

PROCEEDINGS

DECUS Conference

The following papers by A. G. Snapper, S. Stadler, and J. Markowitz were presented at the 1969 Spring Meetings of the Digital Equipment Computer Users Society (DECUS), Psychology Session, held at Wakefield, Mass. Bernard Weiss served as session chairman. The papers are reproduced here through the cooperation of the Society, A. J. Cossette, Executive Secretary, and S. E. Bascom, Biomedical Applications, Digital Equipment Corporation, Maynard, Mass.

A multi-purpose logic module for behavioral experiments¹

ARTHUR G. SNAPPER, *FDR VETERANS ADMINISTRATION HOSPITAL, Montrose, New York 10548*, and JULIUS Z. KNAPP, *RESEARCH LABORATORIES, SCHERING CORPORATION, Bloomfield, New Jersey 07003*

A notational system has been developed that can be used to describe most reinforcement schedules or other sequential procedures by appropriately interconnecting the basic notational units. The notational language has been applied in the past to develop a user-oriented computer program for behavioral experiments. The present paper describes a digital logic module designed to be isomorphic with the basic unit of the notational language, thus permitting rapid programming of new experiments directly from the notational system. The major advantage of the new module is that it requires no electronic sophistication on the part of the user, since the one module serves as the basic unit of all experiments.

Psychologists often study behavior in the experimental laboratory by establishing a set of rules relating responses (defined by a specified electrical event, e.g., closure of a switch activated by a lever) to certain stimuli (e.g., electric shock, food for a deprived subject, etc.). Typically, a stimulus is contingent upon the emission of a specified count of responses, or a response emitted following a specified delay after the previous response, or some combination of these events. In more complex procedures the intensity or frequency characteristics of a stimulus might be adjusted by the response of the subject.

Historically these procedures have been implemented by electrical or electronic circuits in which a group of relay, vacuum tube, or transistorized logic modules are wired to provide the desired experimental relation between behavior and stimuli. A number of different suppliers have provided these modules, and, with a certain amount of trouble, equipment from different suppliers can be used in the same laboratory. However, it is true that voltage and pulse shape requirements often differ widely among the many types of modules available. Furthermore, most, if not all, of the currently available devices require the psychologist or his technician to become fairly adept in the design of complex sequential circuitry as demanded by the idiosyncrasies of a particular module line (e.g., trailing edge versus inverted logic, relay timing problems, etc.).

Recently the advent of low-cost digital computers has made them available for realizing behavioral experiments (Weiss & Siegel, 1967). The principle advantages of the computer are high speed, flexibility, and the ability to program procedures so complex that they are difficult, if not impossible, to realize using standard logic modules. The major disadvantage is that the hard-pressed psychologist is required to gain yet another skill, that of computer programmer, or to obtain the services of a professional for his laboratory.

As a solution to the programming problem we have proposed a new computer language (Snapper, Knapp, Kushner, & Kadden, 1967; Snapper & Kadden, in press) based on the theory and notation of finite automata (Mealy, 1955; Moore, 1956). We have attempted to show that behavioral procedures are finite sequential systems, and that, as such, can be completely described in terms of the exact arrangement of a set of units called states (Snapper, Kanpp, & Kushner, in press). The computer language, originally written for the PDP-8, provides for programming experimental contingencies in terms of states and their interconnections, and is thus a true psychological programming language.

The present paper extends this concept to logic modules by proposing that construction of a unit state and some associated hardware would permit a psychologist to implement an experiment by interconnecting state modules. This procedure would require only that the experimenter notate his experiment in terms of the state system. Once this had been achieved, wiring of the circuit would follow automatically. A second feature of the system is that modules from any manufacturer would be logically compatible with each other, thus making it possible to switch from one set of modules to another without having to learn a new logical system. Furthermore, properties of the state notational system completely eliminate certain problems that otherwise arise in the design of sequential circuits from conventional modules (i.e., critical races in relay circuitry) thus simplifying programming of experiments.

A brief description of the state notational system follows to clarify the power of the language and to specify the properties of the system that must be provided by any form of digital logic used to implement it. A fuller account may be found in Snapper, Knapp, and Kushner (in press).

States

The basic unit of the notational language is the state. Any behavioral procedure or other sequential control system can be subdivided into a set of units, only one of which describes the procedure at a particular time. For psychology, the state may have a stimulus (presented to the S) associated with it and these stimuli may differ or be the same from state to state. Each state also defines the requirements for state change to occur, if these requirements are met. States are notated by enumerated circles, and associated stimuli are abbreviated as SN (or S^R for reinforcement) to the right of a slash following the state number. Thus, in Fig. 1, 1/S1 in the circle on the left indicates that some stimulus, S1, is in effect during that portion of the experiment in which State 1 is present.

Transitions

State change or transition is notated by an arrow leading from one state to the next. In Fig. 1, State 1 will be replaced by

CRF

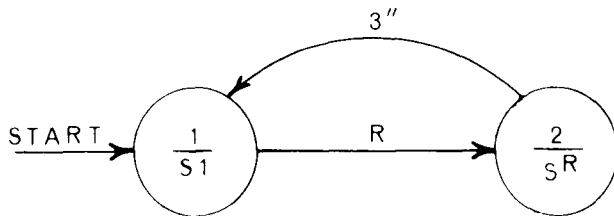


Fig. 1. State diagram of regular reinforcement.

State 2 if a response (R) occurs when State 1 is in effect. Responses and time intervals serve as inputs that may cause state change. The transition from one state to the next is instantaneous (i.e., no other input during a transition can cause a second transition). If more than one transition is drawn from a particular state (e.g., the first state of the lower diagram in Fig. 3) then the input to occur first (20 sec or R in the example), after the state is initiated, will cause a transition.

Outputs

As was noted earlier, each state may have an associated stimulus output that is in effect when the state is in effect. When a transition occurs to the next state, the output or stimulus can be changed. If, however, the stimulus is to remain in two successive states and only their transitions differ, the fact that state change is considered to be instantaneous results in the common stimulus remaining in effect.

The definitions presented here provide the basis for a notation system that can describe any reinforcement schedule. Consider, for example, the diagram of regular reinforcement (CRF) shown in Fig. 1. This diagram describes a procedure that starts with State 1 in effect. Associated with this state is Stimulus S1, which represents the background stimuli of the experimental setting including houselight, etc. If a response occurs in State 1, State 2 immediately replaces State 1 for 3 sec and the associated reinforcing stimulus (S^R) is presented. Three seconds after State 2 is entered, it is replaced by State 1 and conditions are as at the start of the experiment.

A slightly more complex procedure is diagrammed in Fig. 2.

FIXED RATIO

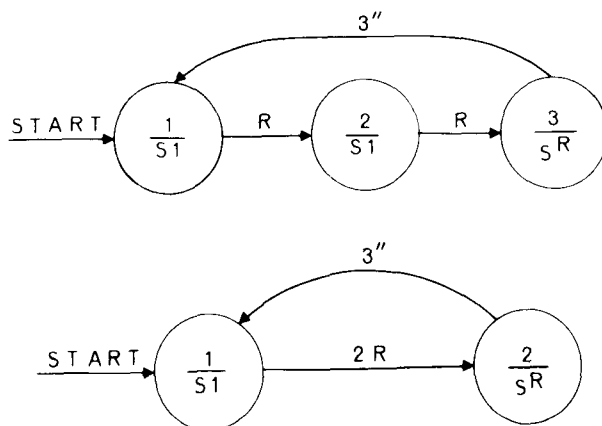


Fig. 2. State diagram of fixed ratio.

FIXED INTERVAL

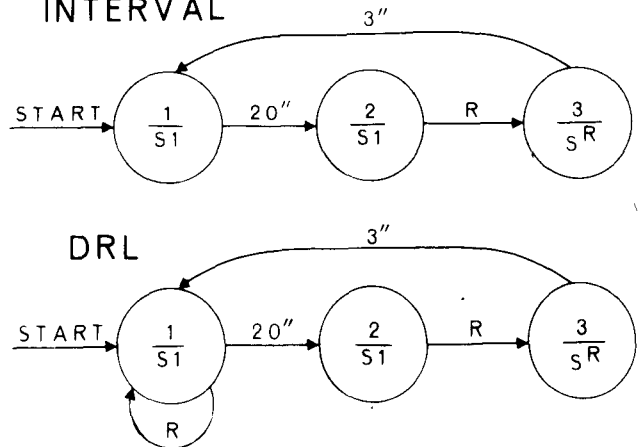


Fig. 3. State diagram of fixed interval and DRL.

The upper drawing, consisting of three sequential states, describes a fixed-ratio schedule (Ferster & Skinner, 1957) with a requirement of two responses for each reinforcement. The experiment starts in State 1 with Stimulus S1 present. A response will cause transition to State 2, but Stimulus S1 will still be in effect. A second response will lead to State 3 with a 3-sec reinforcer replacing S1. Finally, 3 sec after State 3 is entered, State 1 is reinitiated and S1 is again presented to the S. Since fixed-ratio schedules require one more state than the number of responses per reinforcement, it is convenient to define the following higher-order abbreviation:

When a set of sequential states have a common stimulus and the same transition but differ only in the number of transitions required to end the sequence, then the sequence may be expressed as a single state shown to terminate on the *n*th input. In this way, the lower drawing of Fig. 2 is an abbreviation standing for the more complete upper state set. It means that State 1 is replaced by State 2 only when the second response in State 1 is emitted. This abbreviation may be conceptualized as adding a counter of inputs that may be associated with a state. On state entry, the counter is set to zero, and state transition occurs only on the completion of the specified count.

In a similar way, a definition of a resettable clock that may be associated with a state has been developed (Snapper, Knapp, & Kushner, in press). This definition specifies that a timer associated with a state will cause transition after a specified interval following state entry. Thus Fig. 3 shows in the upper diagram a fixed-interval (FI) schedule. Twenty seconds after entry of State 1, State 2 is initiated. Each entry of State 1 restarts the timer so that the full 20 sec will elapse. In the lower drawing of Fig. 3, the diagram of DRL, or differential reinforcement of low rate, is presented. This schedule differs from that of FI only in the consequences of a response in State 1. In the DRL procedure, such a response reinitiates State 1, thus resetting the associated timer.

A slightly more complex schedule is diagrammed in Fig. 4, which illustrates a modified version of a commonly used nondiscriminated avoidance schedule (Sidman, 1953). This procedure starts in State 1. If no response occurs before 20 sec elapse, State 2 with its associated aversive shock (S⁻) is entered. If a response does occur in State 1, its timer is reset and a new 20-sec interval is started. Further responses can keep State 1 in effect and thus avoid shock indefinitely. A response in State 2 turns off shock and initiates State 1. If no response occurs in State 2, State 3 is entered after 3 sec. State 3 stays in effect for

MODIFIED SIDMAN AVOIDANCE

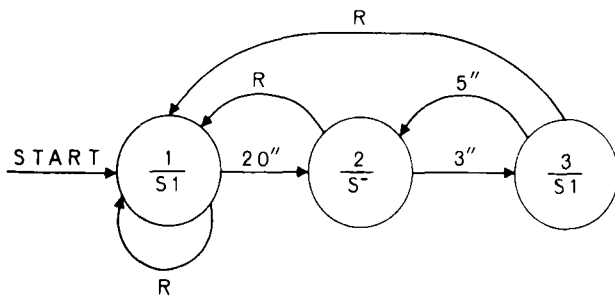


Fig. 4. State diagram of nondiscriminated avoidance.

5 sec if there is no response and then restarts State 2. A response in State 3, however, leads to State 1.

Even more complex procedures can be shown in the state system with more complex diagrams. Other abbreviations can be developed from the basic definitions of the system, and recording of responses can also be diagrammed.

From the basic rules, it follows that a state logic module and associated equipment should, at the very least, provide: (a) that within a state set only one state may be in effect at any one time; (b) that state changes be as fast as possible; (c) that only one transition be allowed to occur at a time; (d) that a timer be provided that resets on state entry; (e) that a counter be developed that is reset on state entry.

A simple relay state module and associated timers, counters, and input circuitry have been developed to illustrate the type of logic module that can be constructed to mimic properties of the state system. Figure 5 shows a simplified schematic of two partially interconnected relay state modules. Each module contains two relays wired to act as flops with separate set and reset inputs. Solid lines represent hard-wiring and dotted lines show external wires representing state transitions. Assume State 1 is in effect. This means that its top relay is locked up through its normally open contacts but that the lower relay has been reset and is off; as in the case of both relays in State 2, a level is provided from the contacts of State 1 to provide stimuli. If a pulse (R) occurs, it passes through the contacts of the closed relay and operates the lower relay of State 2. This relay, in turn, locks up through its own contacts and then simultaneously operates its upper relay and releases the upper relay of State 1. At the end of the response pulse, a clear pulse is generated that releases the lower relay of State 2.

Not shown, but needed in practice, are diodes to isolate the reset lines from each lower relay to the upper relay on each of the other states. Also needed is a wire passing through the normally open contacts of the top relay and the normally closed contacts of the bottom relay that would start a timer that is associated with the state. This latter feature permits reinitiation of the timers by operation of the lower relay of a state module. One must also provide an input interface of relays that sets the duration of the response and timing pulses and prevents responses

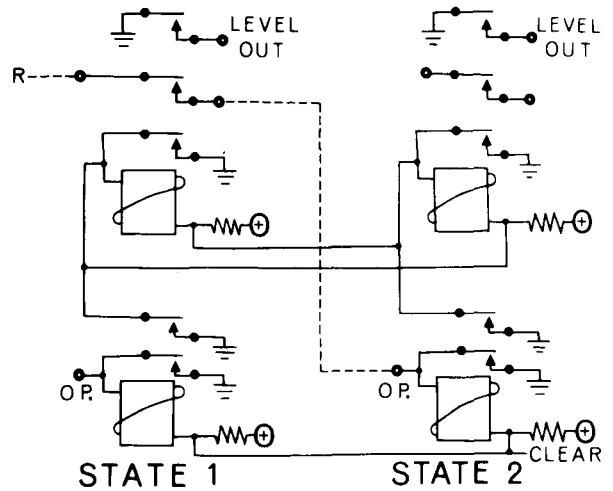


Fig. 5. Schematic of relay state module. External wiring drawn as dotted lines.

and timing pulses from entering the system while transitions are occurring and stores these inputs until the transition is completed. This type of state module has been constructed and has been found to provide programming control for most commonly used reinforcement schedules with three states, three timers, and one predetermined counter at a cost lower than \$200. Similar modules could be constructed at a lower price from transistorized modules that would also offer higher speed and greater flexibility.

REFERENCES

- FERSTER, C. B., & SKINNER, B. F. *Schedules of reinforcement*. New York: Appleton-Century-Crofts, 1957.
- MEALY, G. H. A method for synthesizing sequential circuits. *Bell Systems Technical Journal*, 1955, 34, 1045-1079.
- MOORE, E. F. Gedanken-experiments on sequential machines. In *Automata studies*. Princeton: Princeton University Press, 1956. Pp. 129-153.
- SIDMAN, M. Avoidance conditioning with brief shock and no exteroceptive warning signal. *Science*, 1953, 118, 157-158.
- SNAPPER, A. G., & KADDEN, R. M. Time-sharing in a small computer through the use of a notation system. In B. Weiss (Ed.), *Digital computers in the behavior laboratory*. New York: Appleton-Century-Crofts. In press.
- SNAPPER, A. G., KNAPP, J. Z., & KUSHNER, H. K. Mathematical descriptions of schedules of reinforcement. In W. N. Schoenfeld and J. Farmer (Eds.), *Theories of reinforcement schedules*. New York: Appleton-Century-Crofts. In press.
- SNAPPER, A. G., KNAPP, J. Z., KUSHNER, H. K., & KADDEN, R. M. A notation system and computer program for behavioral experiments. *Digital Equipment Computer Users Society*, New York, 1967.
- WEISS, B., & SIEGEL, L. The laboratory computer in psychophysiology. In C. C. Brown (Ed.), *Methods in psychophysiology*. Baltimore: Williams & Wilkins, 1967. Pp. 459-487.

NOTE

This research was supported by Grant MH-13049 from the National Institute of Mental Health, USPHS, to W. N. Schoenfeld, Queens College, Flushing, New York, and, in part, by the Veterans Administration.