

one recommended by Siegel et al (1971). FOCAL programs are then written to analyze the sorted data in the primary data files on magnetic tape. The results may be stored in files on magnetic tape, printed on the Teletype, or displayed on the oscilloscope.

REFERENCES

- Dyckman, H. L., & Church, R. M. The TIICIE system for interactive control of independent experiments. *Computers in the psychology laboratory*. Vol. II. Maynard, Mass: Digital Equipment Corp., 1972.
Siegel, W., Whittle, K., & Siegel, J. Generating random numbers with Focal. No. 12-61, August 25, 1971, DECUS Program Library.

Use of a notation system for digital control and recording*

ARTHUR G. SNAPPER

Western Michigan University, Kalamazoo, Michigan 49001

This paper describes (1) a notation system for digital control and recording, (2) some improvements to SKED, and (3) future developments and uses of minicomputers.

The main import of this presentation is an appeal to users to develop a standard-process control language that would serve our basic control and recording purposes. The standard language should incorporate provision for adding special-purpose machine language patches for implementing specific requirements, but it should also be as complete as possible. The first section of the paper describes a notation system that could serve as the basis of this language. The second section describes some improvements to SKED, a program for the 4K PDP-8 computer that is based on state notation. And the third section includes a projection of additional tasks a minicomputer could be expected to perform for the research scientist.

THE NOTATIONAL SYSTEM: STATE GRAPHS

The notational language has been designed to characterize any sequential system in terms of inputs, outputs, and states. Although originally developed to aid in the design of computers and other sequential switching networks (Moore, 1956), the state graph notation is easily adapted to completely describe and display the critical features of both behavioral procedures and recording schemes (Snapper, Knapp, & Kushner, 1970). The notation can facilitate communication among scientists concerning the rather complex procedures used in modern psychology. It can be easily demonstrated that the verbal descriptions of reinforcement contingencies typically found in procedure sections of the current literature are often incomplete or imprecise, thus making it difficult, if not impossible, to replicate procedures in exact detail. Although the missing features of the procedures may often be inconsequential, occasionally the descriptions are so incomplete that the main import of the

experiment may be impossible to evaluate without extensive correspondence with the author.

To substantiate this rather strong statement, I would like to report the experience of one of my colleagues (Dr. Jack Michael), who recently conducted a seminar on the experiments to be found in a single issue of *The Journal of the Experimental Analysis of Behavior*. He and his students attempted to notate the procedures of each study from the details of the procedure sections. In many cases, he found at least some procedural ambiguities that forced him to guess at the exact details of the study, and in a few cases he was unable to reconstruct even the critical portions of the experiment from the written description.

I would like to make the case that extensive use of the state notation system would help to eliminate communication difficulties of this sort. The notation makes behavioral procedures and recording arrangements explicit. It is complete in what it attempts to do, that is to formalize statements of the rules by which stimuli of an experiment are modified by time, by responses, or by combinations of both events. Any and all behavioral procedures can be described unambiguously in terms of the notation. Furthermore, the formal representation of the contingencies helps to emphasize critical stages of the procedure with a resulting reduction in the probability of omission of crucial procedural details.

The preceding discussion is a prologue to the rest of this presentation. The notation system has a second purpose: it can serve as a computer language for the control and recording of behavioral experiments. This function of the notation system follows from the fact that it is unambiguous and complete. It is clear that computers can be programmed to translate a formal notational language into the binary code that controls the computer in achieving the desired outcome. In this way, the computer can be used to simplify its own programming.

For example, the well-known FORTRAN language is an unambiguous, complete notation for most mathematical operations. Computers are routinely programmed (usually by the manufacturer) to translate FORTRAN notation into the specific binary format used by that computer to perform the mathematical steps of the program. In this way, computer programming is greatly simplified. Those who use the

*This work was supported in part by Research Scientist Development Award K2-MH-70483 from the National Institute of Mental Health.

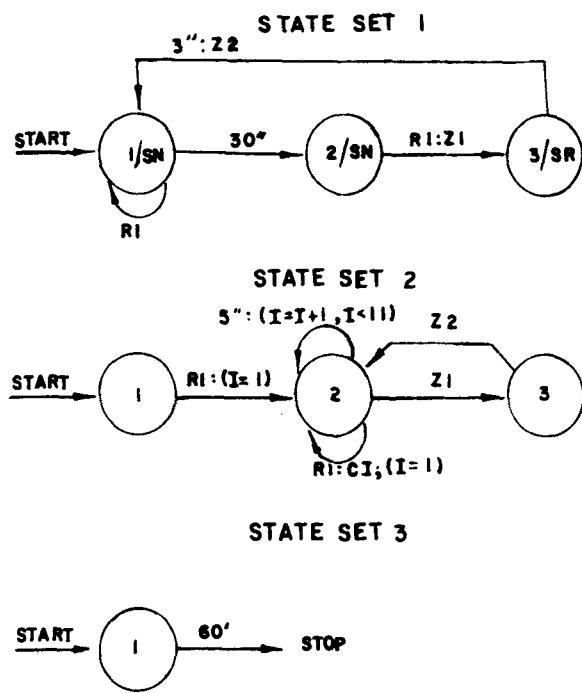


Fig. 1. State graph of DRL with IRT distribution and session timer.

computer as a tool need only learn the FORTRAN language to solve mathematical problems. As modern technology produces new computers, better and cheaper than those of an earlier generation, someone expert in the use of FORTRAN can immediately take advantage of the new systems without extensive retraining. No mathematician is interested in learning the peculiar characteristics of each new machine.

It is the present argument that the time has arrived for the development of a notational language for process control, and in particular for the control and recording of behavioral experiments, so that we can obtain the advantages of standardization that accrue to the users of FORTRAN. We are faced with a large number of minicomputers that vary widely in individual characteristics, but which can be used for our purposes. Those of us who began with 12-bit machines are now faced by the obvious superiority of the newer 16-bit computers, some of which have been specifically designed for process control. Many of the newer machines are now inexpensive enough to compete with the older smaller computers.

It is also apparent that a large amount of duplication of effort is taking place in the field, in that many new systems for behavioral control and recording are reported at each of these meetings. Some of this effort is required to develop a system for use on one of the newer machines. Furthermore, each laboratory has its own peculiar emphasis on one feature or another of process control, so that some work must necessarily be devoted to solving these specific problems. Nevertheless, I feel that the state notation can be used to handle 90% of the

present control applications without modification, and the notation includes rules for the modification needed to handle the other 10%. The use of this notation for a general computer language for process control will simplify the day-to-day use of the computer for most of us. However, the greatest advantages will occur over the long term, since the specification of a standard control language will make the development of the new computer systems a straightforward matter of implementing the notation. At this point, I would like to call for the establishment of a committee to specify the characteristics of a standard control language and to evaluate state notation or any other notational language for the purposes of a general control language for computer usage.

Some Examples of State Graph Notation

For those who are unfamiliar with the state graph notation, I would like to review some of its features. First, an example of a simple, but typical, operant experiment and recording scheme is presented in Fig. 1. The experiment involves a reinforcement schedule named DRL (differential reinforcement of low rates) and an accompanying histogram of an interresponse time distribution. In DRL, the S must wait for a prespecified time between responses for a response to be reinforced. This contingency is shown in the upper state set of Fig. 1. At the start of the experiment, some background stimuli (SN) are presented, as is shown in State 1. The Ss must not respond for 30 sec after entering State 1 for State 2 to be initiated. Responses in State 1 reinstantiate State 1, thus resetting the timer associated with the state. If the S waits 30 sec after entering State 1, State 2 is begun with no correlated stimulus change. In State 2, a response leads to a brief (3-sec) presentation of a reinforcing stimulus, after which a new trial begins. Animals typically develop a good temporal discrimination under this schedule and usually wait a little longer than 30 sec between responses. The usual way to depict the temporal discrimination is to collect a histogram of the time between successive responses. The scheme for accomplishing this is shown in State Set 2 of Fig. 1. This shows that the first response of the session sets the number of Recording Counter I to 1. If another response follows within 5 sec, it is recorded in Counter 1. If 5 sec elapse without a response, the value I is incremented by 1. Each additional 5 sec increments I until either I is at 11 or a response occurs. The response is recorded in the counter with the current value of I, and then I is reset to 1. If a reinforcement occurs, the distribution is suspended until the end of the reinforcement period. The synchronization of the two state sets is accomplished by means of z pulses generated by the upper state set to control the second state set. The third state set merely specifies that the session is 60 min long.

This diagram is a complete representation of a reinforcement schedule and recording scheme of a rather

simple sort. It is complete in that it contains all the information needed to control the experiment by the computer. All that is necessary is to type a representation of this graph on a line-by-line basis on a Teletype to produce the desired procedure.

There are a few changes in format necessary for simplicity of computer operation. The following is a Teletype version of the DRL schedule:

```

/DRL WITH IRT DISTRIBUTION
/ON 1 = HOUSELIGHT
/ON 2 = REINFORCEMENT
/COUNTERS 1 TO 11 FOR IRTS
/R1 = RESPONSE LEVER
/R2 = EXPERIMENTER'S SESSION START FROM TTY

S.S.1,
S1,
    R2:ON 1 --> S2/START SESSION
S2,
    30" --> S3/REINFORCEMENT SET-UP
    R1 --> S2/RESET TIMER
S3,
    R1:ON2;Z1 --> S4/START REINFORCEMENT
S4,
    3":OFF2;Z2 --> S2/START NEW TRIAL

S.S.2,
S1,
    R1 --> S2/START IRT
S2,
    R1:F1(1,1,11) --> S3
    S3,
        S5":F1(1,1,11) --> S3
        R1:C1;F2(1,1) --> S3
        Z1 -----> S4
    S4,
        Z2 -----> S3

S.S.3,
S1,
    R2 --> S2/START SESSION TIMER
S2,
    60" --> STOP/END SESSION AND TURN OFF STIMULI

```

The major differences between this record and Fig. 1 include the addition of an extra state, with S2 being started in each state set by an E typing R2 on the Teletype. Furthermore, each stimulus onset and

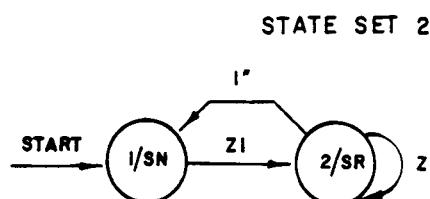
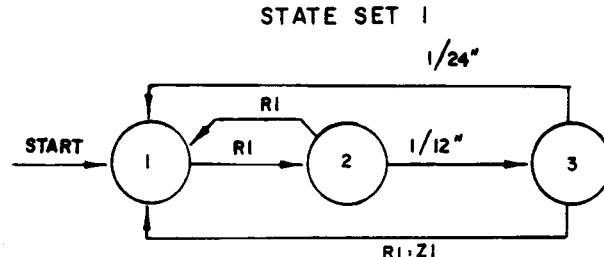


Fig. 2. State graph of alpha conditioning procedure. R1 is a zero crossing of the EEG. SR is the 1-sec reinforcement initiated by each instance of alpha.

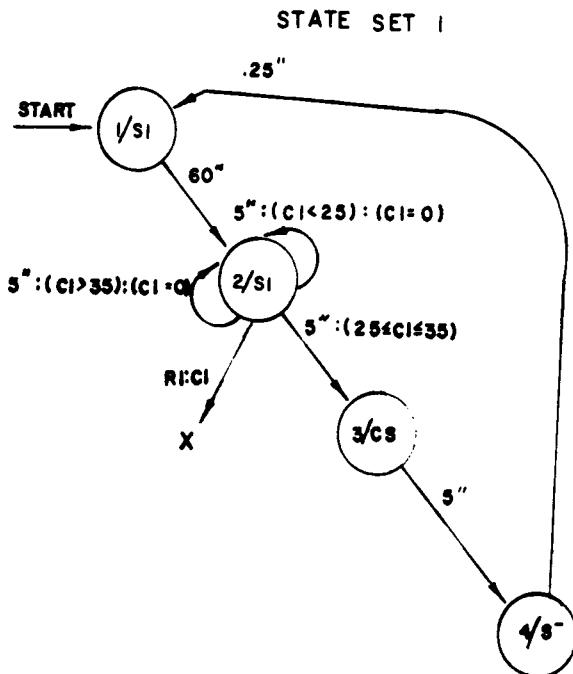


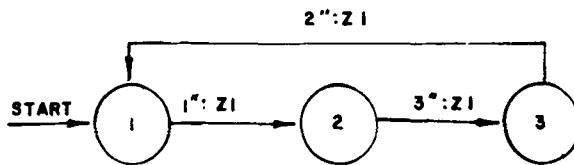
Fig. 3. State graph of cardiac conditioning with trial initiated only if between 25 and 35 heart beats occur at the end of the intertrial interval.

termination must be specified. The F1 and F2 functions trigger the incrementation and initialization of Recording Counter I.

The second example is presented to show some of the flexibility of the system. It is of a crude biofeedback procedure in which a human S is presented with a signal if he produces alpha waves (an EEG wave of 8-12 Hz). In Fig. 2, the zero crossing of an EEG signal is specified as R1. The scheme depends upon an external device to convert the EEG signal to a digitized form. Alternatively, an analog-to-digital converter might be attached to the computer for the same purpose. After the session begins, the first zero crossing starts State 2 with its associated timer. State 3 is entered 1/12 sec later, unless a second zero crossing occurs first. If R1 occurs in State 3, a reinforcement is presented for 1 sec, as is shown in the second state set. If alpha occurs frequently enough, the reinforcing stimulus can be maintained. In actual practice, further, somewhat complex, notation would be required to specify recording of the percent of time spent in alpha.

The third procedure demonstrates how more complex decision making can be integrated into state notation. In Fig. 3, a cardiac-rate conditioning study is shown. In this experiment, a conditioning trial is begun only when a prespecified heart rate has been achieved. One minute after the start of the session, the minimum intertrial interval elapses and State 2 is entered. In State 2, heart beats (R1) are recorded in C1. At the end of 5 sec, a conditioning trial is begun if the number of beats in the 5-sec period lies between 25 and 35. If not, the timer is reinitialized and the counter is set

A

STATE SET I

B

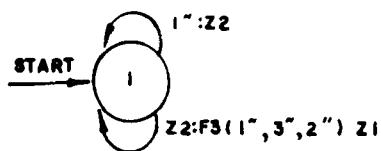
STATE SET I

Fig. 4. State graph of variable intervals. Panel A shows old method and Panel B shows new method.

to 0 for another try. The conditioning trial consists of a 5-sec conditional stimulus (CS) followed by a brief electric shock (S-).

These procedures are very simple examples of state graph notation. They have been selected primarily to demonstrate some common features of procedural variations. A more detailed treatment of the notation may be found elsewhere (Snapper, Knapp, & Kushner, 1970).

IMPROVEMENTS IN SKED

The state graph notational system has been implemented on a 12-bit computer in a series of programs that have been presented earlier (Snapper, Knapp, Kushner, & Kadden, 1967; Snapper & Kadden, in press; Walker & Snapper, 1971). What has changed throughout this series of programs has not been the notation, but rather the strategy of programming and implementation of features that increase the usefulness of the system. In all of these programs a 4K PDP-8 computer has been used with a paper-tape-oriented system. Much thought has been directed toward keeping the system and associated peripheral equipment as inexpensive as possible. The current version in use in my laboratory has a number of features not available in earlier systems.

It is still a two-pass system on a 4K machine. That is, the state graph describing the desired procedure must still be processed by a compiler that utilizes the entire 4K memory bank. That means that program preparation must still take place at a separate time from the actual

running of experiments. The program is also still capable of handling up to 10 independent experiments at one time, and it is still based upon a basic 10-msec clock rate.

New features include a dialogue with the computer to specify assignment of input and output lines to different stations. This permits each station to have access to any combination of input and output lines in the interface. Earlier versions placed an upper limit of 12 lines per station. This restriction has been removed in the new program. The reassignment of input and output lines provides a great deal of flexibility, so that at different times the input-output lines could be divided equally among a number of different stations.

Another new feature of the system has been added to the compiler. This consists of an expansion in the symbolic addressing capability to a second level of variables. As an example, consider the IRT distribution of Fig. 1. The number of the recording counter, CI, is varied by 5-sec pulses from 1 to 11. In some experiments, after a change in the schedule (in a multiple schedule, for example), the same IRT distribution might be desired for a different set of counters, say from 21 to 31. If all that is to be changed are the values of the lower and upper limits of the distribution, in the expressions $R1:CI;(I=1)$ and $5":(I=I+1,I<11)$, they may be compiled as symbolic addresses as follows: $R1:CI;(I=M)$ and $5":(I=I+1,I<N)$. Other statements can then set the values of M and N.

Additional features of the new software include several new subroutines designed to save space in the 4K computer or to provide automatic data output to storage media controlled by the state program rather than by the Teletype commands. These programs depend upon the use of the F3 function that permits special machine language subroutines to be inserted into the state tables. When the run-time system (RTS) is first established by means of the dialogue, it is possible to reserve space for special subroutines. These may then be accessed by an F3 command with a list of arguments following an input.

Of special interest is a table look-up routine to reduce the amount of space required for lists of numbers, as used in the specification of variable intervals or ratios. In past versions of the system, a variable interval might be expressed as a sequence of states, each with its own associated time interval. Each state would enter the next when timed out, and a z pulse might be generated at the end of each interval. Figure 4A shows this sort of arrangement with three successive intervals, each requiring a separate state. In the new F3 routine, the same effect could be achieved by the graph in Fig. 4B. Since this graph requires much less space in the computer, it may be preferred over the earlier method.

A second F3 routine of special interest starts a printout of data onto a selected output device (low- or high-speed paper tape or magnetic cartridge). It may be generated within the state program to provide a record

every 5 min or after 100 reinforcements, for example. The routines possess a queue structure so that requests for outputs will be stored during outputs already in progress. At the end of the first output, the stored requests will then be acted upon. Another feature of the program is a method by which new data can be rejected until the original data has been recorded on tape.

FUTURE DEVELOPMENTS OF LABORATORY COMPUTERS

The final topic I would like to consider is what we can expect to hear at future conferences on computers in psychology. The recent introduction of low-cost 16-bit machines and lower costs of interfacing logic and mass-storage peripherals, including disk memories and/or magnetic cartridges, enlarges my expectations of what economically feasible systems can do for the research scientist. I would expect that for a system price of approximately \$25,000, the price of the original PDP-8 with paper-tape equipment only 7 years ago, a new system today could be constructed to control and record experiments and to serve other functions at the same time. The following list of applications of the new equipment to improve laboratory automation includes functions that are already in use or which could easily be implemented with present technology.

Improvement in the Basic Control and Recording System

I believe that with a reasonable memory size, the compiler could be easily improved to speed up new program development to a matter of minutes rather than hours. Each laboratory would develop and maintain a library of commonly used state tables for such functions as IRT distributions, variable intervals, etc., which often would be included in new experiments in that lab. The new compiler would then permit calling these subroutines by a name specified by the E. It would be a relatively simple matter, then, to include parameter specification in the calling procedures, so that the length of the IRT distribution, for example, could be easily modified at compile time. The assembly of a new program would then be a matter of calling up a number of subroutines and a few modifications of parameters to produce the desired new procedure.

The run-time system could be improved to permit loading of state tables by name also, so that a particular experiment could be assigned to a particular station by a few simple commands. Thus, to initiate a daily session would require only that the station and the compiled study be specified by name. It would even be possible, with enough memory capacity, to program new experiments while other studies were being controlled by the system. Even bigger systems would enable higher level data analysis to be accomplished during the session in which the data was being acquired. An ideal application, for example, would be the decision, based upon stability of a performance in comparison with

previous performance, to administer some variable. Drug administration could be based upon the quality of the sessions in terms of a number of parameters and upon the variation in these parameters from session to session.

Subsidiary Services Performed by the Computer

Since the speed of the new computers is increasing along with the development of inexpensive mass storage and larger memories, the amount of time needed by the RTS to control experiments will be only a small fraction of the total amount of available time. Other tasks could be assigned to the system during experimental sessions, when new program development is not under way. One such task that immediately comes to mind is the automatic production of graphs from the acquired data. An inexpensive plotter attached to the system might be used to plot daily graphs of performance. Furthermore, there should be times at which a good electric typewriter could be attached to the system to provide aid in standard secretarial tasks associated with preparation of manuscripts or research reports.

Information Storage and Retrieval

The minicomputer could also undertake a number of tasks now usually associated with "maxi" systems. With low-cost storage and inexpensive, medium-speed printers now becoming available, each laboratory should be able to develop a personalized information storage and retrieval system that could be used while the minicomputer also controls and records behavioral experiments. Minimally, each scientist should be able to file his own reprint collection according to his own priority system, so that the computer could aid in literature searches undertaken when new experiments are being designed or when reports are being written. An example of this sort of system is the new APA computerized abstract service. Smaller, more individualized, versions would be relatively easily developed for minicomputers.

Computer-Assisted Instruction on the Use of the System

A natural task for the minicomputer system is the training of students in use of the system. Again, the development of larger memories and speedier output peripherals makes the use of the minicomputer as a training device seem quite practical. One station might be reserved for training purposes, so that a program swapped into the computer from a mass-storage device could lead the student into the programming of experiments with the system. The system itself might provide examples and dialogue for this purpose.

I believe that the general import of the recent price reductions in minicomputers and the mass production of peripherals for them leads to the conclusion that minicomputers will be assigned most of the roles recently thought to belong to large time-sharing systems. For most of the needs of the research scientist, a

minicomputer system consisting of one or two heavily implemented computers will provide services that are impossible to obtain from time-sharing terminals. It may be true that for a few specific applications, the number-crunching capabilities of large systems may be essential; but, for almost all applications of computers in the laboratory, it now seems that minicomputers can serve our needs at a low cost.

In summary, I would like to repeat my request for a standardized notational language for process control. I also believe that the state graph system can fulfill this function. It is already being taught to many students of psychology as a communication aid. It could also serve as a general computer programming language.

REFERENCES

- Moore, E. F. *Gedanken-experiments on sequential machines. In Automata studies*. Princeton: Princeton University Press, 1956.
- Snapper, A. G., & Kadden, R. M. Time-sharing in a small computer based on the use of a behavioral 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., & Kadden, R. M. A notation system and computer program for behavioral experiments. Paper presented at the meeting of the Digital Equipment Computer Users Society, New York, June 1967.
- Snapper, A. G., Knapp, J. Z., & Kushner, H. K. Mathematical description of schedules of reinforcement. In W. N. Schoenfeld (Ed.), *Theories of reinforcement schedules*. New York: Appleton-Century-Crofts, 1970. Pp. 247-275.
- Walker, A., & Snapper, A. G. Improvements to the SKED processor central software system. In *DECUS Proceedings, Spring 1971*. Maynard, Mass: Digital Equipment Computer Users Society, 1971. Pp. 7-12.

Discussion of "On-line computers in the animal laboratory"

J. M. NOTTERMAN*

Princeton University, Princeton, New Jersey 08540

There is very little of a substantive nature that I can add to the excellent presentations made by Johnson, Anger, Church, and Snapper, and so I shall not try to amplify on their contributions. Instead, I shall offer some comments addressed to the general topic of this symposium, namely, the on-line use of computers in the animal laboratory. I shall do so with the following key questions in mind: (1) What are the principle objectives of animal research? (2) In what major ways has the on-line use of the computer assisted in achieving these objectives? (3) What is the character of some of the work that remains to be done?

PRINCIPLE OBJECTIVES OF ANIMAL RESEARCH

We all know the answer to the question of why psychologists initially became involved with animal research, but a brief restatement may help to sharpen these remarks.

Watson, in his famous manifesto, indicated the following: First, the use of animals permits us to avoid the perils of mentalistic explanations of human behavior. Second, working with animals in psychological research extends the implications of Darwin's theory of evolution. And third, by experimenting with animals in the laboratory, we are better able to control environmental variables and thereby to satisfy more fully the demands of natural science (Watson, 1913).

These are worthy objectives, but I cannot permit the opportunity to slip by without observing that animal psychology has not done too well by them.

*I thank my research associate, Stephen C. Fowler, for having served as a sample audience. This work was supported by NIMH Grant 18189.

Some of us have been unable to resist the tendency to endow animals with the same mentalistic homunculi that Watson, Skinner, and others were, and still are, trying to avoid in the analysis of human behavior. This serves not only to confound us further, but does violence to a violence.

We do not yet have a broadly conceived animal psychology; the adjective is altogether too often employed as a euphemism for rat psychology.

Lastly, the need to specify carefully environmental variables has led to an overly restrictive view of the ways in which an organism is capable of interacting with the environment. We tend to ignore the rich variety of options available to the animal.

ACHIEVEMENT OF ANIMAL RESEARCH OBJECTIVES THROUGH USE OF THE ON-LINE COMPUTER

However, to the extent that the objectives of animal research have been attained, and by means of on-line computers, the following favorable appraisal may be made: First, the use of computer-programmed reinforcement contingencies assists the scientist in describing complex forms of behavior. Second, the time-sharing feature of the digital computer makes it easier to gather data from several experiments simultaneously, and consequently to decrease the burden of routine laboratory chores. Third, the data storage and retrieval functions of the on-line computer enable the scientist to make informed decisions rapidly, and this is especially important when the outcome of one phase of an experiment influences the procedure selected for the ensuing phase. And fourth, the on-line computer delivers information both to itself and to the scientist with speed and accuracy unmatched by that of the human operator.

But the very same features often are abused and,