markle, Rummler, Schrader, & Smith, 1964; and Markle, 1978). Task analysis in lean program design proceeds through a series of steps, and is an ongoing process throughout program development. Brethower et al. recommend that the steps of lean programming include: (a) a tentative statement of the program objectives and description of students, (b) determination of behaviors related to the subject matter through consultation with experts, (c) redefining the program objectives and identifying points of critical discriminations based on the consultation, (d) writing criterion frames which test each objective, and (e) programming the subject matter so that the learners will be able to perform satisfactorily on criterion frames. These designers emphasize that the programmer and experts are not able to predict the best sequence of programming or the optimum amount of material to be programmed; these are determine by testing with learners to see how much of the target behavior they already possess. The programmer's job is not over when the first draft of a program is completed. During all stages of program development, the programmer must gather data on student performance and continually revise the program. Program revision is a process of testing and modifying the program until the first draft of the program is tested and modified through individual tryout, and as the program becomes more refined, group tryout is appropriate. Markle (1967) describes a three-stage process of empirical testing of programs: (1) the developmental testing phase, the purpose of which is to develop a workable instructional program, (2) the validation testing phase, the purpose of which is to obtain data concerning the performance characteristics of the program with targeted subjects and in targeted settings, and (3) the field testing phase, which involves making the program available for general use with a wider range of subjects and in a variety of settings.

Statement of Program Objectives

Duncan (1972) notes that designers of instruction usually specify instructional objectives as an initial design step. The use of instructional objectives grew out of an interest in establishing criteria against which the effectiveness of training can be evaluated, particularly in the military. This approach is represented by Mager (1962), who advocated the use of objectives which specify overt behaviors that learners are to engage in and which can be measured in the evaluation process.

Analysis of the Behavior of Experts

When determining the general objectives of a program of instruction, Markle and Tiemann (1970) recommend that the programmer determine how, and under what conditions, an expert or authority in the field behaves with respect to the subject matter. Ideally, after completion of a program of instruction, students who complete that program would perform as well as experts in that field. To design such a program, the programmer would need to know the specific behaviors that experts engage in, and in what situations. Obtaining this information requires some special analysis of the behavior of the expert. As Skinner (1968c, p. 206) noted, an expert performer is seldom aware of the behaviors that constitute skilled performance, and is often unable to describe or teach these skills to others. Therefore, as these authors have recommended (Brethower et al., 1964; Gilbert, 1962a, 1962b; Markle, 1978; Markle & Tiemann, 1970; Resnick, Wang, & Kaplan, 1973; Tiemann & Markle, 1978), the analysis of the behavior of the skilled or expert performer should be included in the initial stages of program development. The programmer must interact with the expert in a manner which will yield as many components as possible of the terminal behavior, even if they are performed quickly or covertly (Resnick et al., 1973). Gilbert (1962a) recommends that the programmer can then include those behaviors in the program, resulting in learners who can behave in much the same manner as the expert performer.

Development of Criterion Frames

The information obtained from observing the behavior of expert performers and the general objectives of the program can be further analyzed to produce criterion frames. These are frames which test the mastery of each program objective. Brethower et al. (1964) recommend that the criterion frames require the same types of behaviors as the task that the learner will perform when the program is completed. For example, if the general objectives of a program indicate that learners should be proficient in performing a certain behavior, then the criterion frames should not test stating of the procedures or rules that one must follow to complete the task correctly. Gilbert (1962b) notes that all responses which comprise a mastery response need not be overt; covert responses are actually more desirable than overt responses because of the ease in which they can be made. Brethower et al. (1964) notes that a learner who can respond correctly to examples of the concept being taught has probably induced the relevant rule or principle and has perceived the correct conditions under which application of that rule or principle is appropriate. Markle (1978) also advocates a heavy emphasis on application of principles rather than stating of principles, if the general goals of the program are also of this nature.

Construction of Teaching Frames

A comparison of the programming strategies recommended by several sources yields several common features. The first is the emphasis on "leanness", or tendency to err on the side of providing too little instruction during initial design of the program (Rummler, 1965). The programmer can best determine what changes in the program are needed to produce success by analyzing student performance during initial testing. To produce a lean program, Brethower et al. have recommended that only the number of teaching frames necessary to produce mastery performance on the criterion frame should be provided. Gilbert (1962a) recommends that teaching frames be presented in a three step progression which prepares the learner for mastery performance on the criterion frame. The desired behavior is initially fully demonstrated, prompted in the next frame or in several subsequent frames, and finally performed without prompting, or "released." This "demonstrate - prompt - release" sequence is followed for each objective. In the initial stages of program design, a minimum of prompting is included. Additional prompts can be provided if they seem necessary as a result of student error during testing. As the program progresses, behaviors which were demonstrated, prompted, and completed without prompting in previous objectives can now be incorporated into subsequent, more complex objectives.

## Purpose of State Notation Program

State notation (Michael, 1986; Snapper, Kadden, & Inglis, 1982) is a notation system which is used to visually represent procedures such as those used in an operant laboratory setting. As Snapper et al. have noted, the major advantage of such a system is to clarify procedures which are difficult to explain with words and are easily misunderstood by the reader when presented in a descriptive format. State notation is useful in teaching basic behavioral principles and experimental procedures, facilitating more effective communication between researchers, and in programming experimental procedures on digital computers (Snapper et al., 1982). Michael (1986) has compiled a manual which details the major events and relations of state notation. The manual then presents behavioral procedures of relatively increasing complexity, with accompanying state diagrams of these procedures.

Before a program of instruction is written for a particular subject matter, a need for the program must be demonstrated. Espich and Williams (1967) present a series of conditions which should be met in order to determine whether the need for a program exists. These conditions include: a subject matter which will remain relevant over time, a subject matter which is difficult to master and time consuming to teach with traditional instructional methods, and the lack of other available programs which will teach the subject matter. State notation meets these conditions as stated by Espich and Williams. Even though the aforementioned articles and manuals on state notation currently exist, they are most appropriate for the learner who is already somewhat proficient in state notation and knowledgeable in behavior analysis. Learners who have never been exposed to state notation and those who are just learning the elementary behavior analysis principles may find these materials too complex to quickly master the skill of reading verbal descriptions of procedures and drawing state diagrams. The need exists for a set of instructional materials which teach the basic components of state notation in an efficient manner, and which are suitable for the undergraduate student in behavior analysis. If the essential features of state notation can be easily mastered by a student in an introductory course in behavior analysis, the student will then be able to clarify behavioral principles and procedures as they are presented in the course, possibly resulting in quicker mastery of the course material. Of course, students in a more advanced course in behavior analysis may benefit from these materials in the same way. These materials can also be used as a convenient aid for those who have previously worked with state notation but have forgotten many of the essential features, or for those who would like to become more proficient in programming digital computers to conduct operant research.

## Design Considerations

The subject matter of state notation has several features which tend to indicate that a mathetical approach (Gilbert, 1962a; 1962b) is an appropriate programming method. State diagramming requires that many decisions must be made, and that these decisions are based on rules, many of which are of the "if-then" variety. Even though mathetics has primarily been used to teach tasks whose components must be performed in a fixed order and state notation is not a skill of this nature, a general order exists in the process of state diagramming. Mathetics is much more appropriate as a programming method to teach state notation than linear programming, which is most suitable when a large intraverbal repertoire (Skinner, 1957) is to be taught about a subject matter, as in the Holland and Skinner program (1961). State notation can be considered a complex, or problem solving task, since a number of decisions may need to be made at various points in diagramming. Some of the procedures which are involved in programming complex subject matters were deemed appropriate in programming state notation, such as using decision trees or flowcharts (Duncan, 1972).

The design process of lean programming seemed particularly applicable to designing instruction for state notation. State notation is a complex subject matter, and many questions remain concerning the optimum methods of programming for complex subject matters. The design steps involved in lean programming allow design decisions to be made during the design process itself, depending on learner behavior. Because the design process in lean programming recommends that the programmer initially err on the side of presenting too little instruction, the process minimizes the tendency of the programmer to be controlled by a particular design framework, and instead, to discover the optimum programming method through the design process itself.

## Research Objectives

The original purpose of this study was to produce a program of instruction to teach state notation. Once the subject matter was decided, the most effective programming method to teach the subject matter was investigated and the design process as previously described was followed. During the process of program development and revision, the question arose as to whether the program as written reflected the largest operant spans possible as described by Gilbert (1962a). Two successive versions were written, the second written with an attempt to make operant spans larger. The secondary purpose of the study was to determine if the second version would result in more efficient learning.

# CHAPTER II

## METHOD

### Program Design

The initial stages of program design involved a sequence of steps which included: writing general objectives, analysis of the behavior of two expert state diagrammers, expanding the general objectives into program objectives and criterion frames, and writing a minimal number of teaching frames for each program objective. At this stage in program design, these teaching frames were piloted with twelve volunteers from the Psychology 151 class at Western Michigan University titled "Introduction to Behavior Analysis", Fall Semester, 1986. Students in Psychology 151 learn many of the basic terms and concepts in the field of behavior analysis and many of the basic functional relations in the field. When the materials were piloted, students had already had some exposure to state notation as a means of clarifying these concepts and relations. The general objectives, protocol for analyzing the behavior of experts and summary of these findings, program objectives, and criterion frames used in this pilot study are included in Appendix A. The information obtained from this pilot study was used to develop a complete version of a state notation program, which was piloted with four graduate student volunteers during the Winter Semester, 1987. The experimenter observed as each student worked through the program, and collected data on time to complete each objective, score on a comprehensive post test, comments about the program as the student worked through the program, and a post-questionnaire which further assessed the student's opinions about the clarity and effectiveness of the program. These data were used to modify the pilot program to produce the programmed materials which were used in the present study.

From the outset of program design, the assumption was made that the subjects in this study and any future users of the program would possess a certain level of skill in behavior analysis. No attempt was made to teach principles and concepts of behavior analysis such as reinforcement, punishment, and simple procedures used in respondent and operant conditioning. However, names of behavioral procedures such as VI5', IRT>T, etc., were not used because of the knowledge that some of the potential subjects in this study would not be familiar with these terms.

## Independent Variable -- Two-Group Comparison

The development of a program of instruction to teach state notation raised a question which this study attempted to answer: Will a program in which each step contains the largest operant span possible result in the most learning in the shortest amount of time? To answer this question, two versions of a program were developed and tested. Both versions were based on the same general objectives and taught the same content, but the second version was an attempt to closer approximate Gilbert's (1962a) recommendation that the largest operant span be used for each step. In order to increase the operant span of each step in the second version, several program objectives from the first version were rearranged and combined. The first version had seven sections and the second version had five sections, resulting in a shorter program. Also, the presentation of the material within sections was different in each version. The first version typically presented one example at a time, with occasional comparison and contrasting of the various types of examples within the section. In the second version, an attempt was made to introduce each new section with an overview of as much of the task as possible. For example, a section was sometimes introduced with one example or several examples that illustrated all of the important features of the objective that was to be taught within that section. For other sections in the second variation, a decision tree or list of rules that would assist the learner in diagramming was presented at the beginning of the section. Several such lists and decision trees in a job-aid format were presented throughout the program. The learner was encouraged to take these job aids out of the program and use them at any time while working through the program. Both versions of the program were designed so that, as well as could be estimated, no more than five hours would be required to complete the entire program. Sample pages from both versions of the program are included in Appendix B.

## Subjects

Eighteen undergraduates and ten graduate students participated in the comparison of these two versions. Subjects were recruited from the Psychology department at WMU on a voluntary basis. Subjects were recruited from two sources: (1) undergraduate students who were enrolled in Psychology 151 (Introduction to Applied Behavior Analysis), or graduate students who were taking Psychology 151 for Psychology 510 credit, and (2) graduate students in the department who were interested in learning state notation. Any student who had received prior instruction in state notation in any class was ineligible to participate. Psychology 151/510 was considered a likely source of volunteers since state notation was to be taught during the second half of the semester using conventional methods such as lectures, reading material, objectives, and exams. Through an announcement by the professor in class, students enrolled in Psychology 151/510 were informed of the upcoming units on state notation and the availability of additional instruction in state notation. The announcement was made at approximately mid-semester; two weeks before the topic of state notation was to be introduced. State notation had never been mentioned in class prior to this announcement. After a brief explanation of the history and purpose of state notation, students were informed that they would be starting the first unit of regular instruction on state notation in two weeks and that completion of the program should help them in mastering the material that they would be studying in class. Students were asked to volunteer if interested. All students from this pool completed the program during the two weeks before the first unit on state notation was introduced in class. The graduate students who were not 510 students were independently recruited through personal contact by the experimenter, and completed the program at their convenience. Each subject in the two-version comparison was randomly assigned to Group one (Version I) or Group two (Version II) of the program.

The procedure for subject selection and research protocol was submitted to the Human Subjects Institutional Review Board at Western Michigan University and was deemed exempt from review (Appendix L).

## Setting

Research was conducted in the Psychology Department at WMU, in an empty classroom.

## Materials and Administration

Each student received a packet of materials, containing either Version I or Version II of the program, and was instructed to read the first two pages. Both versions were preceded by a one-page "Instructions to Learners," which explained the nature of the study. The following page, "Pretest and Questionnaire," explained the nature of the pretest and questionnaire that the student was about to receive. The experimenter then handed a three-page questionnaire to each student. A different version of the questionnaire was used for undergraduate and graduate students, with the graduate student questionnaire more detailed to reflect a wider range of experiences. Both questionnaires contained questions about courses taken and experience in conducting operant research. Both versions of the pre-questionnaire are in Appendix C. After the subject completed the pre-questionnaire, the experimenter handed the student the pretest, which was the Diagramming subtest of the Computer Programmer Aptitude Battery (1967). This subtest was chosen as a pretest for this study because it requires subjects to follow a flowchart and determine what information is missing from several places within the diagram; a skill which intuitively has similarities to state diagramming. The subject was instructed to read the first two pages of the subtest, which contained instructions for the subtest and a sample problem with answers, and to notify the experimenter when s/he felt comfortable with the task. The subject was allowed as much time as desired to read these two pages. Then, the subject was given 15 minutes to complete as many questions as possible within that time period. After completion of the Diagramming subtest, the subject was instructed to read a page called "Construction of the Program," and to begin the state notation program. No time limit was placed upon completion of the program. After the subject completed the entire program, a comprehensive posttest and an exit questionnaire were administered. The posttest contained five verbal descriptions of procedures which the student was to diagram. The procedures were designed to contain most of the important features of state notation taught in the program. The exit questionnaire solicited each subject's opinions about the clarity and usefulness of the program. The posttest is in Appendix D, and the exit questionnaire with data included is in Appendix E.

# Experimental Procedure

## Dependent Variables

### Posttest Scores

The major dependent variables were the score on the posttest items and the time to complete the program. Scoring criteria for the posttest are contained in Appendix F. All posttests were independently scored by a graduate student in the Psychology program at Western Michigan University, and who had been a subject in the two-group comparison. The experimenter provided training to the subject by first reviewing his errors on the posttest with him and administering the posttest to him again until he scored 100%. After receiving an explanation of the scoring criteria, he then scored several posttests, and his scoring was compared with the experimenter's scoring and any discrepancies were discussed.

For each subject, time to complete each individual objective of the program and time to complete the entire program were also recorded.

### Follow-up Comparison

After three weeks of state notation instruction in Psychology 151, a follow-up comparison on proficiency in state diagramming was obtained between two groups of Psychology 151 students; one group of ten students had completed a version of the program, and another group of ten students had not. Students were chosen according to number of points accumulated in the course up to the point in the semester right before state diagramming was introduced; a student who had completed the program was matched with a student who had not completed the program. In addition, all of the 20 students had attended all of the in-class lectures which covered state notation.

Students ranged from having an "A" average to a "D" average in the course, based on the number of points accumulated to that point. Proficiency in state diagramming was assessed by an examination question which presented a description of a behavioral procedure that was to be diagrammed. The diagram was judged to be relatively complex; it contained many features of state diagramming that could be drawn in one state set. The examination item and protocol for scoring are presented in Appendix G. A graduate student in the Psychology program at Western Michigan University independently scored the examination item.

A follow-up questionnaire was administered to each student in the class during the examination a week before the follow-up diagram was administered. This questionnaire solicited their opinions concerning the effectiveness of the instructional methods that they had received in state notation and their perceived level of proficiency in various aspects of state diagramming. This questionnaire and summary of results is presented in Appendix H.

## Data Analysis

Post-test performance was analyzed by employing an analysis of variance. Two independent variables were: (1) version of program and (2) academic level. The first independent variable consisted of either of the two versions of the program. The second independent variable compared the performance of undergraduate students with that of graduate students. Because of unequal cell sizes (number of undergraduate versus graduate students), the analysis of variance was computed using a regression approach (Huitema, 1980). This approach was chosen because it most closely approximates the assumptions of the standard analysis of variance model. Additionally, two analyses of covariance were conducted with version as the independent variable. For one analysis, the covariate was a composite score obtained from the entry questionnaire and the score obtained on the Diagramming subtest on the Computer Programmer Aptitude Battery (1967). For the second analysis, the covariate was total points attained in Psychology 151 at the end of the semester. However, the data set for this analysis did not include all 28 subjects, since not all subjects were enrolled in the class. The data set contained 22 subjects, and the missing six subjects were all graduate students. An analysis of covariance could not be used for the comparison involving graduate and undergraduate students because the covariate was highly correlated with that independent variable. The scoring protocol used to obtain the covariate score from the entry questionnaire is contained in Appendix I.

A second two-way ANOVA was computed with Version (one and two) and types of skills tested in the posttest (Question one, which tested skills related to diagramming single state sets versus Questions two through five, which tested skills related to diagramming parallel state sets) as independent variables. The purpose of this test was to determine whether a difference in types of skills learned existed between versions.

Two Pearson product-moment correlations were computed on the following combinations of variables: total time to complete the program versus level in school (graduate or undergraduate), and total time to complete the program versus posttest score.

Two additional Pearson product-moment correlations were computed on the following combinations of variables: posttest and number of points in class, and pretest and posttest scores. These correlations were computed in order to reveal sources of within-group variability.

## Field Testing

Three phases of field testing were conducted since the two-group comparison was done in Winter Semester, 1987, in psychology classes at WMU. The first and third phases were conducted in the Psychology 151 class, and the second phase was conducted in a graduate level class to teach SKED programming. During each phase of field testing, the instructor of that class requested that the state notation program be made available to his class. The program was revised before each phase of field testing, based on feedback obtained from questionnaires and personal interviews with students and the instructor of the class in which it had been previously taught.

For the first phase of field testing, conducted in the Psychology 151 class of Winter 1988, Version II of the program which had been tested in the two-group comparison was revised to produce a new version. The program of instruction was assigned to the class at the professor's discretion. In addition to the assigned sections of the state notation program, students received two additional sources of instruction concerning state notation; lecture material and assigned readings from material on state notation written by the course instructor. A three-week unit on state notation and schedules of reinforcement was conducted. Students received a weekly quiz in which questions concerning state notation were presented. These questions were of three basic types: (1) factual questions about state notation, such as "List all of the inputs in state notation"; (2) Drawing state diagrams from a description of the procedure or name of schedule; and (3) Writing a description of a state diagram. To assess student satisfaction with the program of instruction, a questionnaire was administered to the entire class at the end of the semester. A summary of data from this questionnaire appears in Appendix J.

For the second phase of field testing conducted in Summer 1988, the program was revised based on comments obtained from questionnaire data obtained from the 151 class of Winter 1988, and the professor's comments. The instructor of the Summer 1988 class assigned the entire program of instruction to the class as their first assignment. Students were not examined over the program per se -- it simply served as an introduction to SKED programming. Students informed the instructor of any problems or errors within the program, and this list served as the basis for further revisions of the program.

The third phase of field testing was conducted with the Psychology 151 class, Fall Semester, 1988. Slight revisions of the program were made from the version that had been used in the summer. As with the 151 class of Winter 1988, the professor used the program as one of several methods of teaching state notation. The final revision of the state notation program, which has been used in the Fall 1988 and Winter 1989 Psychology 151 classes, appears in Appendix K.

CHAPTER III

# RESULTS

## Comparison of Version I and Version II

### Posttest Scores

An analysis of variance was performed using two independent variables: (1) graduate versus undergraduate and (2) Version I versus Version II. No statistically significant differences were found on the comparison of Version I with Version II, and no statistically significant interaction was found between the two independent variables. The cell means and total means for this analysis are listed in Table 1.

A comparison of means for undergraduates versus graduate students indicates that graduate students showed higher performance than undergraduates (see Table 1). These results were statistically significant ($p<.01$). A comparison of means for graduate students indicates that graduate students who completed Version I scored higher than graduate students who completed Version II, though not significantly so. Undergraduate students scored higher on Version II than Version I, although these results were not statistically significant.

An analysis of covariance was performed using the pre-questionnaire and Computer Programmer Aptitude Battery (1967) as a covariate, comparing Version I with Version II. The results were not significant. Adjusted means are presented in Table 1. A second analysis of covariance was performed using total points accumulated in Psychology 151 as a covariate, comparing Version I with Version II. The results were not significant. This data set contained 22 instead of 28 subjects, because only students enrolled in Psychology 151 could be included. As a result, means differed from those of the entire data set as included in Table 1. Means were 17.18 for Version I and 18.64 for Version II. Adjusted means were 17.23 for Version I and 18.59 for Version II.

Table 1

Total Posttest Means

| Version | Undergraduate | Graduate | Total | Adjusted Total |
|---|---|---|---|---|
| I | 12.89 | 27.80 | 21.07 | 13.22 |
| II | 19.89 | 24.00 | 22.00 | 15.66 |
| Total | 16.39 | 25.90 | | |

Note: Total possible points = 40

Mastery Level

Using 90% or above as a mastery criterion for posttest performance, very few students scored at the mastery level on overall posttest scores or on individual questions of the posttest. There was considerable variability within scores, ranging from 8% to 90% correct. Figure 1 illustrates the range of scores for undergraduates and Figure 2 illustrates the range of scores for graduate students. No undergraduates attained mastery on the posttest as a whole. Of the graduate students, twenty percent who had Version I achieved mastery on the total posttest, and forty percent who had Version II achieved mastery on the total posttest. For individual questions, mastery was achieved most often on Question three for Version I and on Question five for Version II. Question three tested the skill of drawing a parallel state set that does not interact, and Question five tested the skill of drawing a parallel state set when a decision diamond is needed. Data are summarized in Table 2.

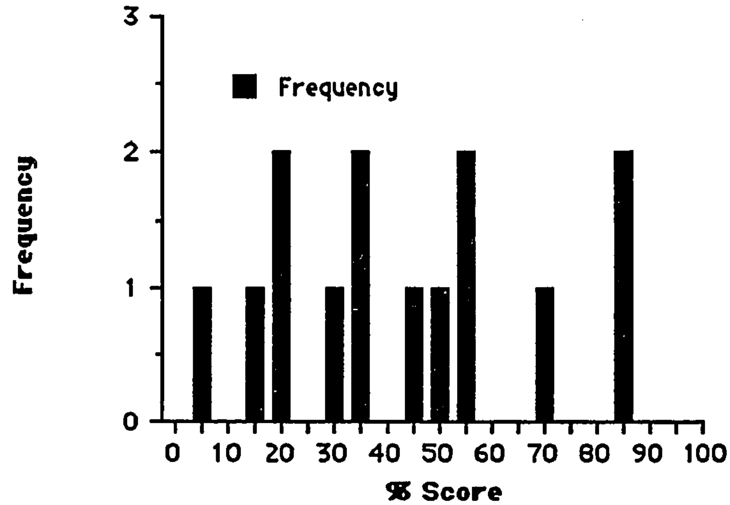



Figure 1. Frequency Distribution of Posttest Scores for Undergraduate Students.

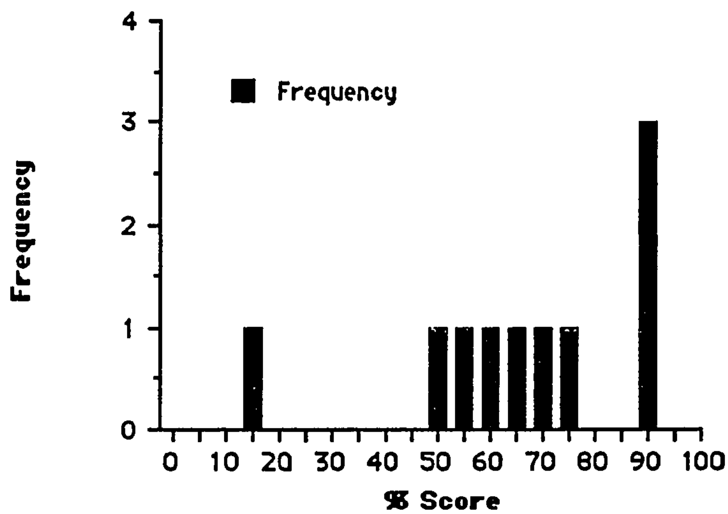



Figure 2. Frequency Distribution of Posttest Scores for Graduate Students.

Table 2

Percent of Students Who Attained Mastery on Posttest
as Compared by Version

| Version | I | | | II | | |
|---|---|---|---|---|---|---|
| | Undergraduate | Graduate | Combined | Undergraduate | Graduate | Combined |
| Question | | | | | | |
| One | 0 | 20 | 7 | 11 | 40 | 21 |
| Two | 0 | 0 | 0 | 11 | 40 | 21 |
| Three | 11 | 40 | 21 | 22 | 20 | 21 |
| Four | 0 | 0 | 0 | 11 | 0 | 7 |
| Five | 0 | 40 | 14 | 33 | 60 | 43 |
| Total posttest | 0 | 20 | 7 | 0 | 40 | 14 |

Note. Total undergraduate N = 18, total graduate N = 10, halved for each version.

State Notation Skills

Version II of the program resulted in higher means for three of five questions on the posttest (Questions 1, 2, and 5). These data are displayed in Table 3. Question one required the student to draw a complex single state set which tested a number of basic drawing skills taught in Sections one through four of both versions of the program. Question two tested the student's ability to draw a parallel state set in which the response involved duration, and for which a Z pulse was necessary. Question five tested the student's ability to draw a parallel state set in which a decision diamond was necessary. Since Questions two through five tested skills related to parallel state sets, they can be examined as a whole to determine the extent of skills acquired in drawing parallel state sets. When means on Questions two through five were examined as a whole, results showed that Version II means were higher than Version I means.

Table 3

Means For Individual Posttest Questions

| Questions | Version I | Version II | Total Possible |
|---|---|---|---|
| One | 8.71 | 11.43 | 18 |
| Two | 1.71 | 2.36 | 5 |
| Three | 3.29 | 3.14 | 5 |
| Four | 2.29 | 1.86 | 7 |
| Five | 2.50 | 3.21 | 5 |
| Two through five | 9.79 | 10.43 | 22 |

Scores on Questions two through five as a whole can be compared to scores on Question one in order to determine whether one version of the program was more successful than the other in teaching skills related to single or parallel state sets. For both subsets of skills, mean percent correct for Version II was higher than mean percent correct for Version I. For Question one, mean percent correct for Version II was 63.4, and mean percent correct for Version I was 48.5. For Questions two through five, mean percent correct for Version II was 47.4, and mean percent correct for Version I was 44.5. However, the results of a two-way ANOVA for independent measures comparing the effectiveness of the two versions in teaching these subsets of skills were not statistically significant for either main effect or interaction effect. Independent variables were version and type of skill tested in the posttest (Question 1, which tested skills relevant to diagramming single state sets, and Questions two through five combined, which tested skills relevant to diagramming parallel state sets).

A closer examination of whether one version of the program resulted in greater mastery of particular skills was accomplished by a breakdown of the scoring for each question on the posttest. As described in Appendix F, each of the five posttest questions had from 5 to 15 separate skills that were tested, for a total of forty skills. Version II resulted in significantly higher posttest performance for five skills when a one-tailed t test was used. Three of these skills were tested in question one, which was concerned with skills related to diagramming of single state sets but not exclusively restricted to single state sets, and were taught in the first four sections of the program. These skills are: (1) diagramming "START: on S" when appropriate, (2) drawing a transition to a previous state when a state has two inputs, and (3) diagramming the entire sequence of events in order in a complex state set. In question two, which contained five separate skills, one of these skills was performed at significantly higher levels by students who had completed Version II. Subjects who completed Version II were better able to identify that a Z pulse was needed when a response involved duration, such as pressing and releasing a lever. In question four, which contained seven skills, one of these skills was performed at significantly higher levels by students who had completed Version II. Subjects who completed Version II were better able to identify that three separate state sets were needed when diagramming a procedure that involved completion of two separate schedule requirements before reinforcement is delivered.

On every measure which compared Versions I and II, Version II showed higher variability in terms of standard deviations, although the difference in variability was not statistically significant.

Regardless of the version completed, skills relevant to single state diagramming appeared to be mastered to a greater extent than skills relevant to diagramming complex state sets. For all twenty-eight subjects, mean percentages were calculated for Question one and for Questions two through five combined. The mean score for Question one was 56% as compared to a mean percentage of 46% for Questions two through five combined. However, significance was not achieved when the two groups of skills were compared when a two-way ANOVA for independent measures was conducted.

## Time Spent

Means show that students in Version II spent more time working on the program (238 minutes for Version II as compared to 190 minutes for Version I) although this difference was not significant. No significant difference was found between the versions in time spent completing the posttest. Most notably, no significant correlation exists between time spent working on the program and posttest score.

## Subject Characteristics

Graduate students scored significantly better than undergraduate students on the posttest ($p<.01$), when a two-way analysis of variance was conducted, comparing the two class levels of students. Interestingly, graduate students also spent more total time working the program than undergraduate students, according to an examination of means (233 minutes for graduate students as compared to 204 minutes for undergraduate students). However, these results were not statistically significant. No significant correlation exists between time spent working on the program and level in college.

## Follow-up Comparison

The Psychology 151 class of Winter 1987 contained students who had participated in the two-group comparison. To measure whether having completed the state notation program before the topic was introduced in class ultimately resulted in better mastery of the subject area, a comparison was made between some students who had completed the program and some students who had not. Ten pairs of students were matched according to the total number of points accumulated in the class before state notation was introduced. One student in the pair had completed the program and one had not. Students in pairs had no absences during the time that state notation was covered in class. An examination question was designed which tested relevant skills in diagramming single state sets. A copy of the examination question appears in Appendix G. A dependent t test was conducted in order to compare the examination question scores of students who had completed the program with scores of students who had not completed the program. Results were not statistically significant. When total points earned in class before state notation was introduced was correlated with score on the examination item, results were highly significant, indicating that students were more likely to do well on the examination item if their level of performance in the class had been high from the beginning of the semester.

## Measures of Learner Satisfaction

Learners who participated in the two-group comparison were given a questionnaire to complete immediately after taking the posttest. This questionnaire measured their opinions about the effectiveness of the program and their perceived skill in state notation. Summary data appear in Appendix E. Two additional measures were obtained with students in two Psychology 151 classes; Winter 1987 and Winter 1988.

The Winter 1987 class contained some students who had participated in the two-group comparison, and some students who had not. Students in the Winter 1988 class had received a combination of the state notation program, lecture material, and other written material by the instructor about state notation. The questionnaire given to these students measured their opinion of the effectiveness of the state notation program, their perceived skill in state notation, and their opinion of how the state notation program compares with the other methods of learning state notation that they experienced in the class. Summary questionnaire data for students in the Winter 1987 class appear in Appendix H, and summary data for students in the Winter 1988 class appear in Appendix J.

Two-Group Comparison

Students who participated in the two-group comparison indicated general satisfaction with features of the state notation program, regardless of whether they completed Version I or Version II. Table four displays those items relevant to satisfaction with features from the program, as contained in the questionnaire administered immediately after completing the state notation program. Problems with both versions of the program were reported in response to Question one, in which almost half of students who had Version I and half of students who had Version II reported that the instructions in the program were difficult to follow. A large difference between versions was reported in question five. Seventy-nine percent of students who had Version I reported that they needed to look at most of the answers to the items before they understood the items, compared to 36% of students who had Version II. Students from both groups reported that they usually felt confident that their answers were correct as they worked through the sections of the program that taught single state sets, and felt less confident as they worked through the latter sections of both versions.

Table 4
Satisfaction With Features of the State Notation Program
as Reported by Learners in the Two-Group Comparison

1. Were the instructions and explanations in the program easy or difficult to follow?

| | Easy/Generally Easy | Somewhat Difficult | Difficult |
|---|---|---|---|
| Version I | 50% | 7% | 43% |
| Version II | 50% | 0% | 50% |

3. Did any of the material seem unnecessary? (Do you feel that you could have learned the same amount even if some of the material were cut out? If so, what parts?

| | All necessary | Some Unnecessary | More Needed |
|---|---|---|---|
| Version I | 100% | 0% | 14% |
| Version II | 86% | 7% | 7% |

4. Were any important concepts about state notation omitted from the program, or not covered in as much detail as they should have been?

| | None Omitted | Don't Know | Some Omitted |
|---|---|---|---|
| Version I | 21% | 21% | 58% |
| Version II | 28% | 35% | 37% |

5. Did you feel that you knew how to answer MOST of the items, or did you have to look at the answers before you understood the items?

| | Knew Most | Looked at Some | Looked at Most |
|---|---|---|---|
| Version I | 14% | 7% | 79% |
| Version II | 43% | 21% | 36% |

Table 4--Continued

6. Was the presentation of the material interesting/could have been made more interesting/dull? Please explain.

| | Interesting | Could be More Interesting | Dull |
|---|---|---|---|
| Version I | 86% | 14% | 0% |
| Version II | 93% | 7% | 0% |

7. Did you find the job aids helpful? Did you use them?

| | Used Them/Helpful | Didn't Use Them |
|---|---|---|
| Version I | 93% | 7% |
| Version II | 93% | 7% |

Students who participated in the two-group comparison also reported in question eight that they acquired many skills related to the basics of state notation, but still needed practice on skills related to more complex state diagramming. Students were unanimous in reporting that they only felt proficient in skills which were taught in the beginning sections of the program, regardless of the version that they had completed. For skills in which more practice was needed, students almost universally listed only those skills that related to more complex diagramming, including parallel state sets, Z pulses, and decision diamonds.

Another way of measuring satisfaction with the program was to determine whether students would choose to learn state notation with the program if they could do it again, and whether they would use the program as a reference. In question 10, no students excluded the program as a method of learning state notation, although 36% of students who had Version I and 57% of students who had Version II reported that they did not know what method they would choose, since they had not read the professor's manual about state notation. In question 11, most students appropriately reported that they did not have enough experience with the professor's manual to determine which document they would use as a reference. However, only one student from either group excluded the program from future consideration for use as a reference. A complete summary of the data from the questionnaire can be found in Appendix E.

Psychology 151, Winter 1987

All students who completed Psychology 151 in Winter 1987 completed a questionnaire at the end of the semester, as summarized in Appendix H. Twenty-three students who had the program completed a questionnaire, while 33 students who did not have the program completed the questionnaire. The purpose of the questionnaire was to ascertain students' opinion of their state notation abilities, since state notation had been taught during the semester. As reported in Table 5, students who had completed the program were more likely to say that they could do a variety of state notation tasks fairly well or very well than students who had not completed the program. Question 1 asked students whether they could draw a state diagram if given the name of a procedure; a skill that was not practiced in the program but tested for in class. Ninety-one percent of students who had completed the program chose "fairly well" or "very well," while 58% of the students who did not have the program chose these categories. In question two, students who had the program were slightly more likely to report that if given a description of a procedure, they could draw a diagram fairly well or very well. Ninety-two percent of students who had the program marked this item in these two categories as compared to 85% of students who had not completed the program. Fifty-seven percent of students who had the program marked the "very well" category, as compared to 39% of students who did not have the program. Question three yielded similar results, with slightly more students who had completed the program reporting that they could explain a state diagram fairly well or very well. Notably, more than twice as many students who had completed the program marked the "very well" category as those students who had not completed the program. Question four asked students to report whether they could look at a state diagram and tell whether it was drawn correctly, a skill which was not explicitly taught in either the program or in class. Very little difference was noted in the perceived skill of either group of students. Data from these four questions indicate that students who had completed the state notation program felt more confident about their abilities on two types of skills; those that were explicitly taught either in the program or in class. However, students who had the program did not feel more confident about their abilities on skills that were not explicitly taught, either in the program or in class. These data suggest that completing the state notation program before the material was introduced in class allowed the students to better acquire related state notation skills, and to refine skills that were taught in the program. These data also suggest that completing the state notation program probably did not result in acquiring related skills in an incidental manner.

Table 5

Perceived Knowledge of State Notation As Reported by
Students in Psychology 151, Winter 1987

1. If given the name of a schedule of reinforcement or a procedure assigned from Concepts and Principles, I could draw the state diagram.

| | 1<br>Not at all | 2<br>Poorly | 3<br>Somewhat | 4<br>Fairly Well | 5<br>Very Well |
|---|---|---|---|---|---|
| Had program | 0% | 0% | 9% | 65% | 26% |
| Didn't have | 3% | 9% | 30% | 49% | 9% |

2. If given a description of a procedure that I had studied in class, such as "The first response made after 1 minute receives 3" of grain, but any response made before 1 minute resets the 1 minute timer", I could draw the diagram.

| | 1<br>Not at all | 2<br>Poorly | 3<br>Somewhat | 4<br>Fairly Well | 5<br>Very Well |
|---|---|---|---|---|---|
| Had program | 0% | 0% | 9% | 35% | 57% |
| Didn't have | 3% | 3% | 9% | 46% | 39% |

3. I could look at a state diagram without being told the name of the procedure and be able to explain the procedure accurately. The procedure would be similar to, but not exactly like ones studied in class.

| | 1<br>Not at all | 2<br>Poorly | 3<br>Somewhat | 4<br>Fairly Well | 5<br>Very Well |
|---|---|---|---|---|---|
| Had program | 0% | 4% | 17% | 35% | 44% |
| Didn't have | 0% | 0% | 27% | 52% | 21% |

Table 5--Continued

4. I could read a diagram of a procedure and then look at a state diagram of the procedure, and tell whether the diagram was drawn correctly or not. For example, I would know whether each state had the correct inputs and outputs, whether resets and transitions were used correctly, etc. The procedure would be similar to, but not exactly like ones I had studied in class.

| | 1<br>Not at all | 2<br>Poorly | 3<br>Somewhat | 4<br>Fairly Well | 5<br>Very Well |
|---|---|---|---|---|---|
| Had program | 0% | 4% | 13% | 48% | 35% |
| Didn't have | 3% | 1% | 24% | 39% | 33% |

In question II, students from both groups generally reported that they felt proficient in the same types of state notation skills. Students from both groups also generally reported that they needed more practice in the same types of skills, with one exception. Students who had the program reported that they needed more practice in complex procedures, Z pulses, parallel state sets, and decision diamonds, while no student who did not have the program mentioned these skills. These type of procedures were not greatly emphasized in class.

At the end of the semester, students who completed the state notation program were more likely to report in Question 9 that they expect to use state notation in the future than those who had not completed the program. Table 6 indicates that at the end of the semester, fifty-two percent of students who completed the program reported that they would use state notation in the future, as compared to thirty-three percent of students who had not completed the program. However, among students who had completed the program, their anticipation of using state notation in the future decreased between the time that the program was completed and the end of the semester. Seventy-nine percent of those students reported that they anticipated using state notation in the future immediately after completion, and this figure dropped to 52% by the end of the semester.

Table 6

Responses of Students in Psychology 151, Winter 1987 When Asked if They Anticipate Using State Notation in the Future

| | Yes | No | Don't Know | No Response |
|---|---|---|---|---|
| Had Program | | | | |
| Immediately After | 79% | 0% | 21% | 0% |
| End of Semester | 52% | 13% | 35% | 0% |
| Didn't Have Program | | | | |
| End of Semester | 33% | 39% | 15% | 13% |

In question IV, students were asked to choose the method or combination of methods that they would choose if they wanted to learn state notation. These choices were: (a) taking Psychology 151 just as they did, which included lectures, the professor's writings, objectives, and exams over state notation, (b) the professor's lectures only, (c) reading the professor's writings on state notation, and (d) the state notation program. Thirty percent of students who did not have the program reported that they would take Psychology 151 only, while none of the students who had completed the program chose this alternative. Most students who had completed the program included the program in combination with other alternatives, while students who had not completed the program were much less likely to include it in their combination of choices. A small percentage of students who did not have the program included it in their choices. As can be seen in Appendix H, some students who did not have the program chose every combination in which the program was included, except for one, in which lectures and program were combined. Only one student who had the program (4%) reported a preferred combination of choices that did not include the program.

Psychology 151, Winter 1988

The questionnaire given to the Psychology 151 class in Winter 1988 afforded the opportunity to discover whether the revisions made in the program since the two-group comparison resulted in more favorable comments than were obtained from the Winter 1987 students. Although the questions in the two questionnaires were not identically worded, they probed for the same types of information. Additionally, since the program was part of a unit of study on state notation, students also had the opportunity to indicate their satisfaction with the program used in this way. Many of the responses from the Winter, 1988 class reflected greater satisfaction with the program than was expressed by students in the two-group comparison. Satisfaction was measured by comparing five items on the questionnaires provided to each group of students. In particular, the class of 1988 reported an increased level of satisfaction with the clarity of instructions, their belief that all important concepts about state notation were covered, and that they knew how to answer most of the questions on the first try without looking at the answers. These data are displayed in Table 7. The class of 1988 did not report an increased belief that all material in the program was necessary. More practice items were included in the Winter 1988 version, and some students mentioned that the amount of practice that was included was unnecessary. Students in the Winter 1988 class were also less likely to report that the material was interesting.

Table 7

Comparison of Percent Satisfied With Features of the State Notation Program by Students in the Two-Group Comparison and Students in Psychology 151, Winter 1988

| | Winter, 1987 | | Winter, 1988 |
|---|---|---|---|
| | Version I | Version II | |
| Instructions easy/generally easy to understand | 50% | 50% | 96% |
| All material was necessary for learning | 100% | 86% | 87% |
| All important concepts covered | 21% | 28% | 78% |
| Could answer most questions on first try | 14% | 43% | 82% |
| Presentation of material was interesting | 86% | 93% | 70% |

Students in the Winter 1988 class were less likely to report that they would use state notation in the future than any other group of students previously polled, as depicted in Table 8. Of those students that answered "Yes," twelve said that they expected to use state notation in other psychology classes, five said that they expected to use state notation in research, and two said that they might use state notation in applied settings.

Table 8

Responses of Students When Asked if They Anticipate Using State Notation in the Future

| | Yes | No | Don't Know | No Response |
|---|---|---|---|---|
| Two-Group Comparison, Winter 1987 | | | | |
| Immediately After | 79% | 0% | 21% | 0% |
| End of Semester | 52% | 13% | 35% | 0% |
| Didn't Have Program, Winter 1987 | 33% | 39% | 15% | 13% |
| Winter 1988 | 29% | 25% | 36% | 10% |

In Question 10, most students reported that they would recommend the state notation program to a friend, either alone or in combination with other methods, if that friend wished to learn state notation. The majority of students, 48%, chose a combination of the program, the professor's writings on state notation, and the professor's lectures on state notation. The second largest percentage, 28%, chose the state notation program by itself. Twelve percent of students excluded the state notation program from their choices, leaving 88% of students to recommend the program as an effective method of learning state notation. In Question 11, seventy-four percent of students reported that they would use the state notation program to refresh their memory about state notation in the future, either alone or in combination with the professor's writings on state notation. Fifty-five percent said that they would choose only the state notation program, and 19% said that they would choose both the program and the professor's writings on state notation. Twenty-four percent reported that they would choose the professor's writings only.

## CHAPTER IV

## DISCUSSION

This study had the primary objective of creating a program of instruction to teach the basics of state notation, so that a student with no prior knowledge of state notation or schedules of reinforcement could independently work through the program and master the material in a reasonable amount of time and effort. A secondary objective of this study was to test Gilbert's (1962a) assertion that a program designed with steps containing the largest possible operant spans would be more effective than a program with steps containing smaller operant spans. This assertion was examined by comparing two subsequent versions of the program, the second of which was an effort to refine the first version and to approach the goal of creating a program with the largest operant spans possible. These research objectives will be discussed first, followed by a discussion of the methodological weaknesses of the study and suggestions for future improvement. Suggestions for future research related to the study's research objectives will be proposed. Additionally, suggestions for improvement of the program itself will be discussed.

### Success of the Program

This study has not demonstrated that by itself, the program of instruction can result in mastery of the subject matter. Only three of 28 students scored at a mastery level on the posttest after working either Version I or Version II of the program, as defined by a 90% or greater score on the posttest. The most recent version of the program as appears in Appendix K, has not been independently tested with students to determine whether it can produce mastery learning in the majority of students who complete it.

The program has only been field tested in conjunction with lecture and supplementary material on state notation in a psychology class. While students and faculty report satisfaction with the program as used in this way, and the present version represents the fifth revision of the original program, the effectiveness of the program as an independent learning tool has yet to be demonstrated. In order to test whether the program can be used as an independent learning tool, the latest version of the program should be administered to a random sample of students who have had no training or exposure to state notation.

## Comparison of Versions I and II

According to Gilbert (1962a), a subject matter programmed according to the largest operant spans will result in the most effective learning. With each successive revision of the program, an attempt was made to approximate the largest operant span as much as possible. Version I was written before Version II, and Version II was a substantial modification of Version I in an attempt to condense the material into larger operant spans. A number of statistical comparisons were made between the two versions, and these comparisons generally failed to indicate that Version II resulted in greater learning than Version I. A trend was evident in favor of Version II, suggesting that Version II could have resulted in greater learning than Version I. Version II resulted in higher means for three of five posttest questions, and when means were examined for questions two through five combined, Version II resulted in more learning than Version I. Undergraduate students scored higher on Version II than on Version I, although this trend was reversed for graduate students. As previously mentioned, Version II resulted in higher scores on some specific state notation skills as measured by each individual item as it was graded on the posttest. Version II was not more efficient in terms of time spent completing the program, however, as students spent more time working Version II than Version I. There was a high degree of variability in posttest scores, and this variability could be obscuring differences in the versions that might exist. This variability can particularly be seen in the wide distribution of posttest scores, as illustrated in Figure 1.

Several problems exist with the assumption that Version II was, in fact, a closer approximation to larger operant spans. One is that the notion of the operant span is not sufficiently precise to allow comparison of the programs in this way. Gilbert's (1962a) definition of the operant span is "the largest gain towards mastery that can be produced in a single exercise" (p. 24), and an exercise is defined as "all the material designed to establish a single new operant in the chain of mastery" (p. 24). Gilbert (1962a) discusses the arbitrary nature of defining a behavior element, and recommends that "the properties defining a behavior element are arbitrarily set by the behavior scientist to be consistent with his aims" (p. 24). A major problem with defining the operant span in this way is the subjectiveness of the definition. Tasks vary greatly in nature, and size of operant spans will differ from one task to another. The major procedure used to develop a program and determine the operant span of a subject matter is the testing procedure (Brethower et al., 1964). This method is sufficiently imprecise to result in a great deal of variation and depends on the skill of the programmer in analyzing the task and using data obtained from learner performance to modify subsequent versions of the program. A future research objective might be to operationalize the method by which the instructional designer determines the operant span. As Gilbert suggests, the operant span may vary with each different task, but the instructional designer could benefit from a set of rules or algorithms which can guide successful instructional design.

A meaningful comparison of Versions I and II requires that other features of the program must be held constant so that only the size of the operant spans vary. A confound in the comparison between Versions I and II was the presumed increase in the skill of the programmer between development of the first and second versions. The second version contained improvements which were the result of the experience of writing the first version, and these improvements were in addition to the fact that the operant spans were greater. The second version contained additional practice items, particularly in sections two and three. Version II also contained job aids which were not found in Version I. Additionally, without an operational definition of the operant span, there was no way to empirically test whether an increase in step size was present, or to determine the size of the increase.

Future attempts to study whether larger operant spans result in more efficient learning would be more effective if several conditions were different. First, the amount of material presented to learners could be greatly decreased, so that all other variables involved in programming instruction can be held as constant as possible. For example, a task or problem could be chosen which is less complex than state notation. Second, group size could be increased and characteristics of the group should be kept more homogenous than the group used in this study. Within-group variation was large on many of the comparisons used in this study. This variability may have been due to several factors. A large variation in subject characteristics existed as measured by the pre-questionnaire regarding prior exposure to topics that are relevant to state notation. A comparison of mean posttest scores indicated that graduate students scored significantly higher than undergraduates. Also, a significant positive correlation was noted between scores on the posttest and total points accumulated by the end of the semester, possibly indicating that students with greater academic abilities are better able to benefit from the program. Third, two versions of the state notation program could be written which are more obviously different from one another than Version I and Version II. One version could be similar to the present version of the program, which contains a large number of examples and practice items. The second version could be a condensed version, containing only rules for diagramming and a small subset of examples. This comparison should also include several generalization tests, in which examples with varying features are presented after completion of one version or the other of the program. The success of both programs could be measured by how well learners do on these generalization items, depending on the version of the program used. Fourth, more than two versions of the program could be tested. Since the current program represents the fifth revision, all five revisions could have been tested and the resulting posttest data examined for trends. If a trend was evident toward greater learning as measured by increasingly higher posttest scores for successive versions of the program, then a conclusion could be drawn that successive versions of the program were better approximating the largest possible operant spans.

## Design Process

The successful development of a program of instruction depends on the programmer's ability to skillfully execute the sequence of events suggested by Brethower et al. (1964) and by Gilbert (1962a). The design steps suggested by these authors can be thought of as guidelines to successful program design rather than specific steps to follow. Three stages in the design process for which operationalizing of the design process could be considered are: (1) the consulting of subject matter experts, (2) the initial stages of program development, and (3) revision of the program after field testing.

Consulting Subject Matter Experts

Some broad guidelines concerning the consulting of subject matter experts are provided (Gilbert, 1962a,b; Brethower et al., 1964; Markle, 1978; Markle and Tiemann, 1970; Resnick et al., 1973; Tiemann & Markle, 1978). As with other design steps, the skill of the designer appears to be critical in the resulting success from this process. In this study, two experts were consulted and given a sample problem to work. They talked through the problem as they worked at the request of the programmer, and were asked to complete a checklist reflecting whether they made certain decisions as they diagrammed. The automaticity of their responses seemed evident, so much so that they could not identify whether they emitted certain steps on the checklist. Both expert diagrammers stated that their thought process did not generally correspond to the list of decisions that was provided for them to check, and that the thought processes of these experts were largely inaccessible. The programmer was left with the task of analyzing what the experts did and determining how to teach these skills to learners. The eventual success of the program would depend on the programmer's skills in accomplishing these goals.

From this consultation with experts, a general conclusion was made that if expert behavior is truly automatic, stating rules about diagramming would not be taught as a terminal goal, but rather as an initial step to aid the learner. The objective of programming was to provide enough practice so that the basics of diagramming would become automatic. Comments from learners who had completed the program indicated that they felt confident in the basics of diagramming. More complex diagramming skills such as editing one's own work was also a part of expert diagramming behavior. Learner performance tended to suggest that most learners were not easily able to look at their own work and identify whether the diagram was drawn correctly or not.

Initial Stages of Program Development

The initial stages of program development, perhaps, depend most heavily on the skill of the instructional designer. The designer must determine the type of teaching frames and the amount of teaching frames to present during initial tryout. The designer must make on-the-spot decisions about how to answer learners' questions, when to provide assistance, and when to encourage learners to continue trying a difficult item. Brethower et al. (1964) provide some basic suggestions for what the designer should say when interacting with learners during the tryout phase. Future research could compare various methods of interacting with learners during the tryout phase, as well as determining how much material and what type of material to present during the tryout phase.

Continued Revision of the Program

After each successive tryout of a program, revisions are made based on learner performance. The state notation program was revised largely on the basis of comments made by learners on questionnaire data, and by global measures such as posttest scores on individual questions. Further revisions of the program could be made by a closer look at each individual response made, using a mastery criterion of 90%. Sections five and six of the current program, which teach parallel state sets, have been particularly troublesome and could possibly benefit from such an analysis.

## Programming For Problem Solving Tasks

State diagramming is a task in which a number of decisions must be made at various points during the diagramming process. Some of the decisions must be made for each diagram, while others are made only occasionally. No fixed sequence of events can be followed, and each individual may follow a slightly different sequence of events and still produce a correct diagram. For these reasons, state diagramming may be characterized as a problem solving task. As previously discussed, no specific programming method exists specifically for problem solving tasks. Based on measures of student satisfaction with the program, mathetics, or lean programming, in conjunction with the use of job aids, appears to be an appropriate method for programming state notation.

Several aspects of programming for problem solving tasks should be investigated further. The use of job aids, or algorithms, is suggested by several authors (Davies, 1967, 1972; Gilbert, 1962b; Gilbert, 1978; Lewis & Horabin, 1977; Tiemann, 1977; Tiemann & Markle, 1978). The literature contains examples of a wide variety of job aids which the instructional designer can adapt to suit the learning task being programmed. Version II of the state notation program contained several job aids. One was in the form of a list of rules which the learner was to refer to, and exercises were structured so that the learner was required to consult the list of rules before making a response. A flowchart in section five of Version II was also used to assist the learner in making decisions in knowing whether a parallel state set was needed, and if so, would the state sets interact, and how. This flowchart was discontinued in later versions because learners indicated that it was confusing. Learners were instead provided with various examples of parallel state sets, and received examples of parallel state sets on the posttest that had been taught in the program. Further research concerning programming for complex subject matters such as state notation could investigate the types of job aids that are most effective. Learners could be provided with two types of programmed materials for parallel state sets: (1) a version which teaches by example and practice only (as in the current version) and (2) a version which teaches by providing a job aid which contains all of the possible decisions that could be made when diagramming parallel state sets. Learners could be tested on their ability to construct a variety of parallel state sets not specifically taught in the program.

Another question which was not addressed in the present study was the possibility that varying the sequence of teaching the components of the state notation task may have resulted in varying degrees of learning. The sequence of components was taught using an analysis of simple to complex tasks; reading state diagrams, diagramming simple to complex single state sets, and diagramming parallel state sets. Instead, learners may be exposed to varying sequences of the state notation task. Depending on the sequence of presentation, learners may be able to acquire some components of state notation without receiving specific instruction in those components.

APPENDICES

Appendix A

Initial Stages of Program Design

## Analysis Of the Behavior of Experts

An analysis of the behavior of two experienced state diagrammers was one of the first steps of program development. Both experts were given a verbal description of a behavioral procedure which they were to diagram, and were then given two checklists which contained potential decisions that could have been made during diagramming. The experts were to indicate whether they actually made those decisions while diagramming, and whether the conditions listed for each decision which would lead one to make the decision were plausible. The procedure which was given to the two experts to diagram (MULT VI 20" NON R > v30"; from Michael, 1986), was chosen because it contained many of the major features of state notation, and was felt to be representative of one of the more complex procedures that learners would be able to diagram after completing the program. The decisions which the experts evaluated were analogous to criterion frames which would be included in the actual program. Brethower et al. (1964) recommend that subject matter experts check criterion frames to determine whether they are accurate and relevant to the objective being taught.

Results of both performances yielded several common strategies. Both experts reported that they did not recall making most of the decisions listed. Rather, as one expert explained, his behavior of diagramming was controlled by the specific words in the diagram. The behavior of consulting a list of decisions or even thinking about what decision to make was unnecessary. For both experts, the editing process was ongoing as they drew the diagram. The editing process consisted of verbalizing about the correspondence of the diagram to the verbal description, and reading the sequence of events aloud while following the path of the diagram. Statements such as "If I draw this, how will it affect this?" were typical. Both experts edited their finished diagrams carefully when completed, as well.

One major difference in strategy involved the attempt of the diagrammer to "understand" the entire procedure before constructing the diagram. The first expert read the description quickly, and then began drawing the first state specified in the description. His diagramming appeared to be guided by the sequence of events as specified in the description. When asked to describe his strategy, he explained that he begins diagramming by drawing one state, and then continues based on the event that will lead from that state into the next state. He occasionally paused to read certain aspects of the description carefully, perhaps when some aspect of the diagram did not seem clear, or at a point when a major feature of the diagram had to be drawn, such as the interaction between parallel state sets. The other second diagrammer spent much more time reading and verbalizing about features of the procedure, and in making notes about the procedure before he began the diagram. He read the entire procedure three times, and described the entire procedure aloud before beginning to diagram. Since both diagrammers were able to produce a correct diagram, the benefits of one strategy over another do not seem obvious. However, for the beginning diagrammer, perhaps reading the verbal description thoroughly and making notes before diagramming, as the second expert did, may help.

The protocol used with the second expert performer is presented; this protocol includes all of the major portions used with the first performer, but is slightly modified based on the comments of the first performer.

Diagram 1

I would like to identify the decisions that you are making as you draw this diagram. Please talk aloud so that I can get an idea of the sequence of steps that you take as you compose the diagram.

---

Assume a food deprived rat who will receive a food pellet each time "on SR" is in effect.

Two timers are simultaneously in effect: A variable 20" timer, and a tone and a variable 30" timer. Until the variable 30" timer times out the rat is on a variable interval 20" food reinforcement schedule and the tone is on. Depending on the variable 20" timer the rat may receive none, one, or more than one reinforcement for lever pressing in the tone. When the variable 30" timer times out the tone is turned off and a condition becomes in effect in which reinforcement is not available. Meanwhile, another 30" timer is timing. If a response occurs during this period it resets the timer back to the beginning of the interval (v30") that is being timed. When a 30" interval has elapsed without any responding the tone is turned on and the variable interval 20" schedule of food reinforcement is in effect again (Note: if the tone is turned off at the moment when the reinforcement condition has been met, an additional response is required).

Now look at the checklist below of "Possible Decisions That Can Be Made When State Diagramming." Put a checkmark next to the ones that you made while constructing this diagram, and put an "X" next to any that you did not make. (You may look at your diagram and notes if you wish). If any important decisions seem to be omitted from the checklist, please write them at the end of the list. Also, if you can recall the ORDER in which you made these decisions, please number the decisions.

| ORDER | DECISION |
|---|---|
| _______ | Will I need more than one state set? |
| _______ | Will the state sets interact? |
| _______ | If state sets will interact, will I use Z pulses? |
| _______ | Do all of my transition arrows have the correct order and number of inputs and outputs? |
| _______ | Will I use a START? |
| _______ | Do I need any resets (to same or different states?) |
| _______ | Do I wish to consider any alternative methods of drawing this diagram? |
| _______ | Do I need a decision diamond? |
| _______ | Do I use ON SR, op feeder, or op dipper? |

Here is an expanded version of the checklist. Each decision has conditions listed under it which might possibly lead the diagrammer to make the decision. Put a checkmark next to the conditions which seem appropriate to the decision, and put an X next to any conditions which do not seem appropriate to making the decision. Please write in any other conditions which seem relevant to each decision.

---

DECISION

Will I need more than one state set?

- Time or response limit on session?
- Two or more conditions occurring simultaneously?
- Are the demands of the description such that it would be impossible to diagram with only one state set?
- Would more than one state set clarify the procedure?

Will the state sets interact?

- Is a procedure other than concurrent schedules specified?

- Does at least one condition depend upon the status of another condition?

Will I use Z pulses?

- Will session end after N minutes or N responses?

- Are there 2 or more contingencies or conditions that must both be met?

- Is there more than 1 response that can be made with a different consequence for each?

- Does an extinction condition vary with an SR condition?

Do my transition arrows have the correct order and number of inputs and outputs?

- Is there one and only one input for each transition arrow?

Will I use a START?

- Does a stimulus condition exist which must be in effect at the beginning of the session?

- Is synchronicity between two state sets required?

Do I need any resets?

- A NON R or IRT>T condition?

- If the organism emits a certain response, must either a timer or counter begin again?

Do I wish to consider any alternative methods of drawing the diagram?

- After the diagram is complete, does it seem unduly complex?

- Would the diagram correspond more closely to the verbal description if drawn another way?

- Would the procedure seem easier to understand if the diagram were modified slightly?

Do I need a decision diamond?

- Was a probability stated?
- Is one of two alternatives possible?
- Do two states need to interact, but Z pulses cannot be used for some reason?
- With parallel state sets, does a transition in one state set depend upon which state is active in the other state set?

Do I use ON SR? op feeder? op dipper?

- Do I note the type of organism I am diagramming for and look for a description within the diagram to indicate which I will use?

## General Objectives

The analysis of the performance of the two expert state diagrammers indicated that a learner who completes a program of instruction designed to teach state notation would, upon completion of the program, meet the following objectives:

1. When given a state diagram including any combination of the symbols for inputs and outputs, the learner will correctly describe the behavioral procedure depicted in the diagram.
2. When given a verbal description of a behavioral procedure of moderate complexity (in general, most of the diagrams in Michael's manual do not exceed moderate complexity), the learner will be able to produce the correct diagram.
3. The learner will edit diagrams, which consists of one or more of the following behaviors:
   (a) identifying features which are used incorrectly;
   (b) re-draw diagrams which contain one or more features used incorrectly;
   (c) re-draw diagrams which are not technically incorrect, but for which the procedure would be clarified if drawn differently.

## Program Steps and Criterion Frames

At this stage of program development, these general objectives were expanded into the specific behaviors that comprise mastery. This stage of development is analogous to Gilbert's (1962a) writing of the prescription. In addition, the decisions which the expert state diagrammers were asked to evaluate were matched with the behaviors which are part of the prescription. This stage is analogous to Gilbert's (1962a) development of the domain theory. These two stages of program development are presented in the proposed order in which they are to be included in the first draft of the program. Some of the behaviors are starred; these are behaviors which are explicitly tested in criterion frames. The behaviors which are not starred will not be explicitly tested in criterion frames and may not be taught as separate steps in the program, depending upon the performance of the learners who try out the first drafts of the program. Program objectives and corresponding criterion frames follow.

Criterion Frames

---

1. The learner will verbalize the proper sequence of events in a state diagram by:
   (a) correctly verbalizing the instantaneous nature of transitions from one state to another when presented with a simple diagram
 * (b) reading the events in their proper sequence (from left to right on each transition line) when presented with simple diagrams
 * (c) identifying correct and incorrect descriptions of states and transitions and rewriting the descriptions if necessary

Decision(s): (a) Beginning with each new state, am I reading from left to right?
(b) Do I say "transition to state ____" for each transition arrow?
(c) Do I read all of the events in one state before going on to the next state?

---

This step should make the benefits of state diagramming obvious. Simple schedules of reinforcement which are similar will be contrasted. The reader will be able to easily identify the critical features of each schedule from reading and comparing the state diagrams.

---

---

2. The learner will diagram inputs and outputs in their proper sequence by:
   (a) correctly identifying verbal descriptions of environmental events as either inputs or outputs
   (b) drawing the inputs and outputs correctly in a diagram
 *  (c) identifying correct and incorrect depictions of events as inputs and outputs given the verbal description and the diagram, and re-drawing the diagram if necessary

Decision(s):
(a) Does each transition line have one (and no more than one) input?
(b) Are responses and time the only two variables used as inputs?
(c) Are inputs and outputs on the same transition line separated by a colon?
(d) Are two outputs on a transition line separated by a semicolon?

===

At this point, the learner has had practice in reading and in drawing parts of simple diagrams. A step can be added here in which the learner draws some simple diagrams without prompting within the program, but by consulting a reference showing inputs and outputs, using the components learned in steps 1 and 2. Inputs and outputs will be presented in a logical order; either one symbol at a time or sets of symbols with common features.

===

---

3. The learner will diagram the appropriate designation of reinforcement/punishment delivery (on SR, op feeder, op dipper, on Sk) by:
   (a) indicating the appropriate method of diagramming reinforcement or punishment delivery given a verbal description
   (b) drawing the appropriate designation in diagrams
   * (c) indicating the appropriateness of the method of drawing reinforcement or punishment delivery given a verbal description and a diagram, and re-draw if necessary

Decision: Is the reinforcement the type which is delivered as a discrete event (i.e. food pellet, drop of water, or token), or is the reinforcement left available for a specified period of time (i.e. grain hopper, music)?

Note: Until this step in the program, the learner has only been introduced to "op feeder".

===

At this point, the learner can read and diagram simple diagrams which consist of the "main" inputs and outputs. Now, more features will be taught.

===

---

4. The learner will use the reset and transition from one state to another state by:
   (a) drawing the reset or transition portion of the diagram correctly
   * (b) indicating the correctness/incorrectness of the use of various resets and transitions and re-drawing if necessary

Decision(s): (a) Is differential reinforcement of low rate behavior specified?
(b) Is a non-response period greater than a certain period of time specified?
(c) Does an escape and/or avoidance situation exist?

===

----------------------------------------------------------------------

5. The learner will draw the appropriate pattern (for single state sets which involve some variation of a standard, linear diagram) by:
(a) matching the skeleton diagram to the procedure
(b) drawing part or all of one of the skeleton diagrams introduced, given a verbal description of the procedure
* (c) Given two diagrams of the same procedure (one correct and one incorrect, or just two valid ways to draw the same procedure), the learner will choose the correct or "clearest" diagram, and edit if necessary.

Decision(s): (a) Are there two alternative ways to get to the same state?
(b) Does the procedure repeat by returning to a specific state?
(c) Can the procedure be diagrammed in more than one way, and which would clarify the procedure best?

==========================================================

Note: The purpose of step 5 is to introduce the learner to some variations in diagramming structures which will be used in more complex diagrams. This step also introduces the learner to some of the more sophisticated skills of choosing the diagram that best represents the procedure (these skills are commonly thought of as part of the "art" of state notation - which the sophisticated user has acquired through extensive experience)

==========================================================

---

6. The learner will use the decision diamond (in the single state set) appropriately by:
(a) identifying occasions in which a decision diamond should be used, given verbal descriptions of procedures
* (b) drawing part or all of diagrams in which a decision diamond is used

Decision(s): (a) Was a probability explicitly stated in the verbal description? (i.e. "Each response has a .10 probability of being reinforced, else...")
(b) Was a probability suggested in the verbal description by describing a "roulette wheel situation?" (i.e. at a regular time interval, an electronic "roulette wheel" is spun. The wheel has "X" number of spaces and only one leads to the next response being reinforced, else...)

===

Note: These uses of the decision diamond are relatively straightforward. The decision diamond is used in several other ways, which will be introduced in objective 8.

===

---

7. The learner will use parallel state sets appropriately by
* (a) identifying occasions in which parallel state sets are and are not needed, given verbal descriptions of procedures.

Decision(s): (a) Is there a time or response limit on the session?
(b) Are there two or more conditions occurring simultaneously? (such as two timers, or two separate response requirements that must be met for reinforcement to be delivered)
(c) Do two or more conditions alternate? (such as a reinforcement schedule and a punishment procedure)
(d) Does the presence of one condition influence the presence or absence of another condition (i.e. response duration of a specified time period causes reinforcement to be delivered)

==========================================================

Note: At this point in the program, not all possible choices for use of parallel vs. single state sets will be covered. The learner will get more practice in objective 10.

==========================================================

---

8. The learner will identify whether parallel state sets need to interact, and will use the Z pulse or decision diamond (whichever is most appropriate) for interaction between parallel state sets by:
(a) identifying occasions in which one or the other is appropriate, given verbal descriptions of procedures
(b) drawing part or all of a diagram correctly, given a verbal description and possibly a skeleton diagram
* (c) identifying occasions in which the appropriate interaction conventions were or were not used appropriately, and re-drawing if necessary
* (d) identifying the rare occasions when two parallel state sets will NOT interact (CONC schedules without a changeover delay, a CONC punishment and reinforcement procedure (in which the response must occasionally be reinforced in order to have some behavior to punish)

Decisions: (a) Is the situation such that two alternative paths in the same state must be accounted for depending upon the condition of a state in another state set? (decision diamond used here)
(b) When a condition is met in one state set, can it be produced as an output in that state set to function as an input in another state set? (Z pulse used here)

===

--------------------------------------------------------------------

9. The learner will use a Start and Stop appropriately by
   (a) identifying occasions in which the Start and Stop is needed
   * (b) diagramming the Start and Stop correctly, given a procedure in which they are needed (this includes synchronizing the events in the procedure correctly)

Decisions: (a) Does the verbal description specify that a stimulus condition be present when the session begins?
(b) Do parallel state sets need to be synchronized?

==================================================

--------------------------------------------------------------------

10. The learner will use single or parallel state sets correctly (an extension of objective 8 - procedures which seem complex but can actually best be diagrammed with single state sets are considered in this description)
   (a) identifying whether a simple or parallel state set is most appropriate for a procedure, given the verbal description
   * (b) diagramming the procedure correctly given the verbal description

Decision: Does the procedure involve two or more separate but almost identical sequences of events? (i.e. matching to sample)

==================================================

---

11. The learner will diagram complex features of procedures which involve the following situations:
(a) A Z pulse interrupts a counter or timer; the experimenter wishes to specify whether the counter or timer will reset or continue where interrupted when the procedure resumes (these situations arise in parallel state sets)

Decision: When the Z pulse interrupts the counter or timer, does the diagrammer continue at the point at which the interruption occurred, or does the diagrammer begin the counter or timer from the beginning? This decision can be made on two bases:

(a) Does the verbal description specify what is to occur?
(b) If the verbal description does not specify what is to occur, what does the diagrammer deem appropriate, based on knowledge of behavioral procedures?

===

Note: This step is one of the most difficult (if not the most difficult) of all of the editing skills required of state diagrammers.

===

Appendix B

Sample Pages From Versions I and II

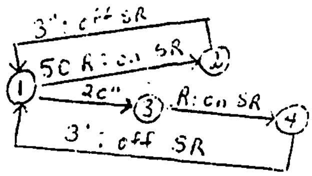
Version One - Response Page One from First Section on Parallel State Sets

There are several clever variations of diagrams which are just right for certain procedures. Here are some examples:

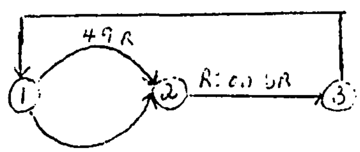
---

This procedure specifies that a response is reinforced on <u>either</u> of two contingencies; whichever is met first.
For example: The pigeon can receive 3 seconds of grain if
a) he makes 50 responses, OR
b) he makes one response after a 20" timer times out.

If we don't know the correct variation, we might try to draw it this way:




Too complicated! Now, get rid of the duplication by making the same pathway for reinforcement delivery for both alternatives. Fill in the missing parts of the diagram below:

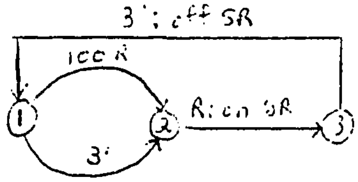



When you use the same reinforcement pathway for both alternatives, you can take the last R off and put it in the state that delivers SR.

---

The diagram below is incorrect. Redraw the diagram.

The pigeon receives three seconds of grain if 100 key pecks are made,or if a response is made after 3 minutes.

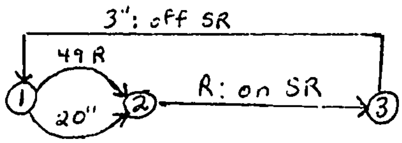


Version One - Response Page Two from First Section on
Parallel State Sets

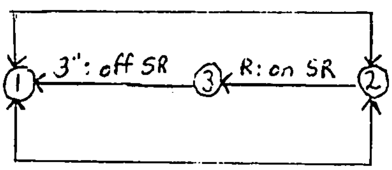
Here's the procedure again:

The pigeon can receive 3 seconds of grain if
a) he makes 50 responses, OR
b) he makes one response after a 20" timer times out.

There's nothing really wrong with this way to draw:




BUT... This way really clarifies the procedure!
Fill in the missing parts of the diagram:




You can reverse the direction of the transition arrows to lead back to state 1.

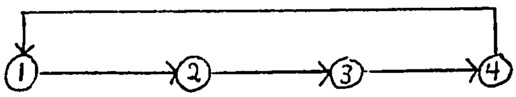
---

Diagram this procedure two ways. First, do it with the transition arrows reversed, as above. Then, do it the way you first learned it. Both are equally acceptable!

A rat can receive a food pellet if the following conditions are met: First, a 20 second timer times out. Then, either the thirty-first lever press or two lever presses made after a 30" timer times out must occur.

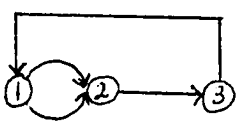
Version One - Response Page Three from First Section on Parallel State Sets

Here is a "standard" diagram. There are no resets and no alternative ways to get to the same state. (This is diagram "A")

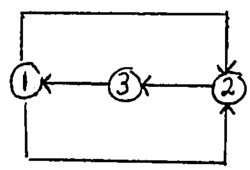



You have had variations of diagrams which look like this:

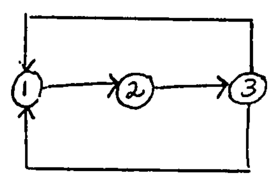
B.




C.




D.
(reset to a different state)




Match the letter A,B,C, or D to the verbal description. You may select more than one letter for each description.

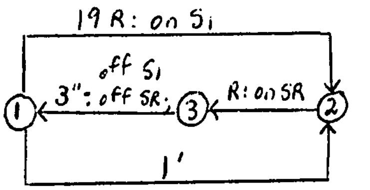
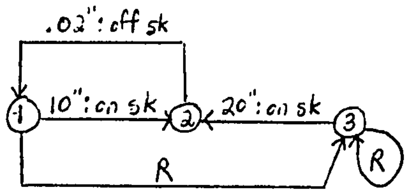
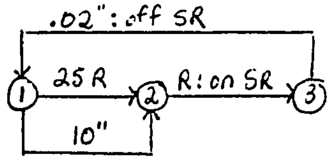
_____ The first of two response requirements to be met will be reinforced. A pigeon can make 10 responses, OR the first response to be made after 10 seconds will be reinforced with 3 seconds of grain.

_____ The pigeon must meet BOTH response requirements before reinforcement is delivered. First, 25 responses must be made. Then, the first response after a 20 second timer times out will be reinforced with 3 seconds of grain.

_____ After 5 seconds, a red light comes on. Then after 3 seconds, shock is turned on. A response will escape the shock and reset the 5 second timer. If no response is made, shock will stay on for 2 seconds and the 5 second timer will reset, starting the procedure over again.

Version One - Criterion Frame for First Section on Parallel State Sets

For each description, check the way that the diagram is drawn. If there are any errors or if the diagram could be drawn more clearly, redraw the diagram.

| Procedure | Diagram | Redraw if needed |
|---|---|---|
| Both response requirements must be met. First, the pigeon must make 20 responses and a green light will turn on. Next, the first response to be made after 1 minute will be reinforced with 3 seconds of grain. The green light is turned off and the procedure begins again. |   | |
| The procedure begins with a 10 second timer, after which a brief pulse of shock is delivered. The organism can avoid the shock onset by making a response, which puts him in a "safe" state. Once he is in the safe state, a 20 second timer will turn on a brief pulse of shock. However, once the organism is in the safe state, a response before the 20 second timer times out allows him to remain in that state indefinitely. |   | |
| Either 25 lever presses or the first response after 10 seconds will be reinforced with a food pellet. |   | |

Version Two - Introduction to Unit on Parallel State Sets

In the future when you are given a procedure to diagram, you will have several decisions to make. Just read over the following to get the general idea:

1. First, you will have to decide whether the procedure requires a single OR a parallel state set.

2. If a parallel state set is needed,then you will have to decide whether they must interact (something that happens in one state set affects the other state set). The one that you saw on the preceding page did not interact. However, many parallel state sets DO interact.

3. If the state sets DO need to interact,there are two main ways that they can do so. There are two methods called Z pulses and decision diamonds by which parallel state sets interact.You will have to decide by which method they will interact.

---

This may sound like a lot to learn, but when you find out how to make these decisions, it'll be easier! Now turn the page.

## Version Two - Flowchart, Parallel State Sets

Here is a flowchart that can help you make these decisions. You may want to pull this page out of the program and keep it in front of you to use as an aid in diagramming.

1. Do I need a parallel state set? NO - - > Use a single state set

   A. Are there two timers (or schedules) which are timing simultaneously?

   B. Should the response be broken down into two separate components, such as "sitting down" and "standing up?"

   C. Do I need to keep track of time or responses (i.e. to stop the session) while a procedure is running?

|
| YES
|

2. Do the state sets need to interact? NO - -> Just draw them separately

   A. In one state set, a decision has to be made depending on what's happening in another state set.

   B. In one state set, something will happen when a response or time requirement is met in another state set.

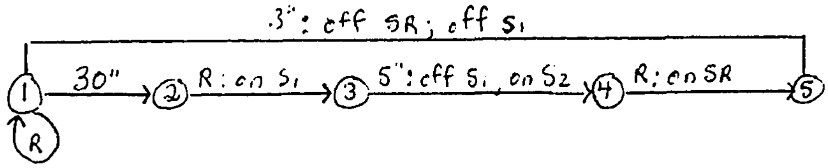
|
| YES
|

3. Use a decision diamond if the answer to #2 above was A.

   Use a z pulse if the answer to #2 above was B.

## Version Two - First Student Response Page on Parallel State Sets

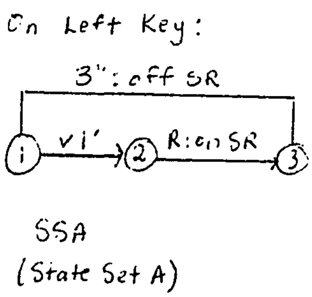
Until now, all of the state diagrams you have been introduced to were single state sets such as this:

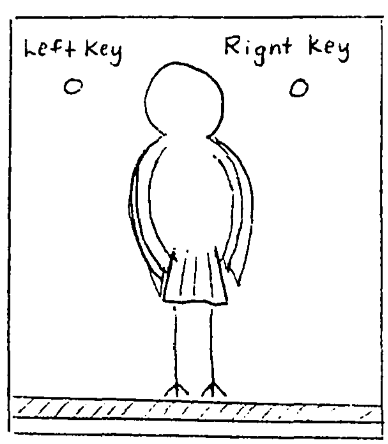


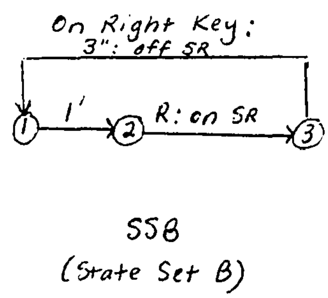

Procedures are often best represented by parallel state sets (two or more state sets, such as in the example below. Label each state set "State set A" (SSA). "State set B" (SSB), etc.

Imagine an operant chamber that looks like this. There are two keys, and each key is programmed with a different procedure. The pigeon can work on either procedure and can switch back and forth between procedures. These procedures are two types of schedules of reinforcement.

---




---

With parallel state sets, the organism is always in one of the states in EACH state set. Suppose that if we looked into the chamber at one instant in time, in which:

a) In SSA the v1' timer is timing and 30" have elapsed;
b) In SSB the grain hopper has just been raised.

Put an arrow above the state in each state set to show where the procedure is at this instant.

---

Appendix C

Undergraduate and Graduate Student Questionnaires

Questionnaire - Undergraduates

1. What is your major?

What is your minor?

What year are you in college? (Freshman, sophomore, etc.)

2. Put a checkmark next to all courses that you have had at WMU:

---- Psychology 194

---- Psychology 151 (If currently enrolled, say so)

---- Any other psychology courses at WMU? If so, please name:

---- Any courses in computer programming at WMU?
(Write the names here:)

---- Any courses in math at WMU?
(Write the names here:)

3. Have you ever had training in state notation?
If so, please describe.

4. Have you had any COLLEGE courses BESIDES 151 (at WMU or anywhere else) which covered the following: If so, write where, and name of class:

---- Schedules of reinforcement (Fixed and variable ratio and interval)

---- Experimental research with NON-humans (i.e. rats and pigeons)

---- Operant conditioning

---- Respondent conditioning

---- Behavior modification with humans

---- Flowcharting

---- Formal logic (briefly describe)

---- Lab work in operant behavior (i.e. rat lab) -
Briefly describe:

5. Did you have any courses in HIGH SCHOOL in:

---- Algebra

---- Geometry

---- Trigonometry

---- Other math courses

---- Computer science

---- Courses which taught formal logic

---- Psychology

---- Rat lab in operant behavior

---- Any experience (at any time) in diagramming electronic systems?

6. ---- Have you ever been involved in operant research with NON-humans in a laboratory setting? If so, please explain what you did.

---- Any experience with SKED? Please explain.

---------------------------------------------------------------------------------

Please make a few brief comments about the state notation program:

1. Were the instructions and explanations in the program easy or difficult to follow?

2. Did you already know any of the concepts being taught in the program? If so, which ones?

3. Did any of the material seem unnecessary? (Do you feel that you could have learned the same amount even if some of the material were cut out? If so, what parts?)

4. Were any important concepts about state notation omitted from the program, or not covered in as much detail as they should have been?

5. Did you feel that you knew how to answer MOST of the items, or did you have to look at the answers before you understood the items?

6. Was the presentation of the material interesting/ could have been made more interesting/ dull? Please explain.

7. Did you find the job aids helpful? Did you use them?

8. In what features of state notation do you feel that you are proficient?

   Need more practice?

9. Do you ever expect to use state notation outside of Psychology 151? If so, how?

10. If you wanted to learn state notation and you could pick any of the following methods, which would you choose and why?

---- Take Psychology 151 with Dr. Michael

---- Read Dr. Michael's manual on state notation on my own

---- Do Esther's program by itself

---- Any combination of the above? Explain.

---- Don't know - I haven't read Dr. Michael's manual yet

11. Which would be more helpful as a REFERENCE about state notation and why?

---- Dr. Michael's manual

---- Esther's program

---- Not enough experience with Dr. Michael's manual to judge

Questionnaire - Graduate Students

1. Write the name of the program that you are in (Ex.: Master's ABA)

2. Put a checkmark next to all the courses that you have had at WMU:

---- Psychology 610

---- Psychology 151/510

---- Rat lab for Psychology 151

---- Psychology 611

---- Psychology 516 (Dave Lyon's course)

---- Any other courses in experimental analysis of behavior at WMU?
(Write the names here:)

---- Any courses in computer programming at WMU?
(Write the names here:)

---- Any courses in math at WMU?
(Write the names here:)

---- List all the courses that you have had at WMU that have taught state notation and briefly describe the level of proficiency in state notation that you attained as a result:

---- Any other training in state notation (at WMU or somewhere else)?
Please describe.

3. Have you had any COLLEGE courses (anywhere besides WMU) which covered the following:

---- Schedules of reinforcement

---- Single-subject research with NON-humans

---- Single-subject research with humans

---- Flowcharting

---- Formal logic (briefly describe)

---- Math (briefly list topics covered)

---- Computer science (briefly describe)

---- Lab work in operant behavior (i.e. rat lab) -
Briefly describe:

4. Did you have any courses in HIGH SCHOOL in:

---- Algebra

---- Geometry

---- Trigonometry

---- Other math courses

---- Computer science

---- Courses which taught formal logic

---- Psychology

---- Rat lab in operant behavior

---- Any experience (at any time) in diagramming electronic systems?

5. Check all the ways in which you have ever been involved in operant research with NON-humans in a laboratory setting: (list all types of organisms here; i.e. rats, pigeons)

---- Put animals in operant chambers; took them out

---- Graphed data

---- Designed your own research

---- Helped design research

---- Programmed relay equipment

---- Other (explain)

---- Any experience with SKED?
(If you have programmed in SKED, approximately what is the level of your knowledge of SKED?)

---- I could write an entire SKED program with no help

---- I could write a SKED program with some help

---- I could modify a few statements of someone else's program

---- Other (explain)

Do you think that a knowledge of state notation (Michael's method) would help/hinder/make no difference to a person who was learning SKED? Please explain.

---- Any other computer language used to program operant chambers?
If so, which?

Please make a few brief comments about the state notation program:

6. Were the instructions and explanations in the program easy or difficult to follow?

7. Did you already know any of the concepts being taught in the program? If so, which ones?

8. Did any of the material seem unnecessary? (Do you feel that you could have learned the same amount even if some of the material were cut out? If so, what parts?)

9. Were any important concepts about state notation omitted from the program, or not covered in as much detail as they should have been?

10. In what features of state notation do you feel that you are proficient?

Need more practice?

11. Do you ever expect to use state notation in the future? If so, how?

12. If you wanted to learn state notation and you could pick any of the following methods, which would you choose and why?

---- Take Psychology 611 with Jack Michael

---- Read Jack Michael's manual on state notation on my own

---- Do Esther's program by itself

---- Any combination of the above? Explain.

13. Which would be more helpful as a REFERENCE about state notation and why?

---- Michael's manual

---- Esther's program

Appendix D

Posttest

## Posttest

Diagram this procedure:

The procedure begins with the houselight on. After 5 seconds, a light is turned on. If R1 is made, the light is turned off and a tone comes on. Now, the rat only has 3" to make R2 and get a food pellet (and off tone). Otherwise, the procedure restarts with the 5" timer. If the rat HAS made R2 within 3 seconds, the procedure continues with timing of a 10 second timer. An R3 made before the 10 second timer times out restarts the procedure from the beginning (with the 5 second timer). If the 10 second timer times out, 3 seconds of shock are delivered and this cycle of 10 second timer followed by shock continues until R3 is made.

Session length is 30 minutes or 30 R3's - whichever comes first.

Read the procedures and draw the diagram for each.

---

The rat is reinforced with a food pellet when he holds the lever down for 10 consecutive seconds.

---

This is a 2-key chamber for a pigeon. On the left key, the pigeon is reinforced when he pecks 50 times. On the right key, the first peck after 20" passes is reinforced. The pigeon may work on either key and can switch at any time.

---

The pigeon must meet the requirements for both of these schedules (the pigeon is working on both simultaneously) before reinforcement is delivered.

a. Peck 50 times

b. Make one response after 20" times out.

---

If the child is sitting down when a v20" timer times out, a token is delivered.

Appendix E

Exit Questionnaire

## Exit Questionnaire

Please make a few brief comments about the state notation program:

1. Were the instructions and explanations in the program easy or difficult to follow?

| | Easy/Generally Easy | Somewhat Difficult | Difficult |
|---|---|---|---|
| V1 | 50% | 7% | 43% |
| V2 | 50% | 0% | 50% |

2. Did you already know any of the concepts being taught in the program? If so, which ones?

| | No | Yes | |
|---|---|---|---|
| V1 | 71% | 35% | 21%-behavior analysis concepts<br>7%-computer flowcharting<br>7%-Z pulses/SKED |
| V2 | 79% | 21% | 14%-computer flowcharting<br>7%-concepts in first 2 sections |

3. Did any of the material seem unnecessary? (Do you feel that you could have learned the same amount even if some of the material were cut out? If so, what parts?)

| | All necessary | Some unnecessary | More needed |
|---|---|---|---|
| V1 | 100% | 0% | 14% |
| V2 | 86% | 7% | 7% |

4. Were any important concepts about state notation omitted from the program, or not covered in as much detail as they should have been?

| | None omitted | Don't know | Some omitted | |
|---|---|---|---|---|
| V1 | 21% | 21% | 58% | 28% Z pulse, diamond<br>14% VR, FR<br>7% Obj 3 and beyond<br>7% Obj 6 and 7<br>7% Obj 7 |
| V2 | 28% | 35% | 37% | 14% Obj 5<br>14% Z pulse<br>7% Parallel state sets<br>7% Obj 4,5 |

5. Did you feel that you knew how to answer MOST of the items, or did you have to look at the answers before you understood the items?

| | Knew most | Looked at some | Looked at most |
|---|---|---|---|
| V1 | 14% | 7% | 79% |
| V2 | 43% | 21% | 36% |

6. Was the presentation of the material interesting/ could have been made more interesting/ dull? Please explain.

| | Interesting | Could be more interesting | Dull |
|---|---|---|---|
| V1 | 86% | 14%-more human examples | 0% |
| V2 | 93% | 7%-end was confusing | 0% |

7. Did you find the job aids helpful? Did you use them?

| | Used them/helpful | Didn't use them |
|---|---|---|
| V1 | 93% | 7% |
| V2 | 93% | 7% |

8. In what features of state notation do you feel that you are proficient?

| | | |
|---|---|---|
| V1 | 43% | Basics |
| | 21% | First 2 or 3 objectives |
| | 14% | Don't know |
| | 7% | First 4 objectives |
| | 7% | First 5 objectives |
| | 7% | Single state sets |
| | 7% | None |
| V2 | 29% | Sections 1,2,3 |
| | 21% | Single state sets |
| | 21% | Simple features |
| | 14% | Sections 1-4 |
| | 14% | Read diagram |
| | 7% | Most features |

Need more practice?

| | | |
|---|---|---|
| V1 | 43% | Z pulse |
| | 36% | Parallel state sets |
| | 36% | Decision diamond |
| | 28% | General practice |
| | 14% | Objectives 5-7 |
| | 7% | Drawing diagrams |
| | 7% | Objectives 6,7 |
| | 7% | Stop and start |
| | 7% | Section 4 |
| V2 | 43% | Z pulse |
| | 21% | Section 5 |
| | 21% | Parallel state sets |
| | 14% | General practice |
| | 14% | Sections 4,5 |
| | 7% | Decision diamond |

9. Do you ever expect to use state notation in the future?
If so, how?

| | Yes | | No | Don't know |
|---|---|---|---|---|
| V1 | 50% | Future psychology classes | 0% | 21% |
| | 43% | Experimental work | | |
| | 7% | Explain complex procedures | | |
| | 7% | Teaching | | |
| | 7% | Reading journal articles | | |
| V2 | 21% | Future psychology classes | 14% | 43% |
| | 14% | Explain complex procedures | | |
| | 7% | Computer flowcharting | | |

10. If you wanted to learn state notation and you could pick any of the following methods, which would you choose and why?

V1

<u>0%</u> Take a psychology class with Dr. Michael in which state notation is taught

<u>0%</u> Read Dr. Michael's manual on state notation on my own

<u>21%</u> Do Esther's program by itself

<u>71%</u> Any combination of the above? Explain.*

<u>36%</u> Don't know - I haven't read Dr. Michael's manual yet

* Need more practice; several methods would be better than only one

V2

<u>0%</u> Take a psychology class with Dr. Michael in which state notation is taught

<u>0%</u> Read Dr. Michael's manual on state notation on my own

<u>7%</u> Do Esther's program by itself

<u>43%</u> Any combination of the above? Explain.*

<u>57%</u> Don't know - I haven't read Dr. Michael's manual yet

*Esther's program and the lectures would be a good combination

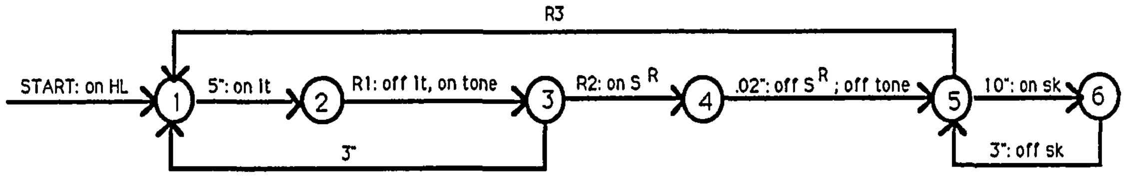
11. Which would be more helpful as a REFERENCE about state notation and why?

| | | |
|---|---|---|
| V1 | 7% | Dr. Michael's manual |
| | 14% | Esther's program |
| | 79% | Not enough experience with Dr. Michael's manual to judge |
| V2 | 0% | Dr. Michael's manual |
| | 7% | Esther's program |
| | 93% | Not enough experience with Dr. Michael's manual to judge |

Appendix F

Posttest Scoring Protocol

# Posttest Scoring Protocol




1. Do all transition lines drawn have an input? (more than 1 ok) (1)

2. Do all transition lines drawn have one and only one input? (1)

3. Are all inputs diagrammed? (even if incorrectly) (1)
   5", R1, R2, .02",* 10", 3", 3", R3
   * give credit if 3" is substituted for .02"

4. Are all outputs diagrammed? (1)
   (must be in the format of "on ___" or "off___")
   on light or on S, off light or off S, on tone or off S

5. Is START: on HL used? (slight format error is ok) (1)

6. Are transitions to a different state correct? (3) (1 for each transition)
   (even if the transition line does not go to or from the states drawn, give credit as long as the basic logic is there)


7. Do all transition lines drawn have arrows? (1)

8. Are all inputs and outputs separated by a colon? (1)

9. Do multiple outputs have semicolons? (1)

10. Are seconds (") used for all times? (1)
    (only for the times that are diagrammed)

11. Is .02" used for reinforcement? (has to be written exactly) (1)

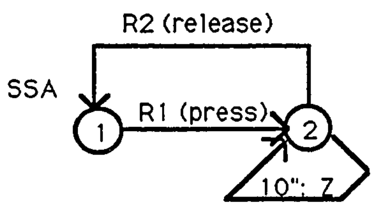
12. Is reinforcement turned on and off with 2 states, (2) (1 for each)
    Is shock turned on and off with two states?
    (slight format errors are ok)

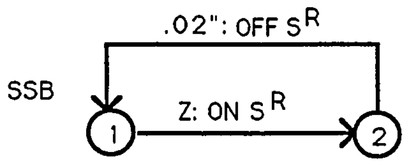
13. Is the sequence of events correct? (all or none) (1)

14. Is an attempt made to stop the session? (1)

15. Is the session stopped correctly? (Accept any logical method) (1)

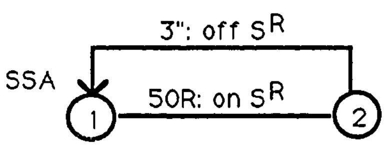
## Question 2

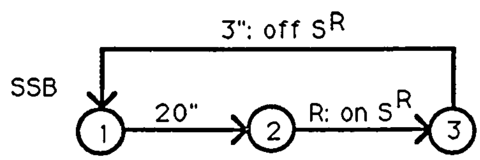



1. Are parallel state sets used? (1)
2. Are R1 and R2 designated correctly in a separate state set? (1)
   (can be written as "press" and "release", or any other format that is logical)
3. Is a Z pulse used? (1)
4. Is a Z pulse used correctly (Z in one state set used to deliver reinforcement in the other state set) (1)
5. Is .02" used for reinforcement? (1)
   (Must be written correctly)

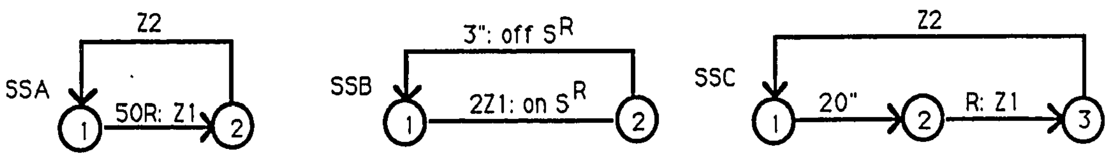
## Question 3



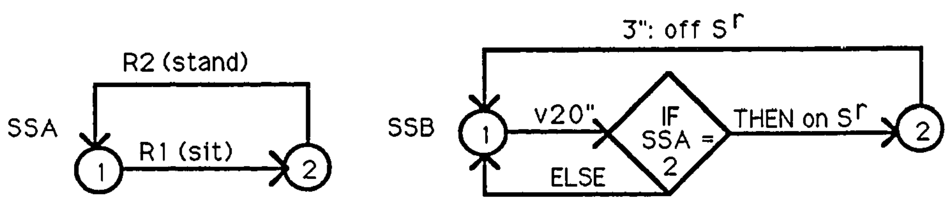

1. Are parallel state sets used? (1)
2. No interaction between state sets (no z or diamond) (1)
3. Is FR 50 drawn correctly? (1)
4. Is FI 20" drawn correctly? (1)
5. Is 3" used for reinforcement? (1)

## Question 4




1. Are parallel state sets used? (1)
2. Are 3 state sets used? (Don't have to be in same format as above answer) (1)
3. Is a Z pulse used to deliver reinforcement? (1)
4. Is a Z pulse used to return to State 1? (1)
5. Is 3" used for reinforcement? (1)
6. Is FR 50 drawn correctly? (1)
7. Is FI 20" drawn correctly? (1)

## Question 5




1. Are parallel state sets used? (1)
2. Is a decision diamond used? (slight format errors ok) (1)
3. Is a state set drawn with R1 and R2 alternating? (Format can vary) (1)
4. Is the diamond drawn with a "then" and "else" path? (slight format error OK) (1)
5. Is .02" used for reinforcement? (1)

Appendix G

Psychology 151, Winter 1987
Examination Item and Scoring Protocol

Examination Item and Scoring, Winter, 1987

Provide a state diagram of the following procedure: Read the description carefully and take time to check your work when you have finished. This procedure includes several that you are familiar with. The diagram must be drawn as a single state set.

Assume a rat in an operant chamber. There is a chain and a lever in the chamber. (Also assume that unlike most rats this rat has good color vision).

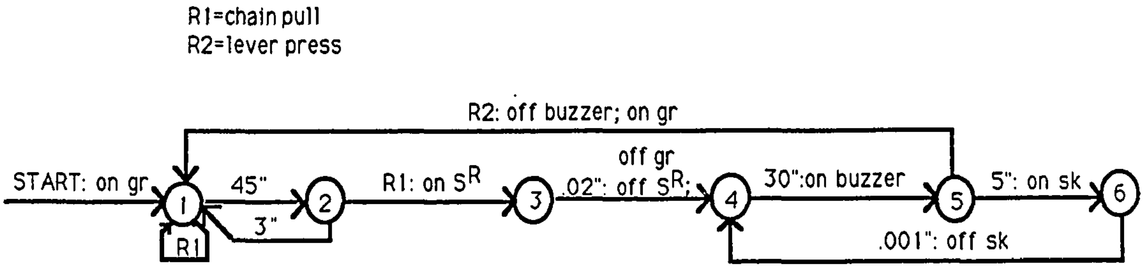
The procedure begins with a green light being on in the chamber. When a 45 second timer times out the rat has only 3 seconds to pull the chain. A chain pull will result in delivery of a food pellet and turn off the green light. If the rat does not pull the chain within the 3 seconds the 45 second timer begins again. Also if the rat pulls the chain while the 45 second timer is timing, the timer resets.

If the animal pulled the chain in time and obtained the food pellet, then the procedure continues with the timing of a 30 second timer. When this timer times out a buzzer is turned on. The rat now has 5 seconds to press the lever, which will turn off the buzzer, turn on the green light, and cause the procedure to begin again with the previous 45 second timer timing again.

If the rat does not press the lever within 5 seconds, a brief pulse of shock (lasting .001 second) is delivered, the buzzer is turned off, and the 30 second timer starts timing again. The cycle consisting of the 30 second timer--buzzer--5 second timer--brief shock keeps occurring until the lever press is eventually made. [1 off for each error up to 9]

Work your state diagram out on the back of one of the test pages, and draw a neat version of your state diagram in the space below. Remember that this procedure must be done with a single state set (a single state of interconnected states).

## Scoring Protocol




1. Do all states have an input? (Input has to be in front of colon) (1)

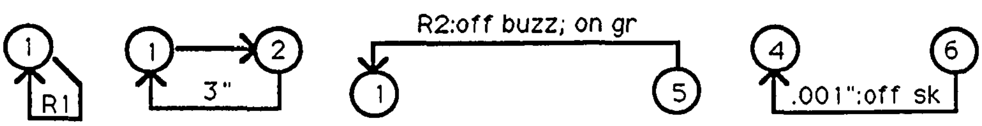
2. Do all states have only one input? (1)

3. Are all of these inputs diagrammed? (45", R1, 30", 5") (1)

4. Are all of these outputs diagrammed? (off gr, on gr, on buzz, on Sk) (1)

5. Is START: on gr used? (1)

6. Are the 4 transitions drawn correctly? (4)




7. Do all transition lines have arrows? (1)

8. Are inputs and outputs separated by a colon? (1)

9. Are multiple outputs separated by a semicolon? (1)

10. Is the seconds symbol used appropriately? (1)

11. Is .02 seconds used for the food pellet? (1)

12. Is SR turned on and off appropriately; is sk turned on and off appropriately? (2)

13. Is the sequence of events correct? (1)

14. Are R1 and R2 differentiated in all 3 places? (1)

18 points possible

Appendix H

Psychology 151, Winter 1987
Follow-Up Questionnaire

Questionnaire - State Notation

Name: ________________

I. The following four questions are asking you to give your opinion about how well you think that you can state diagram. Circle the number that best applies for each question.

1. If given the name of a schedule of reinforcement or a procedure assigned from Concepts and Principles, I could draw the state diagram.

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| | Not at all | Poorly | Somewhat | Fairly well | Very well |
| Had program | 0% | 0% | 9% | 65% | 26% |
| Didn't have | 3% | 9% | 30% | 49% | 9% |

2. If given a description of a procedure that I had studied in class, such as "The first response made after 1 minute receives 3" of grain, but any response made before 1 minute resets the 1 minute timer", I could draw the diagram.

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| | Not at all | Poorly | Somewhat | Fairly well | Very well |
| Had program | 0% | 0% | 9% | 35% | 57% |
| Didn't have | 3% | 3% | 9% | 46% | 39% |

3. I could look at a state diagram without being told the name of the procedure and be able to explain the procedure accurately. The procedure would be similar to, but not exactly like ones I had studied in class.

| | 1<br>Not at all | 2<br>Poorly | 3<br>Somewhat | 4<br>Fairly well | 5<br>Very well |
|---|---|---|---|---|---|
| Had program | 0% | 4% | 7% | 35% | 44% |
| Didn't have | 0% | 0% | 27% | 52% | 21% |

4. I could read a description of a procedure and then look at a state diagram of the procedure, and tell whether the diagram was drawn correctly or not. For example, I would know whether each state had the correct inputs and outputs, whether resets and transitions were used correctly, etc. The procedure would be similar to, but not exactly like ones I had studied in class.

| | 1<br>Not at all | 2<br>Poorly | 3<br>Somewhat | 4<br>Fairly well | 5<br>Very well |
|---|---|---|---|---|---|
| Had program | 0% | 4% | 13% | 48% | 35% |
| Didn't have | 3% | 0% | 24% | 39% | 33% |

II. In what aspects of state notation do you feel proficient?

| | | |
|---|---|---|
| Had program | 6 | Draw diagram from description |
| | 4 | Reading and understanding diagrams |
| | 4 | Describing diagrams |
| | 3 | Draw named diagrams |
| | 3 | Most aspects |
| | 1 | Editing diagrams |
| | 1 | None |
| Didn't have | 10 | Draw diagram from description |
| | 6 | Describing diagrams |
| | 5 | Drawing diagrams |
| | 2 | Most aspects |
| | 1 | Certain types of procedures |
| | 1 | Symbols |
| | 1 | Inputs and outputs |
| | 1 | None |
| | 1 | Draw named diagrams |
| | 1 | Reading diagrams |

Need more practice?

| | | |
|---|---|---|
| Had program | 5 | Naming diagrams |
| | 4 | Draw named diagrams |
| | 3 | Z pulse and decision diamonds |
| | 3 | Draw diagram from description |
| | 2 | Simple diagrams |
| | 1 | Memorizing specific diagrams |
| | 1 | Complex procedures |
| | 1 | Z pulse and parallel state sets |
| | 1 | Z pulse only |
| | 1 | Explaining diagrams |
| | 1 | General practice needed |
| | 1 | No practice needed |
| | 1 | No response |
| Didn't have | 5 | Draw named diagrams |
| | 4 | Label diagrams |
| | 4 | Didn't specify what kind of practice needed |
| | 3 | Label inputs, outputs |
| | 3 | Name procedures |
| | 3 | Draw diagrams from descriptions |
| | 2 | No practice needed |
| | 1 | Transitions |
| | 1 | Diagramming specific procedures |
| | 1 | Don't know |

III. Do you anticipate using state notation outside of Psychology 151?
If so, how?

| Had program | No | Don't know | NR | Yes | |
|---|---|---|---|---|---|
| | 13% | 35% | 0% | 52% | 39%-Other psych classes |
| | | | | | 9%-Explain procedures |
| | | | | | 4%-Computer prog |
| Didn't have | 39% | 15% | 13% | 33% | 24%-Other psych classes |
| | | | | | 6%-Explain procedures |
| | | | | | 3%-Computer prog |

IV. If you had to learn state notation and you could choose any method or combination of methods from the ones listed below, which would you choose? Check all that apply:

A. _____ Take Psychology 151 just as you did, which includes a combination of lectures, readings, objectives, and exams over state notation.

B. _____ Listen to Dr. Michael's lectures about state notation.

C. _____ Read the section in Concepts and Principles by Dr. Michael about state notation on your own.

D. _____ Go through Esther's programmed materials on state notation (a self-instructional program with practice exercises in state diagramming)

Please explain your choice(s) below.

| | Had program | Didn't have |
|---|---|---|
| A only | 0% | 30% |
| B only | 0% | 0% |
| C only | 0% | 6% |
| D only | 9% | 3% |
| AB | 4% | 6% |
| AC | 0% | 0% |
| AD | 35% | 15% |
| BC | 0% | 3% |
| BD | 0% | 0% |
| ABC | 0% | 18% |
| ABD | 17% | 3% |
| BCD | 9% | 9% |
| ACD | 0% | 3% |
| All | 26% | 3% |

Appendix I

Scoring Protocol Used to Obtain Covariate Score

Scoring Protocol for Pre-questionnaire

Undergraduate questionnaire:

| | | |
|---|---|---|
| 1. Major | Psychology (1) | All others (0) |
| Year in college | Fresh/Soph (1) | Junior/Senior (2) |

| | | |
|---|---|---|
| 2. Courses at WMU | 194 | (3) |
| | Other psychology | (2) |
| | Computer programming | (2) |
| | Math | (1) |

3. If prior training in state notation, subject cannot be used in study

| | | |
|---|---|---|
| 4. Topics | Schedules | (2) |
| | EAB research | (2) |
| | Operant conditioning | (2) |
| | Respondent conditioning | (2) |
| | Formal logic | (1) |
| | Behavior modification | (1) |
| | Operant lab (1-4 depending on type of work done) | |

| | | |
|---|---|---|
| 4. High School | Computer science | (2) |
| | Rat lab | (2) |
| | All other courses | (1) |
| | Electronic systems | (2) |

5. Operant research (1-4) depending on type of work done

SKED (1-4) depending on type of work done (but may not be able to be used in the study - ask further questions)

Graduate questionnaire:

| | | |
|---|---|---|
| 1. Program | Not in Masters program yet (PTC) | (1) |
| | Masters student | (3) |
| | Doctoral student | (4) |
| | Add 1 pt. to masters and doctoral student if in the EAB program | |

| | | |
|---|---|---|
| 2. Courses at WMU | 610 | (4) |
| | 608 | (2) |
| | 151 Rat lab | (2) |
| | 611 | (4) |
| | 516 | would probably be ineligible for use in study; 516 teaches state notation |
| | Other EAB courses | (2-4) |
| | Computer prog | (2) |
| | Math | (1) |

| | | |
|---|---|---|
| 3. Topics | Schedules | (2) |
| | EAB research | (2) |
| | Operant conditioning | (2) |
| | Respondent conditioning | (2) |
| | Formal logic | (1) |
| | Flowcharting | (1) |
| | Behavior modification | (1) |
| | Operant lab | (1-4, depending on type of work) |
| 4. High School | Computer science | (2) |
| | All others | (1) |
| | Rat lab | (2) |
| 5. Research | | (1-6, depending on type of research) |

If experience with SKED, check to see that person has not had any formal training in SKED - may make them ineligible for study

Appendix J

Psychology 151, Winter 1988
End-of Semester Questionnaire

Psychology 151, Winter 1988 Questionnaire

Please do not put your name on this questionnaire.

What is your classification at Western? (Freshman, Sophomore, etc.)

| Freshman | Sophomore | Junior | Senior | Graduate | PTC |
|---|---|---|---|---|---|
| 32 | 21 | 4 | 6 | 3 | 1 |
| 48% | 32% | 6% | 9% | 4% | 1% |

Have you had any exposure to state notation before you took this class?
If so, please explain briefly.

All respondents reported "no".

Please make a few brief comments about the state notation program:

1. How understandable were the instructions and explanations in the program?

| Very, quite, easy | Generally clear, ok | Too complex | Want more detail |
|---|---|---|---|
| 49 | 15 | 2 | 1 |
| 74% | 22% | 3% | 1% |

2. Did you already know any of the concepts about state notation that were already taught in the program? If so, which ones?

| No | NR | Yes | |
|---|---|---|---|
| 58 | 6 | 3 | (1-extinction, 2-state to state diagramming in computer programming) |
| 87% | 9% | 4% | |

3. Did any of the material seem unnecessary? (Do you feel that you could have learned the same amount even if some of the material were cut out? If so, what parts?)

| No | Don't know | Yes | |
|---|---|---|---|
| 58 | 3 | 6* | * 5 wanted less practice |
| 87% | 4% | 9% | |

4. Were any important concepts about state notation omitted from the program, or not covered in as much detail as they should have been?

| No | Don't Know | NR | Yes | |
|---|---|---|---|---|
| 52 | 6 | 4 | 5 | 1 - program too difficult at the end |
| 78% | 9% | 6% | 7% | 1 - Z pulses |
| | | | | 1 - program generally needs more explanation |

5. a. Did you answer MOST of the items correctly on the first try?

| No | Some | NR | Yes |
|---|---|---|---|
| 8 | 2 | 2 | 55 |
| 12% | 3% | 3% | 82% |

b. When you missed an item, did you understand the item after reading the answer?

| No | Yes | Mostly | NR |
|---|---|---|---|
| 0 | 38 | 3 | 26 |
| 0% | 57% | 4% | 39% |

6. Was the presentation of the material interesting/could have been made more interesting/dull? Please explain.

| Interesting | Could be more interesting | Dull | NR |
|---|---|---|---|
| 47 | 7* | 1 | 12 |
| 70% | 10% | 1% | 18% |

*Too many terms
Not enough common language used
Want more detail
Want more technical terms
Want more human examples

7. Did you find the job aids helpful? Did you use them?

| Helpful | Don't know | No | NR |
|---|---|---|---|
| 55 | 3 | 4 | 5 |
| 83% | 4% | 6% | 7% |

8. a. Presently, in what features of state notation do you feel that you are proficient?

| None | Most | All | No Response |
|---|---|---|---|
| 1 | 26* | 14 | 26 |
| 1% | 39% | 21% | 39% |

*13 cited that they felt proficient in reading and drawing diagrams

b. Need more practice?

| No | Yes | Don't Know | No Response |
|---|---|---|---|
| 15 | 24* | 1 | 27 |
| 22% | 37% | 1% | 40% |

* 4-reading diagrams
1-drawing diagrams
1-difficult diagrams
1-symbols
1-detail

9. Do you ever expect to use state notation outside of Psychology 151? If so, how?

| Yes | Don't know | No | NR |
|---|---|---|---|
| 19* | 17 | 24 | 7 |
| 29% | 25% | 36% | 10% |

* 12-other psychology classes
* 5-research
* 2-applied settings

10. Suppose that you have a friend who wants to learn state notation. Which of the following methods would you recommend? Check all that apply.

___ Listen to Dr. Michael's lectures on Psychology 151 on state notation.

___ Work through "A program of Instruction to Teach State Notation" by Esther Shafer

___ Read the section in Concepts and Principles written by Dr. Michael titled "State Notation of Common Behavioral Procedures", in which the schedules of reinforcement are diagrammed and described.

Why did you choose this method or combination of methods?

| | | |
|---|---|---|
| 3 | 4% | Michael lectures only |
| 5 | 7% | Michael section only |
| 19 | 28% | Shafer program only |
| 1 | 1% | Michael lectures and program |
| 7 | 10% | Michael lectures and Shafer program |
| 2 | 3% | Michael section and Shafer program |
| 28 | 42% | All 3 methods |
| 2 | 3% | No response |

11. Suppose that in the future you want to refresh your memory about state notation. Which of the following documents will you refer to, and why?

___ Dr. Michael's section titled "State Notation of Common Behavioral Procedures".

___ Esther Shafer's "Program of Instruction to Teach State Notation".

Please explain your choices.

| | | |
|---|---|---|
| 16 | 24% | Michael's section only |
| 37 | 55% | Shafer's program only |
| 13 | 19% | Both documents |

# Appendix K

## Latest Version of State Notation Program

# A PROGRAM OF INSTRUCTION TO TEACH STATE NOTATION

Esther Shafer


## Introduction

State notation is a method of diagramming schedules of reinforcement and other experimental procedures. State notation is comprised of a relatively small number of concepts, yet with these concepts you will be able to diagram the schedules of reinforcement and all of the procedures that you use in the experimental laboratory. A state diagram is the visual representation of a schedule or procedure drawn with the notation system. State notation is a useful tool, because a state diagram shows all of the complexities and relationships in a procedure. Once you have mastered state diagramming, you will find that a state diagram is usually easier to understand than a verbal description of the same procedure. By looking at state diagrams of the schedules of reinforcement, you can easily compare and contrast the schedules, immediately seeing the similarities and differences between each one.

These programmed materials were specifically designed to help introductory level students in behavior analysis learn the basics of state notation in a relatively easy and interesting manner, so that they can use state notation for the applications mentioned above. However, anyone desiring an introduction to the basics of state notation may find this program useful. The program teaches students how to read state diagrams, as well as draw them. The author hopes that students who complete this program will find that state notation is a skill which they will use now, and in the future.

### Basic Concepts

The events and relations which comprise the state notation system will be discussed briefly, just to familiarize you with some of the concepts that you will encounter in the program. You will learn how each concept is used in state notation as you work through the program.

#### States and State Sets

The word "state" refers to a static condition of the environment. A procedure such as one that you conduct in the laboratory can be thought of as a series of events and relations that can occur as a result of the environment changing. Each state is one component of the entire procedure. In each state, one or more events which change the environment are depicted, such as the passage of time, or if a response is made. Each state also depicts events which occur as a result of the environment being changed. For example, after a certain amount of time has passed, a stimulus could be presented or withdrawn. A state diagram, then, is a picture of a procedure that an experimenter has arranged, or the independent variable of the experiment. The state diagram does not show what organisms do - only how the environment will change if the organism should behave in a certain way.

When a procedure is being carried out, only one state is in effect at any given moment, and one state is always in effect. Some procedures are represented by one state set, while some procedures are best represented by two or more state sets, called parallel state sets.

**Input and Output Variables**

The terms "input" and "output" are used to designate the events and relations used in state notation. Each of the 10 variables is either an input or an output except for the Z pulse, which can be both an input and output. The terms "input" and "output" come from the use of a computer as a device to control experiments in the laboratory. Inputs can be thought of as responsible for changing the experimental environment, while outputs are ways that the experimental environment can change. The terms "input" and "output" in state notation are not entirely consistent with the way that they are sometimes used in computer programming; therefore, you will need to be aware that these terms have a special meaning in state notation.

**Structure of the Program**

The program is divided into six sections. Most of the pages in the program instruct you to answer questions and/or draw state diagrams. You should write in the program booklet in the spaces provided. Answer pages for each section are located at the end of each section. You must check each answer immediately after you write it to be sure that you have answered it correctly, because each new question builds on the previous material. Four of the pages in the program are called "job aids" (p. 135, 138, 139, 147). They are to aid you in reading and drawing diagrams while you are still learning. They were designed to be removed from the program so that you can refer to them easily; however, you may wish to xerox these four pages before beginning the program so that you do not lose them. You may want to use them throughout the program, or you may find that you don't need them after gaining some practice. Some of the exercises suggest that you try doing them without the job aids to test yourself.

You are now ready to begin the program!

## Section 1
## The Basics of Reading State Diagrams

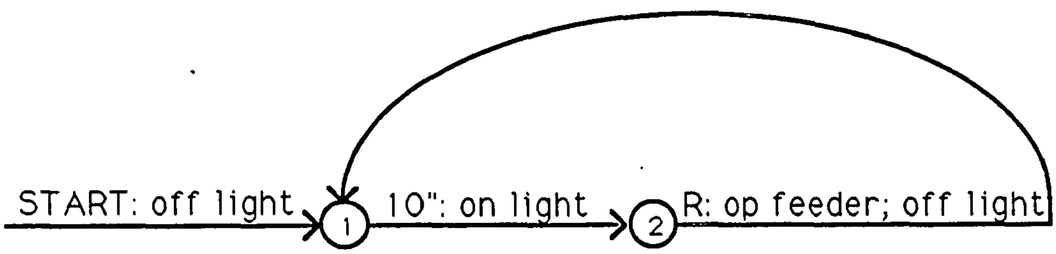
You are probably wondering "What does a state diagram look like?" State diagrams are ways of "drawing" a picture of an experimental procedure in a way that makes the procedure easier to understand.

---

Let's take a simple procedure that you are already familiar with and draw the state diagram.

Pretend that you have a rat in an operant chamber.
The chamber has a light, a lever, and a food pellet dispenser, which you operate.
The procedure starts with the light off. Then, you will wait 10 seconds.
When the 10 seconds are up, you will turn on the light. Then you will wait for the rat to press the lever. When the rat presses the lever, you will deliver a food pellet and turn the light off. Then you will repeat the procedure.

Here is the state diagram. Follow the diagram and see how the description corresponds to the symbols in the diagram.




The procedure starts with the light off.

After 10" (seconds), turn the light on.

When the rat presses the lever, operate the feeder (deliver a food pellet), and turn the light off. Repeat the procedure.*

R = lever press

*If the rat doesn't press the lever, the procedure cannot continue!

---

Notice how the state diagram has all of the information about the procedure in a format that is easy to follow. The first section of this program will teach you to verbalize the sequence of events in a state diagram. This skill is easy, once you learn some basic rules about how state diagrams are read. Now, please go to the next page.

**Rules**

Here is the same state diagram that you just saw. Now we will learn some rules about how to read state diagrams. This page will serve as a reference for you as you work through the program. You may want to take this page out of your program and keep it handy so that you can refer to it later. Notice how the description of the diagram corresponds to the rules.

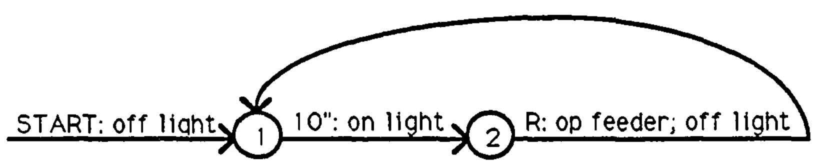
---

1. Circles with numbers are called "states".
   ① is State one, ② is State 2, etc.

2. If there is a START, begin reading there.
   If there is no START, begin reading at
   State one ① .
   ① .

3. Arrows ------> are called "transition arrows". They tell you what state to go to next.

R = lever press




4. For each new state, read from LEFT to RIGHT.

5. Read everything in one state before going to the next state.

The procedure starts with the light off. After 10 seconds, you will turn on the light. When the rat presses the lever, operate the feeder and turn the light off. The procedure repeats at State 1.

---

Now, please go to the next page.

Let's take a closer look at rule #4.

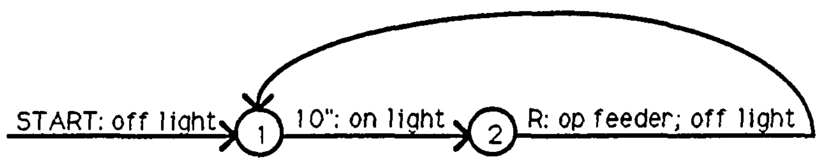
1. Circles with numbers are called "states".
   (1) is State one, (2) is State two, etc.

2. If there is a START, begin reading there.
   If there is no START, begin reading at
   State one (1).

3. Arrows ----> are called "transition arrows".
   They tell you what state to go to next.

R = lever press

4. For each new state, read
   from LEFT to RIGHT.

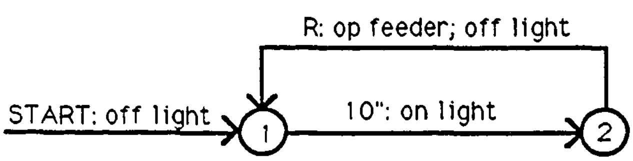



5. Read everything in one state
   before going to the next state.

The procedure starts with the light off. After 10 seconds, turn on the light. When the rat presses the lever, operate the feeder and turn the light off. The procedure repeats at State 1.

The way that the diagram is drawn above makes it easy to read from left to right. However, most diagrams that you will see are drawn like the one below. Notice how the path goes ABOVE the diagram instead of out to the right. When you get to State 2, you STILL read whatever is on the line from left to right.

R = lever press



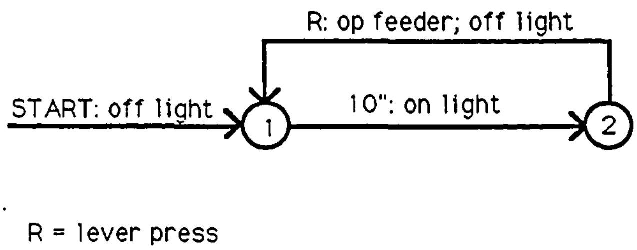

The procedure starts with the light off.

After 10 seconds, the light is turned on.

When the rat presses the lever, operate the feeder and turn the light off. Notice that when the procedure repeats, it returns to State one.

Now, please go to the next page.

On the previous pages, you have seen how to read a state diagram. We can call this "reading in everyday language" - you read it as though you were describing the procedure to someone who doesn't know about state diagrams.




The procedure starts with the light off. After 10 seconds, the light is turned on. When the rat presses the lever, operate the feeder and turn the light off. Repeat the procedure at State 1.

---

Another way to read state diagrams is to say the name of the state ("In State one...") and to say "Transition to State ___") whenever you see the transition arrow going to a new state. We can call this "State diagramming language".

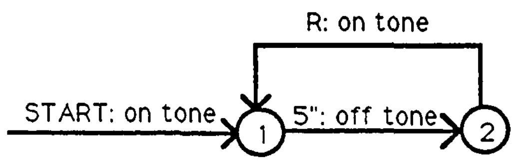
The procedure starts with the light off. In State 1, a 10 second timer times out, after which a light is turned on, and transition to State 2 occurs. In State 2, when a response is made, the feeder operates, the light is immediately turned off, and transition back to State 1 occurs.

---

As you learn about behavioral procedures, you will see both everyday language and "State diagramming language" used to describe those procedures. You should be able to recognize that both types of descriptions mean the same thing. However, when you are describing a complex procedure, you will find that "State diagramming language" is best to use. You will be less likely to omit important details and your descriptions will be easier for a reader to follow.

---

Here is a state diagram. Write two descriptions; one in everyday language, and one in "State diagramming language". Don't forget to check your answer (See p. 143)



## Inputs and Outputs

The next two pages will serve as a handy reference for you as you work through the program. You may want to take these pages out of the program and refer to them whenever you need to.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

The symbols used in state notation are called inputs and outputs.
Here are all of the inputs and outputs used in state notation.
These two pages explain what the symbols stand for and how to talk about them when you are reading a diagram.

### Inputs

1. **START** — Not all state diagrams need a **START**.
START is used when a stimulus (such as a light) needs to be on before the procedure begins.
Say "The procedure starts with..."

2. **R** (response) — Use **R** to designate any behavior.
Say "When (or 'if') a response is made..."

   **R1,R2** — Different types of responses, such as **R1**=lever press,
**R2**=chain pull

   **nR** — Different numbers of responses, such as **10R**

   **v** — v stands for variable, or average.
For example, say "When a variable number of responses are made, the average of which is 10..."

3. Time (T)

   ' minutes — For example: 5' means five minutes.
Say "When 5 minutes passes..." or "When a 5 minute timer times out..."

   " seconds — For example: 5" means five seconds.

   **v** — v means variable, or average.
v5" = say "When a variable 5 second timer times out..."
OR "After a variable period of time passes, the average of which is 5 seconds..."

4. **Z** — Stands for Z pulse. A **Z** pulse as an input functions as an output in another state set. You will learn to use Z pulses later in the program.

## Outputs

1. Transition arrow — Say "Transition to state ____ occurs."

   -------------->

2. Reset — Starts the same state over again.
   Say "Reset to state ___ occurs."


3. Stimulus (S)*

| | |
|---|---|
| **on S1, off S1**<br>**on S2, off S2,** etc. | different kinds of stimuli (light, tone, etc.) |
| **on SR**<br>**off SR** | stands for unconditioned reinforcement |
| **op feeder** | stands for "operate feeder"<br>op feeder is sometimes used as an alternative to "on SR" |
| **op dipper** | stands for "operate water dipper"<br>op dipper is sometimes used as an alternative to "on SR" |
| **on SK**<br>**off SK** | stands for shock |

* In SKED, S (stimulus) is sometimes used to designate all reinforcement operations.

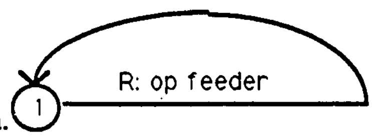
4. **STOP** — Means that the procedure stops here.

5. **Z** — Stands for Z pulse. A Z pulse as an output functions as an input in another state set. You will learn to use Z pulses later in the program.

Now that you have seen the basic rules about state diagramming and have a list of inputs and outputs, you are ready to begin reading state diagrams. You may want to keep the pages titled **Rules** and **Inputs and Outputs** in front of you to use as references.

Write the description of each diagram next to it.

---

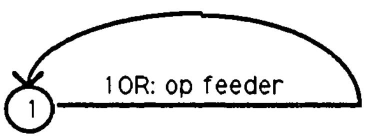
a. 


In State one, when a response is made, the feeder operates and transition back to State one occurs.

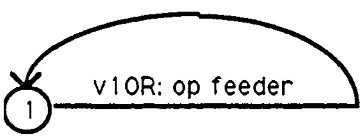
**OR**

When a response is made, the feeder operates. The procedure repeats.

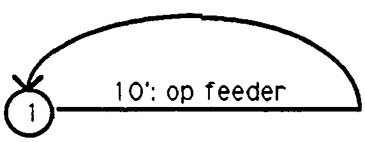
---

b. 

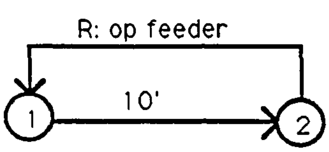

---

c. 

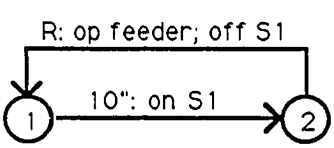

---

d. 


---

e. 


---

f. 


S1 = light

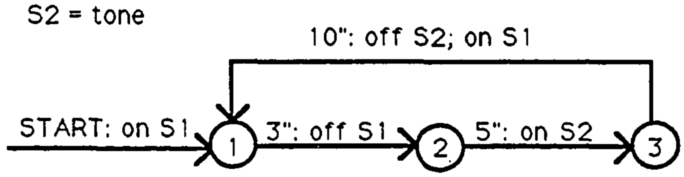
---

The state diagrams that you just read were probably pretty easy for you. Now, we'll add some more features to the diagrams. Remember, use the **Rules** and **Inputs and Outputs** pages as references. Write the descriptions of the diagrams.

---

S1 = light
S2 = tone

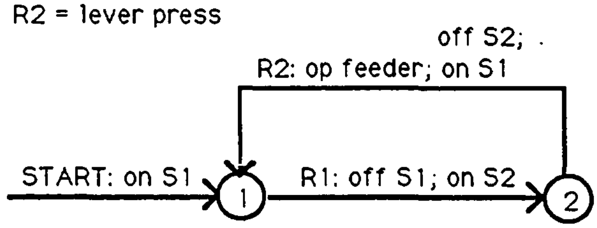
a.




---

S1 = red light
S2 = green light
R1 = chain pull
R2 = lever press

b.

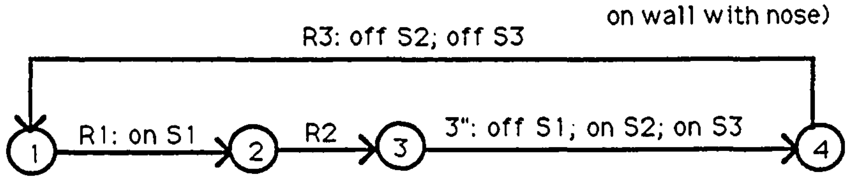



---

S1 = red light
S2 = green light
S3 = tone

R1 = chain pull
R2 = lever press
R3 = spot touch
(rat touches spot on wall with nose)

c.

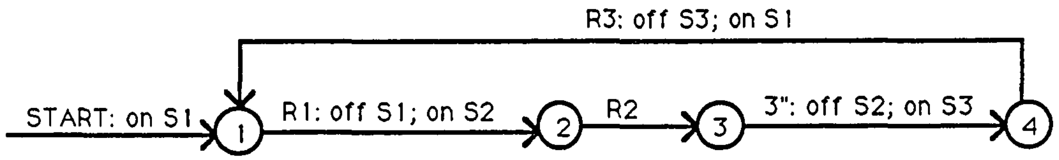


Write a description of this state diagram (either in "State diagramming language" or in "everyday language"). You may want to test yourself by writing this description without looking at your pages of **Rules** or **Inputs and Outputs.**

---

R1 = lever press
R2 = chain pull
R3 = spot touch

S1 = tone
S2 = yellow light
S3 = green light



## Answers - Section 1

Note: Don't worry if your answers are not word-for-word with the answer key - as long as you have all of the information in a logical order.

Page 137 — Everyday language: The procedure starts with a tone on. Then, when 5 seconds passes, the tone is turned off. Now, when a response is made, the tone is turned on again and the procedure repeats.

State diagramming language: The procedure starts with a tone on and transition to State 1. In State 1, when a 5 second timer times out, the tone is immediately turned off and transition to State 2 occurs. In State 2, when a response is made, the tone is immediately turned on and transition back to State 1 occurs. The procedure repeats.

Page 140

b. In State 1, when 10 responses are made, the feeder operates (a food pellet is delivered), and transition back to State 1 occurs.
OR, when 10 responses are made, the feeder operates and the procedure repeats.

c. In State 1, when a variable number of responses are made, the average of which is 10, the feeder operates and transition back to State 1 occurs.
OR, when a variable number of responses occur, the average of which is 10, a food pellet is delivered and the procedure repeats.

d. In State 1, when a 10 minute timer times out, the feeder operates and transition back to State 1 occurs.
OR, when a 10 minute timer times out, a food pellet is delivered and the procedure begins again.

e. In State 1, when a 10 minute timer times out, transition to State 2 occurs. In State 2, when a response is made, the feeder operates and transition back to State 1 occurs.
OR, The procedure begins with timing out of a 10 minute timer. When a response occurs after the timer has timed out, a food pellet is delivered and the procedure repeats.

f. In State 1, after a 10 second timer times out, a light is immediately turned on and transition to State 2 occurs. In State 2, when a response is made, the feeder operates, the light is turned off, and transition back to State 1 occurs. OR, After 10 seconds, a light comes on. Then, when a response is made, the feeder operates and the light is turned off. The procedure repeats.

Page 141 (Only one description will be given from now on).

a. The procedure begins with the onset of a light. Then, when a 3 second timer times out, the light is turned off. Now, when a 5 second timer times out, a tone is turned on. The next event to occur is the timing out of a 10 second timer, after which the tone is turned off and the light is turned on again. The procedure repeats.

* Notice that when the procedure repeats, it goes back to State 1. Since S1 was already on when the procedure started, it must be turned on in State 3 so that it will already be on when the procedure repeats.

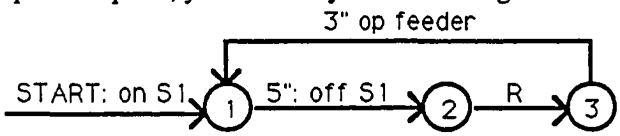
b. The procedure starts with the turning on of a red light. Then, when a chain pull response is made, the red light is turned off and a green light is turned on. Now, when a lever press occurs, a food pellet is delivered, the green light is turned off, the red light is turned on, and the procedure repeats. Notice that S1 is turned on again in State 2.

c. The procedure begins with a chain pull, after which a red light comes on. Then, when a lever press occurs, a 3 second timer begins timing out. After the 3 second timer times out, the red light is turned off, a green light is turned on, and a tone is turned on. Now, when a spot touch is made, the green light and tone are turned off, and the procedure repeats.

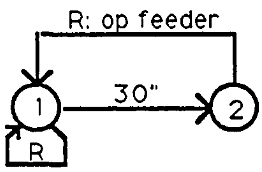
Page 142 The procedure starts with the onset of a tone. When a lever press is made, the tone is turned off and a yellow light is turned on. Now, when a chain pull is made, a 3 second timer times out, after which the yellow light is turned off, and a green light is turned on. Now, when a spot touch is made, the green light is turned off and the tone is turned on. The procedure repeats. Notice that S1 is turned on in State 4.

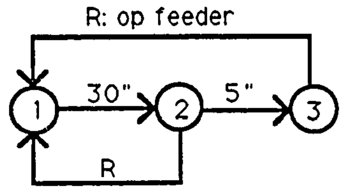
## SECTION TWO
## Reading More Complex Diagrams

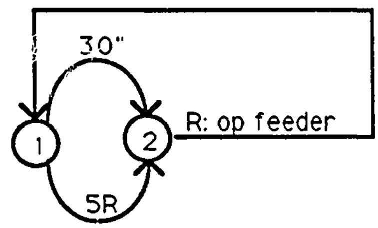
Up to this point, you have only seen state diagrams that look like this:

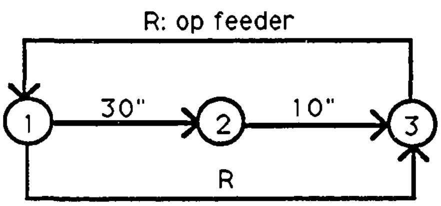


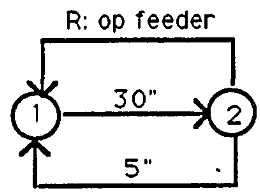

Now you will begin to see state diagrams that look like these:

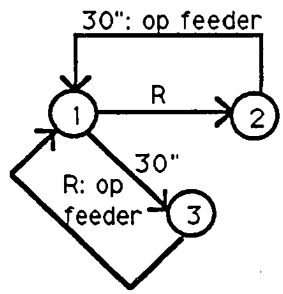


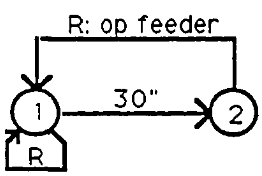




What is the difference between the two types of diagrams?
The first type has ONLY ONE transition line and ONLY ONE input per state, while the second type has MORE THAN ONE transition line and MORE THAN ONE input per state!

---

We still use the same rules for reading the diagrams -
Rule number 5 says: "Read everything in one state before going on to the next state!"

For example:

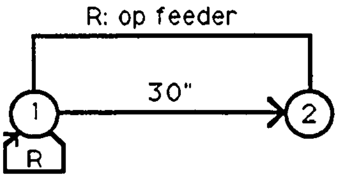



State one has 2 inputs - a 30 second timer and a response.

In State one, there are two events that can happen - the 30 second timer timing out <u>and</u> a response. <u>If</u> a response is made in state one, the 30 second timer is reset (starts over). If a response is <u>not</u> made, the 30 second timer will time out and transition to State 2 will occur. In State 2, if a response is made, the feeder operates and the procedure repeats.

Now, please go to the next page.

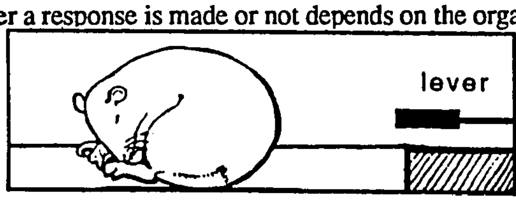
Here are some important details about reading state diagrams that you will need to learn.
Let's take a closer look at the diagram that you just saw.




Put your finger on State one.
Notice that two things can happen in State one - a response (R) can be made, and a 30 second timer can time out.

As soon as State one begins, the 30 second timer starts timing.
(Timers immediately begin timing at the beginning of a state).

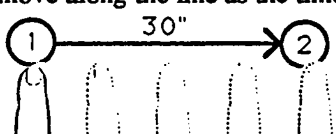
Whether a response is made or not depends on the organism.



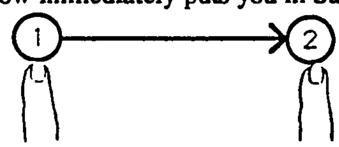

The rat may not make a response at all in State one. If this is the case, the timer will time out and transition to State two will occur. If the rat does not make a response in State two, the procedure will stay in State two indefinitely.

---

In State 1, don't move along the line as the timer is timing out! (Like this!)

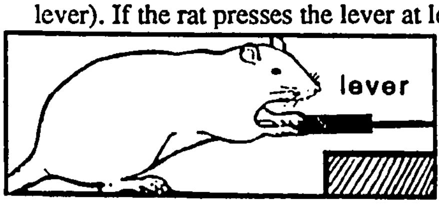


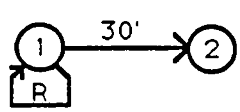

You are always in State one until the transition happens, and then the transition arrow immediately puts you in State two!




---

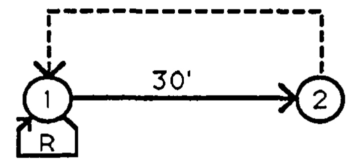
When the rat makes a response, the timer will reset and start over again (as many times as the rat presses the lever). If the rat presses the lever at least once every 30", the procedure will always stay in state one.



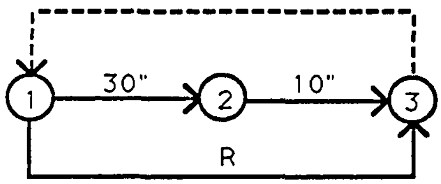



## Two Inputs/One State

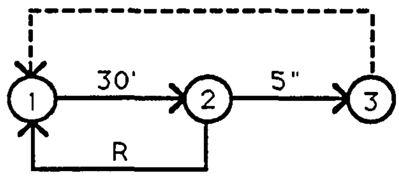
This page shows different types of state diagrams that have more than one input coming out of one state. You may take this page out and use as a reference for future diagrams.



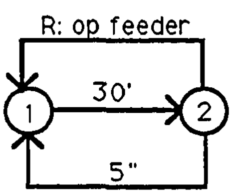

State 1 has a 30" timer AND a response. If the response is made, it will reset the 30" timer. If the response is NOT made, the 30" timer will time out and State 2 will be in effect.



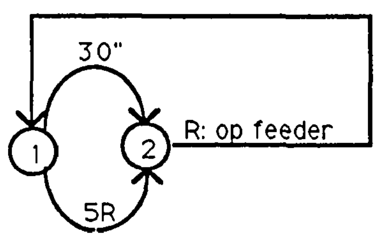

State 1 has a 30" timer AND a response. If the response is made, the procedure goes to State 3. If the response is NOT made, the 30" timer will time out and State 2 will be in effect.



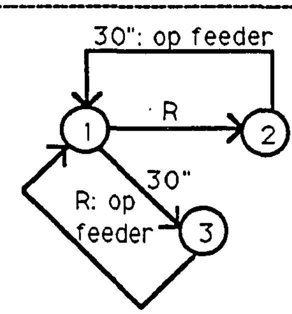

State 2 has a 5" timer AND a response. When State 2 begins, the 5" timer will start timing. IF a response is made before 5" passes, the procedure goes back to State 1. If a response is NOT made, the procedure goes to State 3.




State 2 has a 5" timer AND a response. When State 2 begins, the 5" timer starts timing. IF a response is made before the 5" timer times out, the feeder operates. If a response is NOT made,the procedure goes back to State 1.




State 1 has a 30" timer AND 5 responses. Whichever happens first will cause transition to State 2. At the beginning of State 1, the timer begins timing. If 5 responses are made before the 30" timer times out, State 2 is entered. If 5 responses are not made before the timer times out, State 2 is entered after 30".

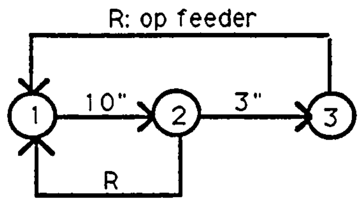



State 1 has 2 paths - a 30" timer and a response. If a response is made before the 30" timer times out, the procedure goes to State 2. If the 30" timer times out before a response is made, the procedure goes to State 3.

Now you are ready to practice reading some diagrams. You may use the page called **"Two Inputs/One State"** to help you.
Write the description of the diagram next to the diagram. (Answers on p. 151).

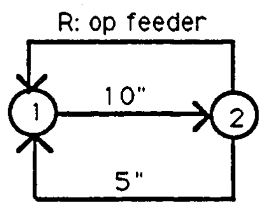
---

a.



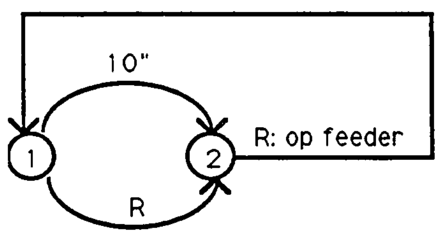

---

b.



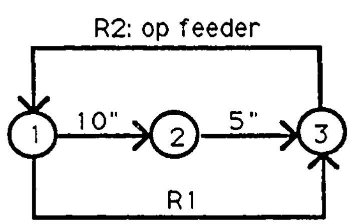

---

c.




---

d.



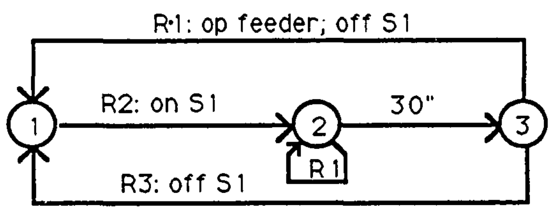

R1 = lever press  R2 = chain pull

---

Here are diagrams which combine several features of diagrams that you have had before.
You may refer to any of the job aids that you used before to help you.
Write the description next to the diagram.

---

a.

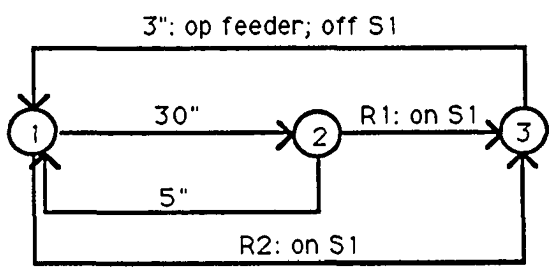



S1 = green light

R1 = lever press
R2 = chain pull
R3 = spot touch

---

b.




S1 = green light

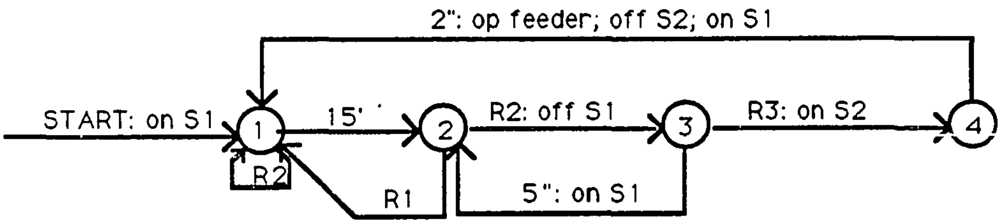
R1 = lever press
R2 - chain pull

---

Write the description of this diagram. You may want to test yourself by not using any of your job aids.

---

S1 = red light R1 = chain pull
S2 = blue light R2 = lever press
R3 = spot touch



## Answers - Section 2

Page 148

a. In State 1, when a 10 second timer times out, transition to State 2 occurs. In State 2, if a response is made before a 3 second timer times out, the 10 second timer in State 1 begins timing again. However, if a response is NOT made before the 3 second timer times out, transition to State 3 occurs. In State 3, when a response is made, the feeder operates and the procedure begins again.

b. In State 1, when a 10 second timer times out, transition to State 2 occurs. In State 2, if a response is made before a 5 second timer times out, the feeder operates and the procedure begins again. However, if a response is not made before a 5 second timer times out, the procedure begins again at State 1.

c. In State 1, whichever event occurs first will result in transition to State 2. The two events are the timing of a 10 second timer and a response. In State 2, when a response is made, the feeder operates and the procedure repeats.

d. First, a 10 second timer times out. However, if a lever press is made before the 10 second timer times out, transition to State 3 occurs. If the lever press is not made in State 1, then a 5 second timer times out in State 2 before transition to State 3 occurs. In State 3, when a chain pull occurs, a food pellet is delivered and the procedure repeats.

Page 149

a. In State 1, a chain pull will result in the turning on of a green light and transition to State 2. In State 2, when a 30 second timer times out, transition to State 3 occurs. However, if a lever press is made in State 2, the 30 second timer resets. In State 3, when a lever press is made, the feeder operates, the green light is turned off, and the procedure repeats. However, if a spot touch is made in State 3, the procedure begins again at State 1.

b. If a chain pull is made in State 1, transition to State 3 occurs and a green light is turned on. If the chain pull is NOT made in State 1, State 2 will be entered after the 30 second timer times out. In State 2, the organism only has 5 seconds to make a lever press; otherwise the procedure begins again in State 1 with the reset of the 30 second timer. If the lever press IS made in State 2, a green light is turned on and transition to State 3 occurs. In State 3, a 3 second timer times out, a food pellet is delivered, the green light is turned off, and transition back to State 1 occurs.

Page 150

The procedure starts with a red light on. In State 1, if the organism presses the lever, the 15' timer resets. If the organism does not press the lever in State 1, the 15' timer times out and transition to State 2 occurs. In State 2, if the organism pulls the chain, State 1 is re-entered. If the organism presses the lever, the red light will be turned off and transition to State 3 will occur. In State 3, the organism must touch the spot before the 5" timer times out; otherwise State 2 will be re-entered. If the organism DOES touch the spot before 5" passes, a blue light will be turned on and transition to State 4 occurs. After a 2" timer times out the feeder will be operated, S2 will be turned off, and S1 will be turned on. The procedure repeats at State 1.

## Section Three
## The Basics of Drawing State Diagrams

Now you are ready to begin drawing state diagrams. You already know most of what you need to know to begin drawing, but here are a few rules:

This diagram is drawn correctly according to the rules.

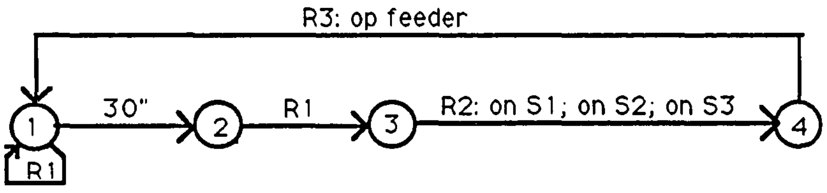
1. Every transition line has an input, and ONLY ONE input! **(30"and R are inputs)**

2. Inputs are the first symbol on the transition line.

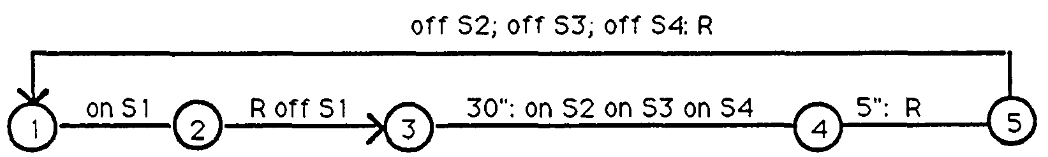
3. Every transition line has an arrowhead.

4. There can be more than one output following an input on the transition line. **(no limit!)**

5. If there are outputs on the transition line, separate the input and the output with a colon (:). Put semicolons (;) between the outputs.




This diagram is NOT drawn correctly. Write the rules that have been broken. (Answers p. 157).



Here are some diagrams to draw. You may use any job aids that you wish.
Some of the descriptions are written in "everyday language" and some are written in "State diagramming language".

---

a. In State 1, when 10 responses are made, transition to State 2 occurs. In State 2, when a 30 second timer times out, a food pellet is delivered and transition back to State 1 occurs. The procedure repeats.

---

b. The procedure starts with a green light on. In State 1, when a 10 second timer times out, transition to State 2 occurs. However, if a response is made in State 1, the 10 second timer resets. Once State 2 begins, a response is followed by the green light turning off and transition to State 3. In State 3, a response is followed by feeder operation, and the green light turning on again. The procedure repeats.

---

c. In State 1, a chain pull response results in a red light being turned on and transition to State 2. In State 2, the rat only has 5 seconds to make a lever press response; otherwise the procedure begins again at State 1 and the red light is turned off. If the rat does make a lever press response within 5 seconds, a drop of water (op dipper) is delivered, the red light is turned off, and the procedure begins again.

Here are some more diagrams to draw:

a. State 1 has two inputs; whichever event occurs first will result in transition to State 2. One of the inputs in State 1 is a variable 30 second timer; the other is a chain pull response. In State 2, when the rat makes five lever presses, a food pellet is delivered and the procedure repeats.

b. This procedure has four states. First, a chain pull response is followed by the turning on of a red light. Then, a lever press causes a 3 second timer to start timing. After the timer times out, the red light is turned off and a green and yellow light are turned on. Then, a spot touch results in the green and yellow light being turned off, and the entire procedure repeats.

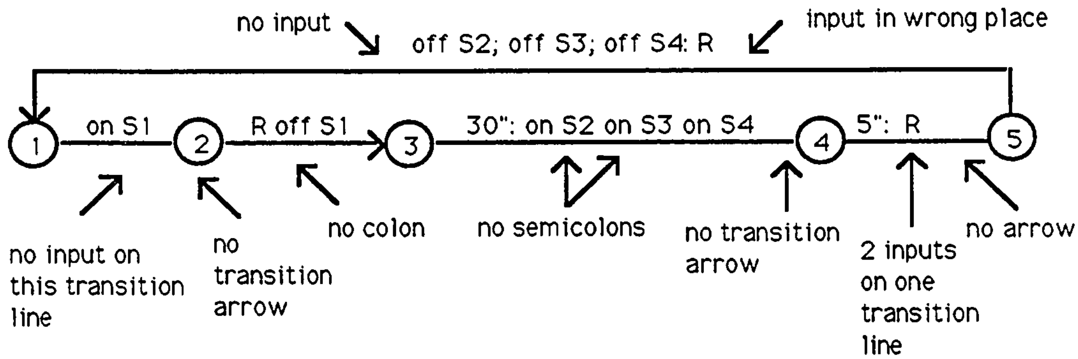
Draw this diagram. You might want to test yourself by not using any of your job aids.

---

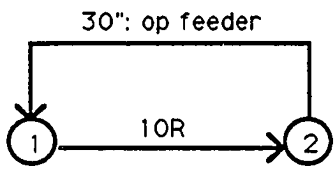
The procedure begins with the timing of a 30 second timer.
If a lever press is made while the timer is timing, the timer must begin timing again.
After the 30 second timer has timed out, if a variable number of lever presses are made, the average of which is 5, a red light is turned on. A chain pull will result in turning off of the red light and delivery of a food pellet. The procedure repeats.

## Answers - Section 3

Page 153

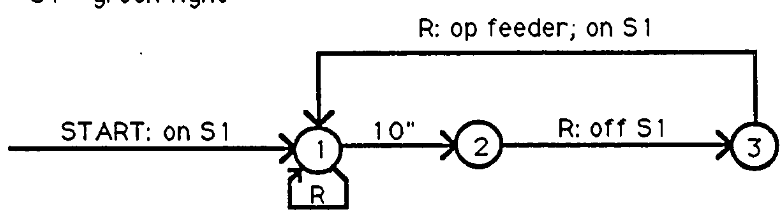



Page 154 **a.**

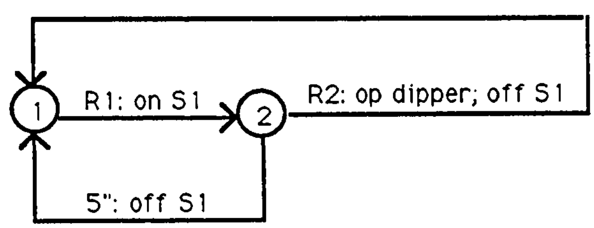



S1 = green light

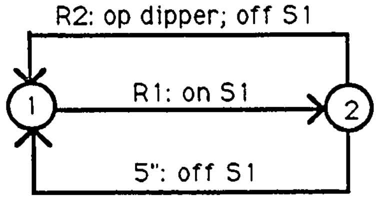
**b.**




**c.** R1 = chain pull
R2 = lever press
S1 = red light



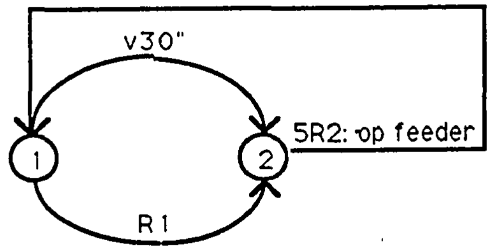

OR:



Page 155

**a.**

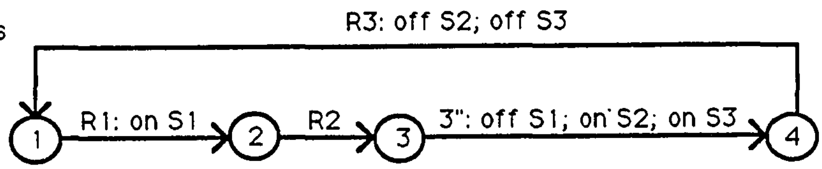
R1 = chain pull
R2 = lever press




**b.**

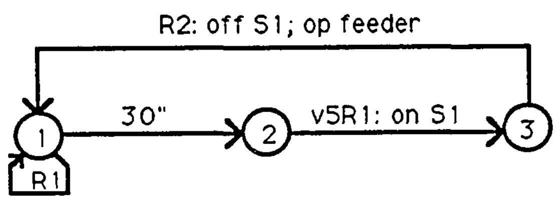
R1 = chain pull
R2 = lever press
R3 = spot touch

S1 = red light
S2 = green light
S3 = yellow light




Page 156

R1 = lever press
S1 = red light
R2 = chain pull



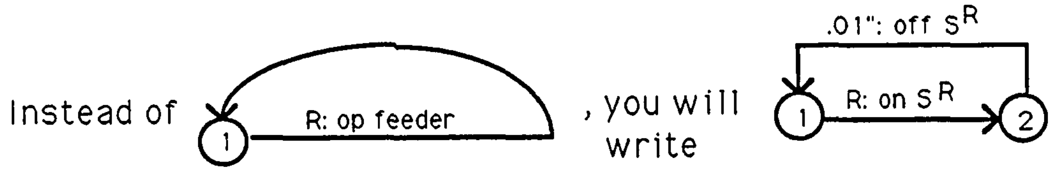
## Section 4
## Diagramming Reinforcement and Punishment Operations

At this point, you know a great deal about state diagramming. However, we have been taking some short cuts in the diagramming of reinforcement to make the task a bit easier for you. You will now learn a method of diagramming reinforcement (and punishment) which is the "standard" for state diagrams. From this point on in the program, you should use this "new" method.

---

This method involves knowing two things:

1) Instead of just writing "op feeder" or "op dipper", you will break reinforcement (and punishment) delivery into TWO separate states. For example:




2) Notice that there is a TIME input in the second state. In some cases, you will have to decide what input goes there. To do this, you will make a decision: "Is the reinforcement (or punishment) LEFT THERE or TAKEN AWAY?"

Turn the page, and you will learn how to decide this!

There are two ways to classify stimuli that are presumed to function as reinforcement:

Reinforcement that is presented and LEFT there (consumed or used later)

Reinforcement that is made available for a period of time and then REMOVED (limited access only)

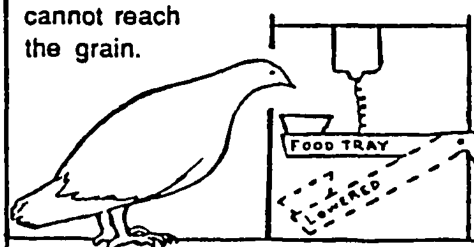
For example:

A food pellet for a rat (delivered with a pellet dispenser)

A pellet dispenser is programmed to drop one food pellet into a cup. The pellet is left there and the rat can walk over and eat it.

Opportunity to eat grain from a grain hopper (pigeon)

The arm of the food tray is raised for the duration of reinforcement delivery. The pigeon can reach the food tray and peck at the grain. When the food tray is lowered, the pigeon cannot reach the grain.




All other types of reinforcement can be classified according to whether they are most like a food pellet for a rat (something that is delivered quickly and LEFT in the presence of the organism), or whether the reinforcement is most like the grain hopper for the pigeon (made available for a specific amount of time and then REMOVED).

Classify these types of reinforcement as LEFT THERE (LT) or REMOVED (R): (Answers p. 166).

1. X number of seconds of music ________

2. A grape for a monkey ________

3. A token (traded later) ________

4. 1 minute of a computer game ________

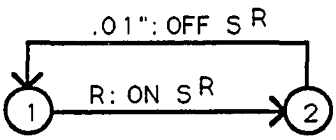
5. A ride around the block in a Corvette ________

This is the way to draw the type of reinforcement that is LEFT there:

This is the way to draw the type of reinforcement that is REMOVED:

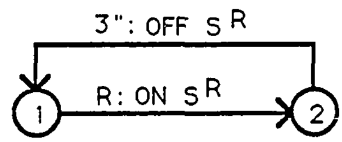
---

For example:
A food pellet for a rat:




.01" is just long enough for the computer to send a brief pulse which operates the feeder.
Use .01" as a standard for all types of reinforcement that are "LEFT there".

For example:
Opportunity to eat from a grain hopper (pigeon)




3" is typically used with grain hoppers. Of course, the time can vary depending on the type of reinforcement.

---

Draw the appropriate way of designating reinforcement:

* (When diagramming stimulus changes that are NOT classified as unconditioned reinforcement, write out the name of the object or event).

1. A token (the teacher drops a token into a cup after the child does one math problem correctly)

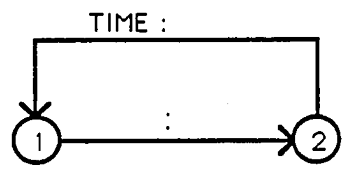
   * (use "ON token" and "OFF token")

2. The child may play a computer game for 3 minutes after he does 10 math problems correctly).

   * (use "ON game" and "OFF game")

A standard way to diagram all types of events that are expected to function as punishment is to use a duration of .01". This standard comes from the use of shock, in which only a brief pulse is delivered. Of course, when the event is to last for a specified amount of time (i.e. 3 seconds of a loud noise), then specify the amount of time in seconds).

Draw the diagrams: (Remember the format is




)

a. A teacher yells a loud "NO!" when a child hits another child. The duration of the "NO" is instantaneous. (Use "on NO!")

b. Shock is delivered when a rat presses a lever. (Use "on SK")

c. Shock is delivered at variable one minute intervals. (Use "on SK")

Now you are ready to practice diagramming using the methods that you have just learned.
You may wish to use any of the preceding materials as job aids.

---

a. First, a 1 minute timer times out. Then, the pigeon only has 5 seconds to make a key peck, which will be followed by 3 seconds of grain. If he does not make the key peck within 5 seconds, the 1 minute timer resets.

---

b. The session begins with timing out of a 6 second timer, after which a red light comes on. Then, 5 seconds later, a green light comes on. Now, when a 10 second timer times out, a food pellet is delivered (to a rat), the lights are turned off, and the procedure repeats. (Note: the rat does not have to make a response in this procedure)

---

c. When the rat presses the lever (R1) 10 times, a food pellet is delivered. Now, if the rat presses the lever again, the procedure simply goes back to State one and begins again. HOWEVER, if the rat pulls the chain (R2), he receives a brief pulse of shock, and the procedure returns to State one. (Hint): State 3 has 2 inputs.

---

Try a few more diagrams, this time with humans.

---

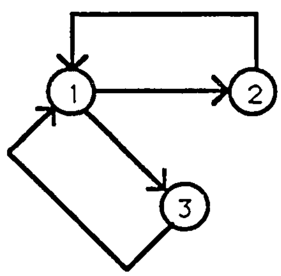
a. When the child goes 10 minutes without talking out in class (talking out=R), then he gets a token. If the child DOES talk out in class, the 10 minute timer resets.

---

b. When the child completes 10 math problems in a row correctly (1 problem=1 response), then he gets an M&M). Then, when he does 5 math problems in a row correctly (1 math problem=1 response), he listens to a record for 5 minutes. The procedure repeats.

---

c. (Here is a skeleton diagram; you fill in the inputs and outputs). If a child can go 5 minutes without hitting another child, she gets 1 minute of sitting on an adult's lap. If she hits another child (R), she is told "NO!" in a loud voice. (The "NO" is instantaneous). The procedure repeats.




---

Now see if you can draw this diagram without the use of any job aids.

---

The procedure starts with a light on. Now, after a response is made, 3 seconds of grain are delivered, the light is turned off, and a tone is turned on (the stimuli remain on during the presentation of the grain). Now, if 5 seconds passes WITHOUT a response being made, 3 seconds of grain are delivered, the tone is turned off, the light is turned on, and the procedure begins again at State one. If a response IS made within 5 seconds, a brief pulse of shock is presented, the tone is turned off and the light is turned on, and the procedure begins again at State one.

## Answers - Section 4

Page 160 1. R

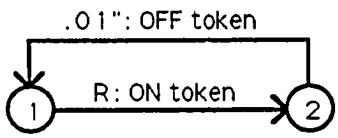
2. LT

3. LT

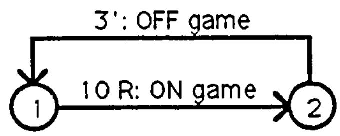
4. R

5. R

Page 161 1.

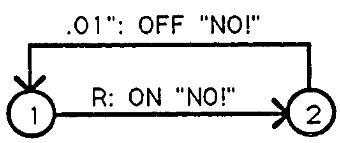



2.

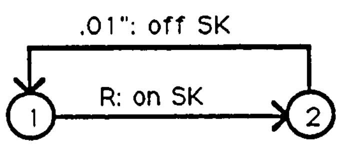



R = 1 correct math problem

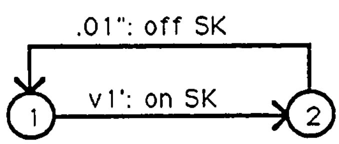
Page 162 a.




b.

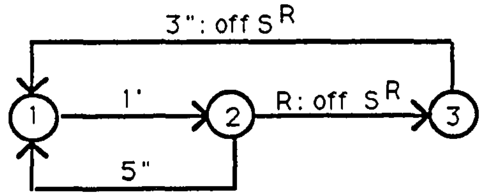



c.

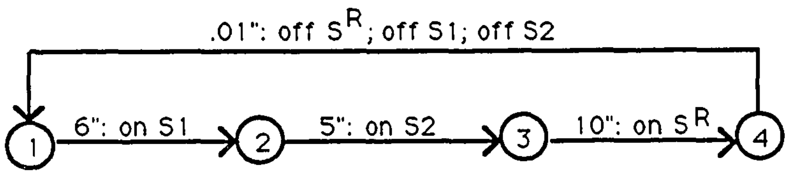


Page
163

a.

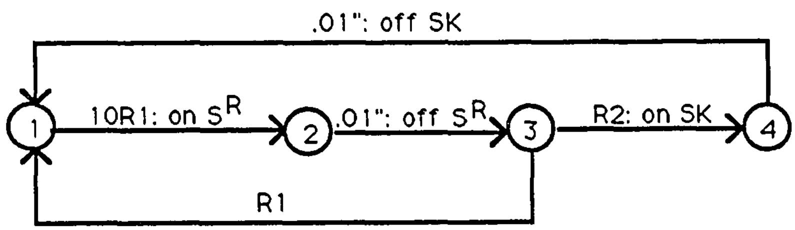



b.




S1 = red light
S2 = green light

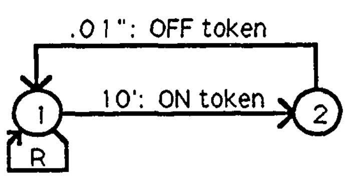
c.



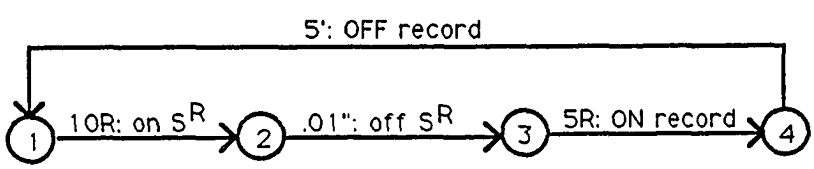

R1 = lever press
R2 = chain pull

Page
164

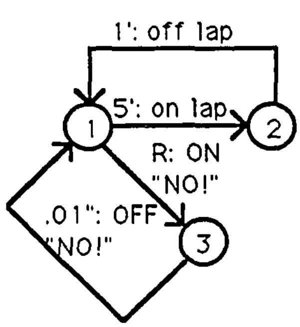
a.



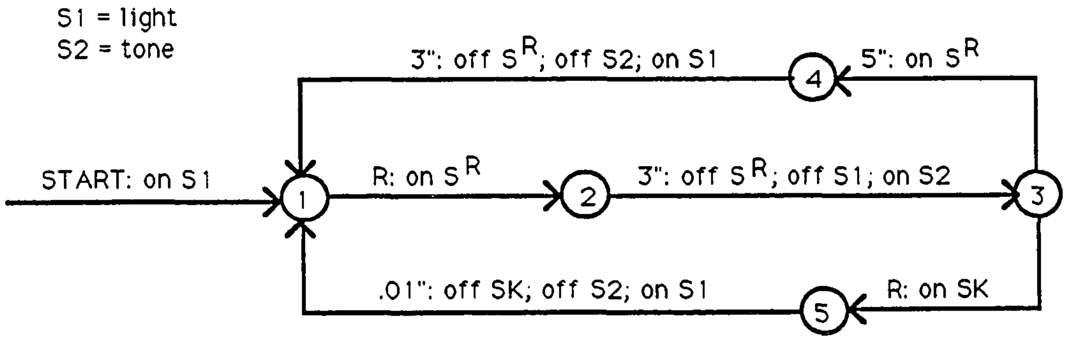

b.




c.

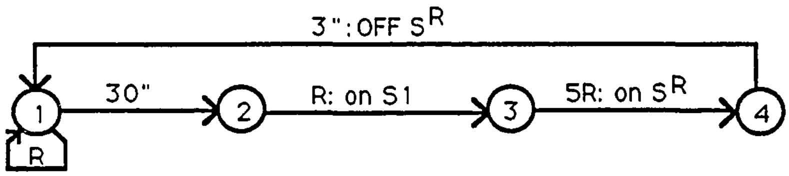


Page
165


S1 = light
S2 = tone
3": off S R; off S2; on S1
5": on S R
START: on S1
R: on S R
3": off S R; off S1; on S2
.01": off SK; off S2; on S1
R: on SK
1
2
3
4
5

## Section 5
## Parallel State Sets

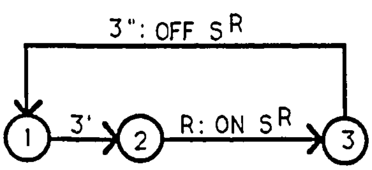
Until now, all of the state diagrams you have been introduced to were single state sets such as this:



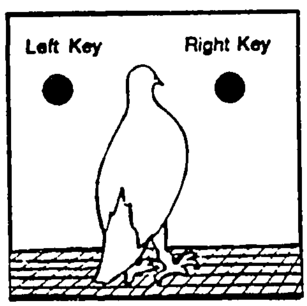

Procedures are often best represented by parallel state sets (two or more state sets, such as in the example below. Each state set is labeled "State set A" (SSA), "State set B" (SSB), etc.

Imagine an operant chamber that looks like this. There are two keys, and each key is programmed with a different procedure. The pigeon can work on either procedure and can switch back and forth between procedures.
These procedures are two types of schedules of reinforcement.

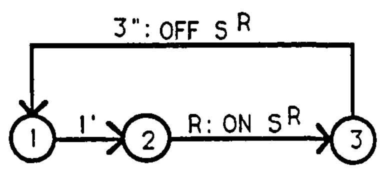
On Left Key:




SSA
(State Set A)




On Right Key:




SSB
(State Set B)

---

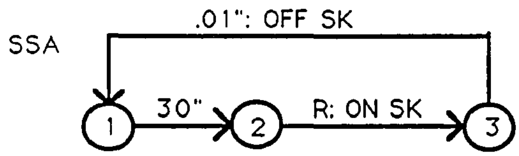
With parallel state sets, the organism is always in one of the states in EACH state set. Suppose that if we looked into the chamber at one instant in time, in which:

a) In SSA the 3' timer is timing and 30" have elapsed;
b) In SSB the grain hopper has just been raised

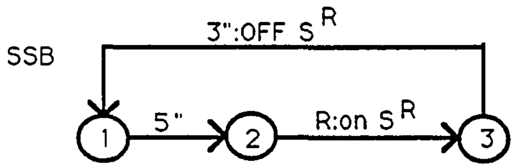
Put an arrow above the state in each state set to show where the procedure is at this instant. (Answersp.175)

---

Here is another example of when a parallel state set would be needed because two timers are operating simultaneously.




After a 30 second timer times out, when a response is made, a brief pulse of shock is delivered.




After a 5 second timer times out, when a response is made, three seconds of reinforcement are delivered.

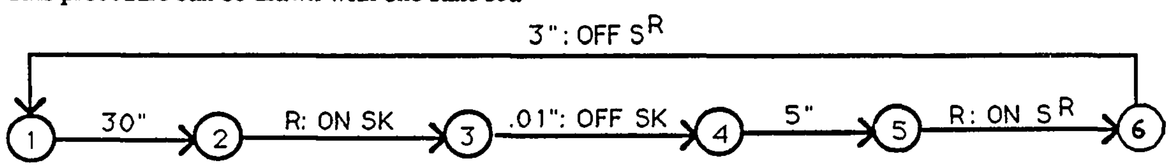
Notice that one timer is controlling the reinforcement schedule, and another timer is controlling the punishment schedule. Both timers are running at the same time. Also notice that if the organism does not make a response and the two timers time out, both state sets will be in State two indefinitely.

Now what happens if the organism makes a response while both state sets are in State two?

a. ______________________________________________________________

______________________________________________________________

Here is a state diagram that shows both a reinforcement and a punishment schedule.
This procedure can be drawn with one state set.

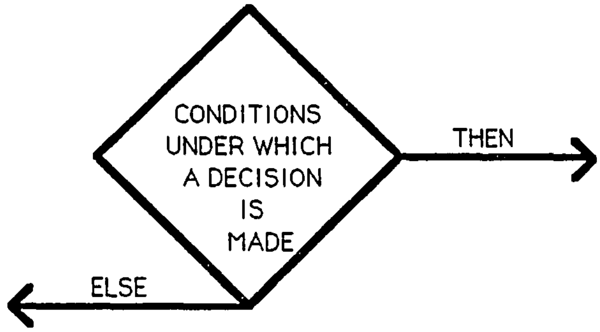



This procedure is very different than the one at the top of the page.
How is it different? (Hint: think about how you answered the question above).

b. ______________________________________________________________

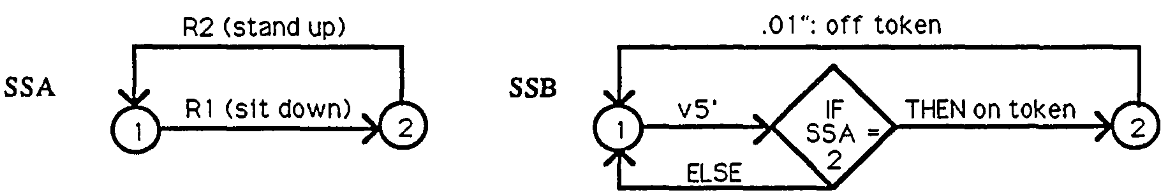
______________________________________________________________

In the examples of parallel state sets that you have already seen, two timers (or two schedules of reinforcement or punishment) were operative at the same time. Now we will see a totally different type of situation in which a parallel state set is needed. For this situation, a symbol called a "decision diamond" is used, meaning that a decision is made about which of two alternatives will occur.




The decision diamond is used in parallel state sets in procedures such as this:

At variable 5 minute intervals., a teacher will check to see whether a child is seated. If the child is seated at the time the teacher looks up, the child receives a token. The teacher resets the timer and the procedure begins again.




Look carefully at State Set A. This is a good time to review the fact that you are always in a state until the events on the transition line occur, and then you immediately go to the next state.

Notice that the decision diamond says "If SSA = 2".
What is the child doing if SSA = 2?

a. ______________________________

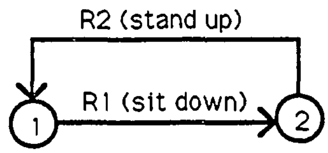
What happens if the child makes R2?

b. ______________________________

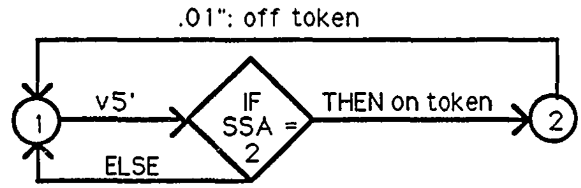
Draw an arrow above the state in SSA in which conditions are right for reinforcement to be delivered.

Here is the procedure again:

SSA

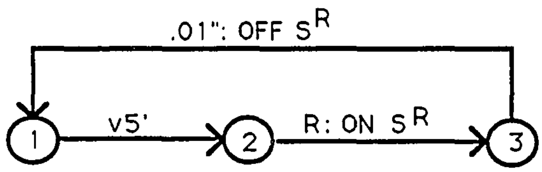



SSB




---

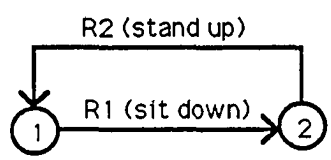
Here is an attempt to draw the same procedure with one state set.




A verbal description of the procedure drawn with one state set is: After a variable 5 minute period, if a response is made (in this case, the child sits down), reinforcement is delivered. (The teacher will look up after the timer times out. The child can then sit down [make R] and get a token).

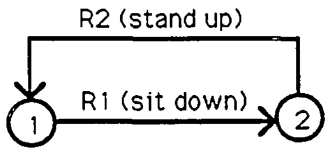
In the procedure drawn with parallel state sets, notice that the child has to be sitting before reinforcement can be delivered. The state diagram drawn with one state set has no way of monitoring whether the child is sitting before State 2 is in effect. That's why the response must be divided into two components (sit down and stand up). Of the parallel state sets, one state set is there just to monitor his responses! (see below).

SSA

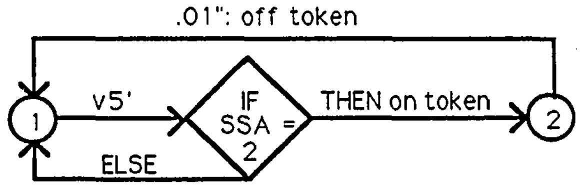



Now look at the procedure as drawn with one state set. Explain why the procedure as drawn with one state set would not effectively change the child's behavior, and explain how the procedure as drawn with parallel state sets corrects the problem!

SSA



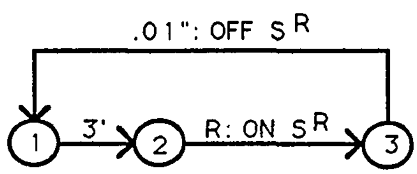
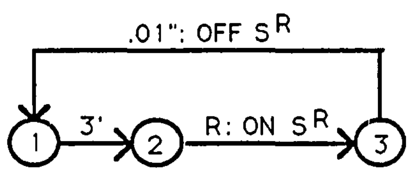
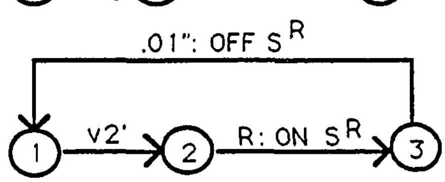
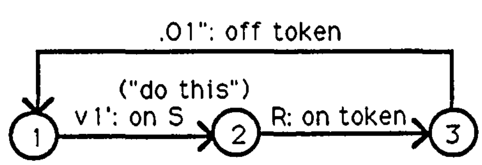

SSB




The procedure drawn above needs a parallel state set because the response of "sitting" is more accurately depicted by dividing the behavior into two parts: "sitting down" and "standing up". In the laboratory, behaviors usually are discrete events (i.e. pressing a lever or pecking a key). However, in this example, the teacher is not interested in whether the child's buttocks can make contact with the seat of the chair (as in a lever press)! The teacher would like to increase the duration with which the child remains seated, and SSA above is better suited to behaviors which involve duration.

---

These situations have been drawn with one state set. You must decide whether they have been drawn correctly or whether they need a parallel state set. To help you make your decision, decide whether the behaviors are discrete events, or whether they involve duration.

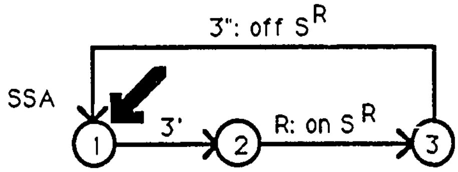
| Situation: | Diagram: | Needs a parallel state set? |
|---|---|---|
| a. If a rat is HOLDING a lever when a three minute timer times out, a food pellet is delivered. This procedure repeats. |  | Yes____ No____ |
| b. When a rat presses a lever every three minutes, a food pellet is delivered. |  | Yes____ No____ |
| c. If a child is lying on his cot (at naptime) at variable 2 minute intervals, he gets a piece of cookie. |  | Yes____ No____ |
| d. If a child imitates a motor movement whenever the teacher says "Do this", the child gets a token. |  | Yes____ No____ |

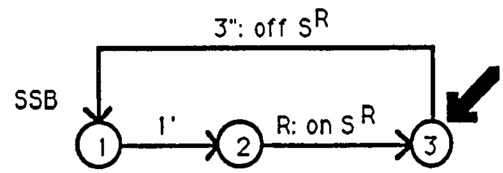
These are descriptions of procedures that you should be familiar with. You are to determine whether a single or parallel state set is needed. Read the verbal description and place a check in the appropriate column.

| Description of Procedure | Is a parallel state set needed? Yes | No |
|---|---|---|
| a. Two schedules are in effect at the same time: The first response to follow timing out of a 15" timer results in 3" of grain, and the first response to follow timing out of a 1' timer results in a brief pulse of shock. | | |
| b. Both of these events must occur before reinforcement is delivered. First, the pigeon must make a response when a 30" timer times out. Then, after 1' passes, the pigeon must make 60 key pecks. | | |
| c. A pigeon is in a chamber with two keys. He can work on either key. The first key is programmed so that he gets 3" of grain after each 10 responses. The second key is programmed so that the first response made after 10 seconds results in 3 seconds of grain. | | |
| d. The teacher will glance up at Johnny every 5 minutes. When he is in his seat, he will get a check mark. When he is not in his seat, he doesn't get a check mark and the teacher will look at him 5 minutes later. | | |

## Answers - Section 5

Page 169



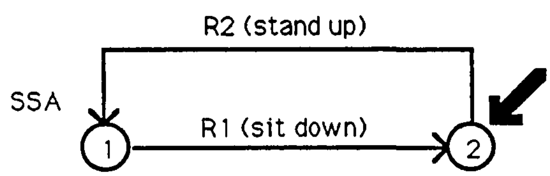

(You stay in State 1 until the 3' timer times out - then go immediately to State 2)

Page 170 **a.** A brief pulse of shock AND three seconds of reinforcement are delivered.

**b.** Reinforcement and shock can never be delivered at the same time. The schedules are sequential - not concurrent.

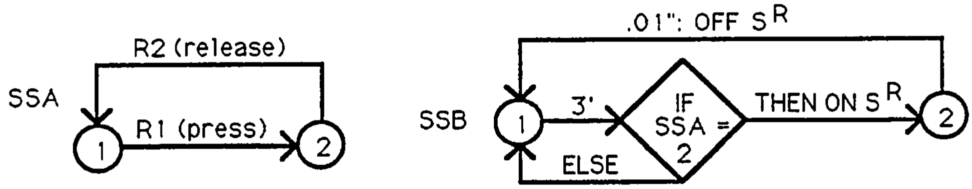
Page 171 **a.** The child is sitting.

**b.** SSA goes immediately to State one.



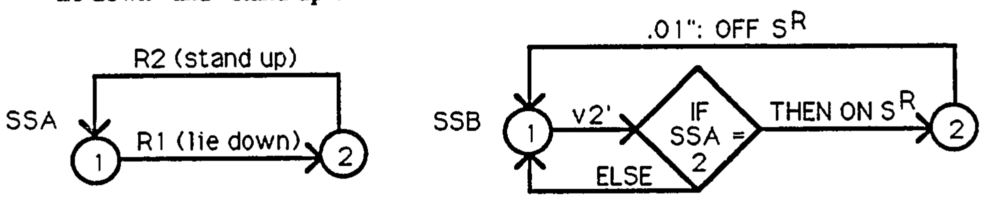

Page 172 The procedure as drawn allows the child to stand up most of the time, sit down ONLY when the teacher looks up, and still get a token! With the parallel state set procedure, the child is only reinforced when he's already sitting.

Page 173 **a.** Yes - because the holding response should be divided into "pressing the lever" and "releasing the lever". Holding a lever is like sitting.



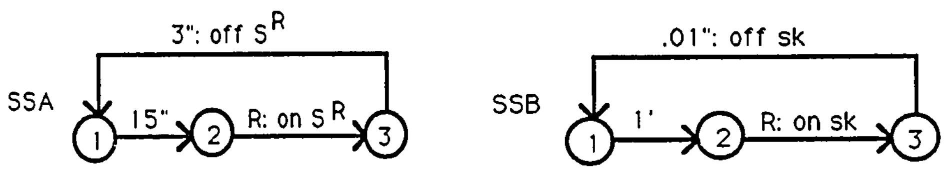

**b.** No

**c.** Yes - because the lying response should be divided into "lie down" and "stand up".

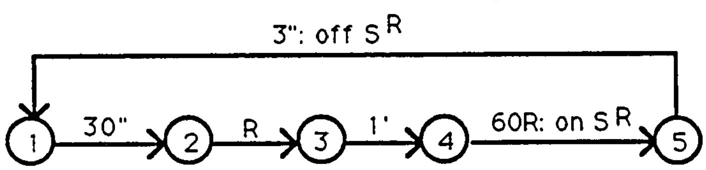



**d.** No

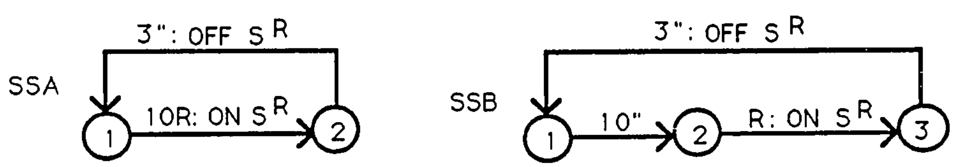
Page 174 **a.** Yes



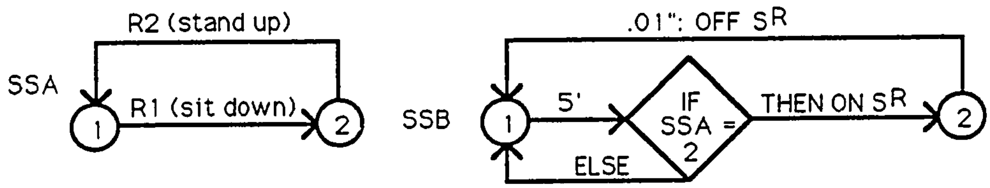

**b.** No




**c.** Yes

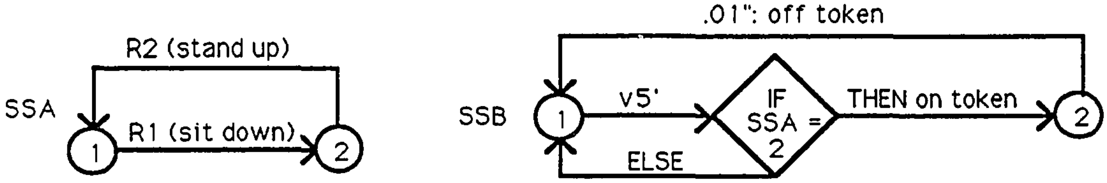



**d.** Yes



## Section 6
## Decision Diamond and Z Pulses

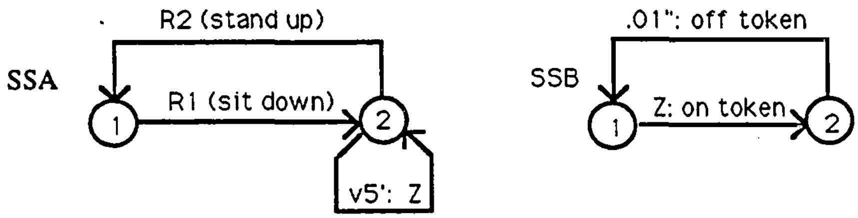
Earlier, you saw a procedure like this:
(Diagram 1)




This procedure required a parallel state set because the response of "sitting" was more accurately represented by dividing the behavior into two parts: "sitting down" and "standing up". However, this procedure only insures that the child is sitting when the teacher looks up - not for the entire V5' duration.

To diagram a procedure in which a child must REMAIN seated for the entire time period, a Z pulse is needed. Look at State 2 of SSA. Remember that when the child sits down, SSA is in State 2.
As long as the procedure is in State 2, a v5' timer is timing. Each time the timer times out, a Z pulse is sent to SSB. As long as SSA is in State 2, the timer continues to time and z pulses are sent to SSB.

(Diagram 2)

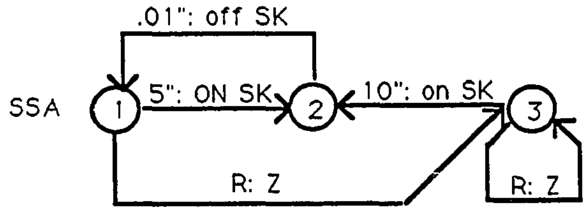


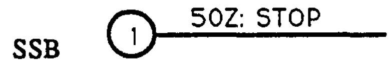

a. What function does the Z pulse serve in SSB? (Answers p. 180).

b. Diagram this procedure. You must decide whether a  or z pulse is needed.

A rat must hold a lever for a variable period of 15 seconds before he receives a drop of water.

a. Here is another way that Z pulses are used. Sometimes an experimenter wants to stop a session when a certain number of responses have been made.



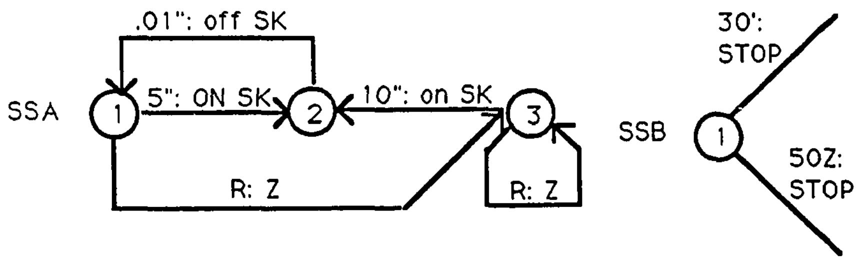




No transition arrow is needed after STOP because the procedure doesn't need to go anywhere. The session is over.

Notice how the Z pulse counts each response and "keeps track of it" somewhere. When there are 50 Z pulses made, the session ends.

---

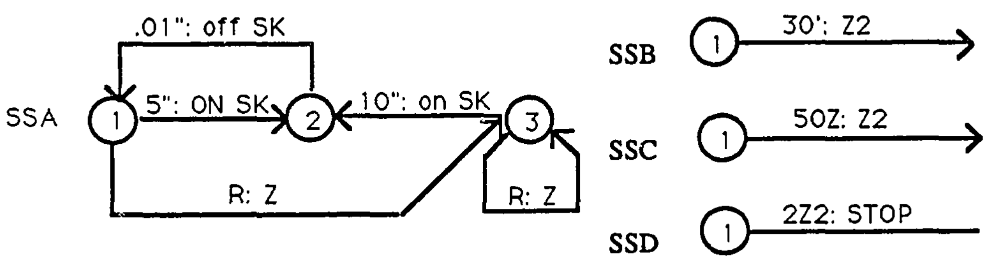
b. Here's another way that Z pulses can be used to stop the session.




Explain State Set B. (You must use the word "or" in your description).

---

c. Here's ANOTHER way that Z pulses can stop a session!

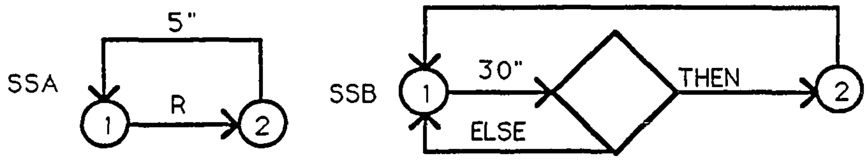



Explain how the Z pulses stop the session. Use the word "and" in your description!

You are now familiar with various types of parallel state sets that interact. For practice, here are some procedures that use either decision diamonds or Z pulses to interact. Fill in the missing inputs and outputs.

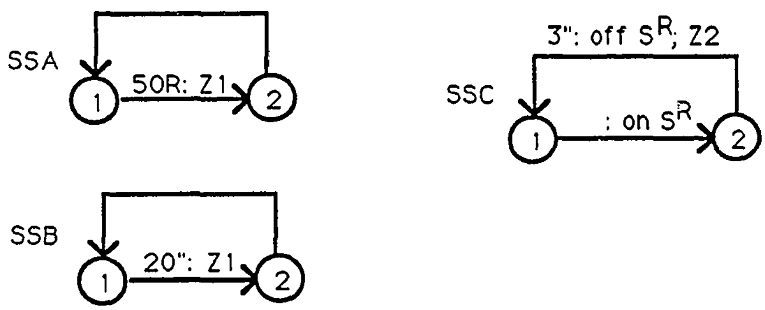
---

a. Reinforcement will be delivered when no more than 5 seconds have passed since R has been made.



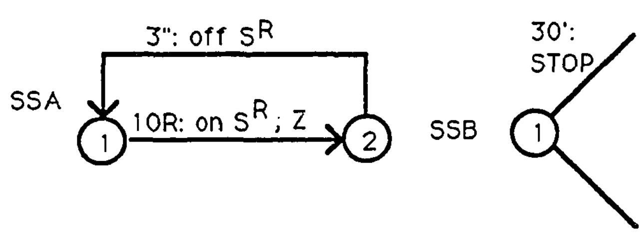

---

b. When 50 responses AND 20 seconds have passed, reinforcement will be delivered. This procedure uses Z pulses in a slightly different way. Notice that two different Z pulses are used: Z1 and Z2. What does Z2 do?



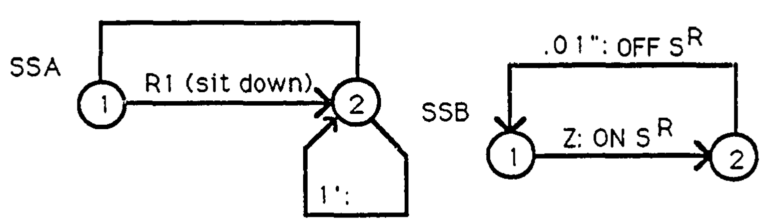

---

c. The session will stop after either 30 minutes or 30 lever presses.




---

d. Johnny must be seated for 1 continuous minute to receive an M&M.

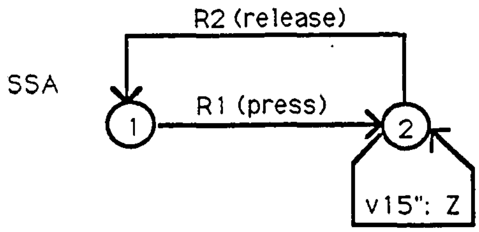


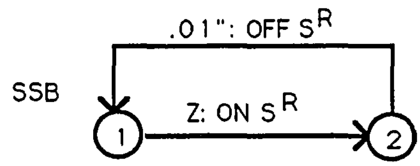

---

## Answers - Section 6

Page 177 **a.** The Z pulse in SSB is an input which is followed by reinforcement delivery. The Z pulse output in SSA goes to SSB, and sends a "signal" that reinforcement delivery should occur.

**b.**

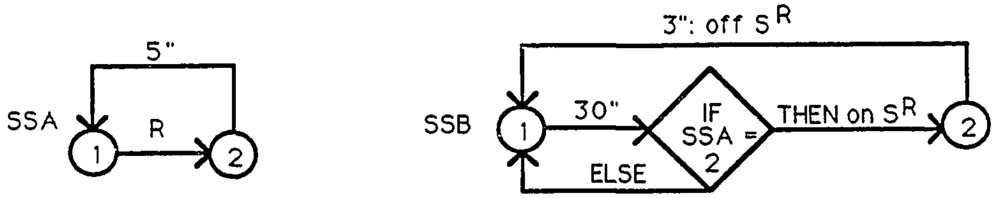


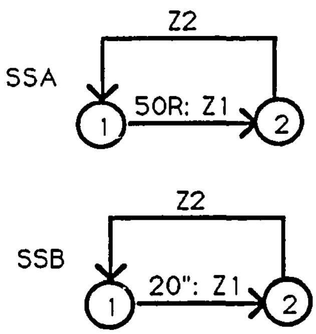

Page 178 **b.** 30 minutes OR 50 responses, whichever happens first, will cause the session to stop.
(Each response made in SSA produces a Z pulse. In SSB, when 50 Z's are accumulated, OR when the 30' timer times out, State 2 is entered and the procedure stops.

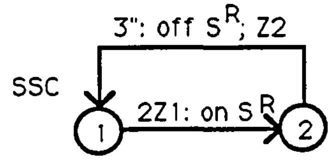
**c.** In SSB, when the 30 minute timer times out, a Z2 is produced. Also, in SSC, when 50 Z's (responses) are accumulated, a Z2 is produced. When 30 minutes AND 50 Z's are accumulated (2Z2), the session stops.

Page 179 **a.**



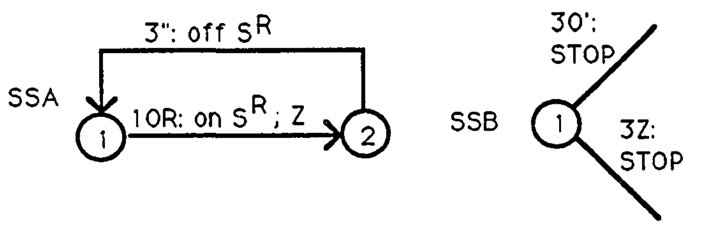

**b.**



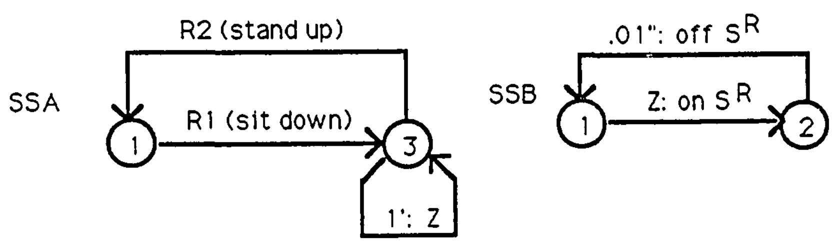

Z2 in SSC causes a transition back to State 1 in SSA and SSB so the procedure can begin again.

**c.**




10 responses = 1Z

**d.**



Appendix L

Human Subjects Institutional Review Board
Letter of Exemption

WMU

Western Michigan University
Kalamazoo, Michigan 49008-3899

*Human Subjects*
*Institutional Review Board*

TO: Esther H. Shafer
Jack Michael

FROM: Ellen Page-Robin, Chair

RE: Research Protocol

DATE: Ocotober 9, 1986

This letter will serve as confirmation that your research protocol, "Development and Validation of Programmed Instructional Materials to Teach State Notation to Psychology Students," has been approved as exempt by the HSIRB, contingent upon your signing the protocol. Please contact either Heather Owner or myself at 383-4917 as soon as possible so that a time can be arranged for you to come in and sign the protocol.

## BIBLIOGRAPHY


Annett, J., & Kay, H. (1956). Skilled performance. Occupational Psychology, 30, 112-117.

Balson, M. (1971). The effect of sequence presentation and operant size on rate and amount of learning. Programmed Learning and Educational Technology: Journal of the Association For Programmed Learning, 8 (3), 202-205.

Brethower, D. M., Markle, D. G., Rummler, G. A., Schrader, A. W., & Smith, D. E. P. (1964). Programmed learning: A practicum. Ann Arbor: Ann Arbor Publishers.

Computer Programmer Aptitude Battery (1967). Chicago: Science Research Associates.

Cox, J. A., & Boren, L. M. (1965). A study of backward chaining. Journal of Educational Psychology, 56, 270-274.

Davies, I. K. (1967). Frame writing: An algorithm for decision making. National Society for Programmed Instruction Journal, 6, 15-18.

Davies, I. K. (1972). Presentation strategies. In Hartley, J. (Ed.), Strategies for programmed instruction: An educational technology (pp. 82-132). London: Butterworth.

DeCecco, J. P. (1964). Educational technology: Readings in programmed instruction. New York: Holt, Rinehart, and Winston.

Duncan, K. (1972). Strategies for analysis of the task. In Hartley, J. (Ed.), Strategies for programmed instruction: An educational technology (pp. 19-81). London: Butterworth.

Espich, J. E., & Williams, B. (1967). Developing programmed instructional materials. Palo Alto: Fearon Press.

Gagne, R. M. (1965). The analysis of instructional objectives for the design of instruction. In Glaser, R. (Ed.), Teaching machines and programmed learning, Vol. 2 (p. 21-65). Washington DC: National Education Association.

Gilbert, T. F. (1962a). Mathetics: The technology of education. Journal of Mathetics, 1(1), 7-73.

Gilbert, T. F. (1962b). Mathetics: II. The design of teaching exercises. Journal of Mathetics, 1(2), 7-42.

Gilbert, T. F. (1978). Guiding worthy performance. Improving Human Performance Quarterly, 7(3), 273-302.

Hartley, J. (1972). Introduction. In Hartley, J. (Ed.), Strategies for programmed instruction: An educational technology (pp. 1-18). London: Butterworths.

Holland, J. L. (1960). Teaching machines: An application of principles from the laboratory. In Lumsdaine, A. A., and Glaser, R. (Eds.), Teaching machines and programmed learning (pp. 215-228). Washington DC: National Education Association.

Holland, J. L., & Skinner, B. F. (1961). The analysis of behavior. New York, McGraw Hill.

Huitema, B. E. (1980). The analysis of covariance and alternatives. New York: John Wiley.

Hunt, G. J. F. (1972). A comparative analysis of the effectiveness of linear and mathetics instructional programming techniques. New Zealand Journal of Education, 7, 172-180.

Lewis, B. N., & Horabin, I. S. (1977). Algorithmics 1967. Improving Human Performance Quarterly, 6 (2-3), 55-86.

MacDonald-Ross, M. (1969). Programmed learning - A decade of development. International Journal of Man-Machine Studies, 1, 73-100.

Mager, R. F. (1962). Preparing objectives for programmed instruction. San Francisco: Fearon Press.

Markle, S. M. (1967). Empirical testing of programs. In: Lange, P. C. (Ed.): Programed instruction: The sixty-sixth yearbook of the national society for the study of education, Part II (pp. 104-138). Chicago: University of Chicago Press.

Markle, S. M. (1978). Designs for instructional designers. Champaign, IL: Stipes.

Markle, S. M., & Tiemann, P. W. (1970). "Behavioral" analysis of "cognitive" content. Educational Technology, 10, 41-45.

Mechner, F. (1965). Science education and behavioral technology. In: Glaser, R. (Ed.), Teaching Machines and Programmed Learning, II: Data and Directions (pp. 441-507). Washington, DC: National Education Association.

Mechner, F. (1967). Behavioral analysis and instructional sequencing. Programmed Instruction: The sixty-sixth yearbook of the national society for the study of education, Part II (pp. 81-103). Chicago: University of Chicago Press.

Michael, J. L. (1986). State notation for describing behavioral contingencies. In Michael, J. L., Concepts and Principles of Behavior Analysis (pp. 61-91). Unpublished manuscript, Western Michigan University.

Naylor, J. C. (1962). Parameters affecting the relative efficiency of part and whole training methods: A review of the literature. U.S. Naval Training Devices Center, Tech. Rep No. NAVTRADEVCEN-950-1. (AD-275921), New York.

Pressey, S. L. (1960). A machine for automatic teaching of drill material. In Lumsdaine, A. A., & Glaser, R. (Eds.): Teaching machines and programmed learning (pp. 42-46). Washington DC: National Education Association (Original work published 1927).

Resnick, L. B., Wang, M. C., & Kaplan, J. (1973). Task analysis in curriculum design: A hierarchically sequenced introductory mathematics curriculum. Journal of Applied Behavior Analysis, 6, 679-709.

Rummler, G. A. (1965). The economics of lean programming. National Society for Programmed Instruction Journal, 4, 8-10.

Sheffield, F. D. (1961). Theoretical considerations in the learning of complex sequential tasks from demonstration and practice. In A. A. Lumsdaine (Ed.), Student response in programmed instruction. Washington, DC : National Research Council, Publication No. 943. 117-131.

Skinner, B. F. (1957). Verbal behavior. New York: Appleton-Century-Crofts.

Skinner, B. F. (1968a). Teaching machines. In Skinner, B. F., The technology of teaching. New York: Appleton-Century-Crofts. (Original work published 1958).

Skinner, B. F. (1968b). The science of learning and the art of teaching. In B. F. Skinner, The technology of teaching. New York: Appleton-Century-Crofts. (Original work published 1954).

Skinner, B. F. (1968c). A review of teaching. In B. F. Skinner, The technology of teaching. New York: Appleton-Century-Crofts.

Snapper, A. G., Kadden, R. M., & Inglis, G. B. (1982). State notation of behavioral procedures. Behavior Research Methods and Instrumentation, 14, 329-342.

Stammers, R. B. (1982). Part and whole practice in training for procedural tasks. Human Learning, 1, 185-207.

Tiemann, P. W. (1977). Analyzing the performance of sequential tasks. Improving Human Performance Quarterly, 6 (2-3), 113-129.

Tiemann, P. W., & Markle, S. M. (1978). Analyzing instructional content: A guide to instruction and evaluation. Champaign, IL: Stipes.

Wilcox, B. (1974). The teaching of serial tasks using chaining strategies. British Journal of Educational Psychology, 44, 175-183.