15-STATE UNIVERSAL TURING MACHINE

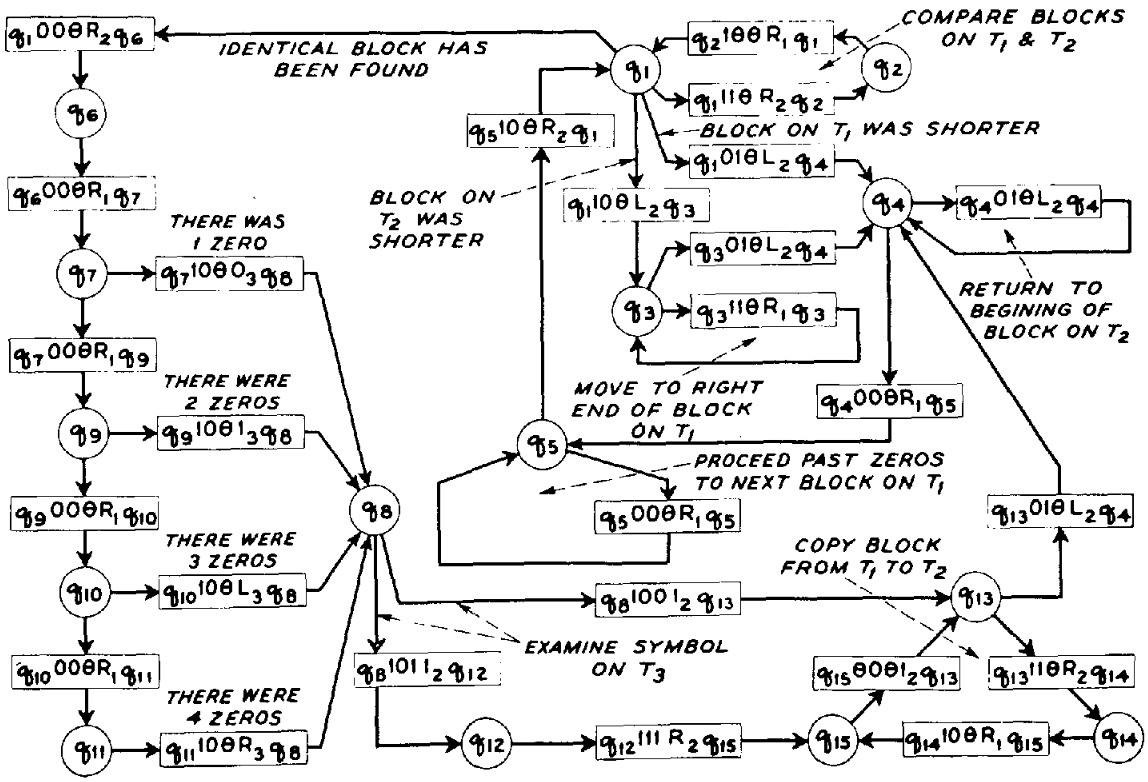

Figure 1

SIMPLE LEARNING BY A DIGITAL COMPUTER

By

The Computation Laboratory
Harvard University

1. Introduction

Digital computers can readily be programmed to exhibit modes of behavior which are usually associated only with the nervous systems of living organisms. This paper describes a concrete example of one practical technique by which the Electronic Delay Storage Automatic Calculator (EDSAC) of the University Mathematical Laboratory, Cambridge, was made capable of modifying its behavior on the basis of experience acquired in the course of operation.

Techniques of this type may have some value for those who, like psychologists and neurophysiologists, are interested in the potentialities of existing digital computers as models of the structure and of the functions of animal nervous systems. The description will be given in two stages. In the first stage (Section 2) the behavior of the EDSAC under the control of a *response learning program* will be presented from the point of view of an experimenter who can control the input of the machine and observe its output, but who is denied access to its internal mechanism. This point of view corresponds to that of an experimenter who attempts to deduce the structure and the internal mode of operation of an animal organism from controlled observations of its functions. In the second stage (Section 3), the factors determining the behavior of the machine are revealed, and are analyzed from the privileged point of view of the designer of the learning program.

2. The Functions of the Response Learning Machine

When the response learning program is introduced into the EDSAC, this machine is changed from a general purpose digital computer into a special purpose machine which will be called the response learning machine. An experimenter asked to describe the behavior of this response learning machine would soon observe that the machine has a sensory device, the input mechanism, which is capable of detecting a stimulus in the form of a number whose magnitude corresponds to intensity. He would note that when such a stimulus $s_t > 0$ is applied at time t, it initiates at random one of a set of possible responses R_i, $i = 1,2,\ldots,5$. The machine signals the occurrence of the response R_i by printing the number i with its output teleprinter. Following the occurrence of a response, the experimenter may express his approval or disapproval. To analyze the machine's behavior in detail, the observer might gather experimental data in a form such as that of Table 1. In this table, s_t is displayed in column 1, the resulting R_i at time t, R_{it}, appears in column 2, and the intensity, a_t, of approval or disapproval is given in column 3. The remaining columns of Table 1 should, for the moment, be disregarded.

At $t = 3$, $s_t = 2$ initiated the response R_2, and at $t = 17$, $s_t = 4$ initiated R_3. An X in column 2 at time t indicates that s_t was too weak to initiate any response whatsoever at that time. In the interval $1 \leq t \leq 12$, $s_t = 2$ is frequently too weak to elicit any response, and those responses that are made occur at random. The experimenter would find it possible to train the machine to give one particular response only, by expressing his approval ($a_t > 0$) whenever this response occurs, and his unconcern ($a_t = 0$) or his disapproval ($a_t < 0$) otherwise. Conversely, a response can be discouraged by repeated disapproval. Table 1 shows that the approval signals $a_t = 2$ given to R_1 at $t = 27$ and $t = 29$. and $a_t = 1$ given at $t = 30$ were sufficient to train the machine to respond to every stimulus with R_1, except at $t = 28$ when, at an early stage of training, R_3 occurred. An earlier attempt, made at $t = 16$, to teach the machine to make the response R_1 failed for reasons whose explanation will be given later.

As the training proceeds, errors become less frequent, and the learned responses may be initiated by a progressively weaker stimulus. The attempt to teach the response learning machine to give the response R_1 begins at $t = 27$ with the stimulus 3; at $t = 30$ the stimulus 2 was tried and found to be sufficient and at $t = 33$, R_1 occurred when the smallest stimulus, 1, was used. At the same time, the need for approval diminishes. At $t = 27$ and $t = 29$, $a_t = 2$, but $a_t = 0$ at $t = 31$ and at $t = 32$, and R_1 nevertheless is still initiated by the small stimulus 1 at $t = 33$. From $t = 34$ onwards, R_1 is discouraged, until it disappears for the stimulus 1 at $t = 37$. At $t = 38$ even the stimulus 2 will not initiate it, but at $t = 39$ the stimulus 3 brings it back. One last sharp disapproval (-4) finally inhibits it. From $t = 40$ onwards, the same procedure is repeated with R_4. Note how at $t = 42$, a premature attempt to reduce the stimulus from 3 to 2 produced no response at all.

It has already been mentioned that a response may be learned by the machine if encouraged by the experimenter, but if the experimenter is neutral and expresses unconcern ($a_t = 0$) for every response, it is nevertheless still possible for some particular response to occur more and more frequently. Eventually, occurring to the exclusion of all others, this response becomes a habit. The high frequency of R_3 from $t = 15$ onwards is due to this effect. To train the machine to give R_1, it was first necessary actively to discourage R_3, which showed promise of becoming a habit. At $t = 21$ the stimulus was reduced to 1 to test how strong a habit R_3 had become by that time. As the stimulus 1 produced no response, $s_t = 3$ was used again at $t = 22$. The reappearance of R_3 then indicated the necessity for disapproval. It is also possible, under similar conditions, for the response learning machine to decay into a lethargic state, making increasingly infrequent responses. In Table 2, columns 1, 2, and 3 for $1 \leq t \leq 15$ are identical with the corresponding columns of Table 1. However, at $t = 17$ and thereafter, s_t (Table 2) was reduced to 1, and the frequency of responses dropped sharply.

3. The Response Learning Program

To control the occurrence of responses, a threshold state number S_i, $S_i \gtreqless 0$, is associated with each response R_i. Columns 4 through 8 of Table 1 display the threshold state numbers at time t, S_{it}, for every i. The set of threshold state numbers is held in the response learning machine's number store. When a stimulus is introduced at time t, the threshold state number store is scanned until the first largest S_{it}, say S_{jt}, is found. At $t = 20$ the scanning proceeded until $S_{3,20}$ was found. S_{jt} found, the machine forms $S_{jt} + s_t$, and tests for the condition $S_{jt} + s_t \geq T$. T is a fixed number, 7 in the present experiments, called the triggering threshold. A response can occur at time t only if the relation $S_{jt} + s_t \geq T$ is satisfied. When this is not the case, X is printed in column 2, and the machine prepares to receive the next stimulus, s_{t+1}. When $S_{jt} + s_t \geq T$, the scanning proceeds, to find and count those S_{kt} satisfying the relation $S_{kt} = S_{jt}$, $k \neq j$. When no such S_{kt} exists, response R_j occurs, and the number j is printed in column 2. At $t = 20$, $S_{3,20} + s_{20} = 5 + 3 = 8 \geq 7$, hence R_3 occurred. When two or more responses compete with each other, as was the case at $t = 17$, the winning response is selected at random. At $t = 17$, R_1 and R_3 were in competition, and R_3 won. The selection of a response by the procedure just outlined relies essentially on the use of conditional orders. Such orders enable the machine to choose among several alternative courses on the basis of the results of earlier operations. Naturally, the programmer must foresee the need for a choice, and provide conditional orders to meet this need, but the actual decision is made by the machine itself, on the basis of information obtained in the course of operation either from its own store, or, by means of the input mechanism, from the outside world.

After a response R_i has occurred, the machine signals to the experimenter and asks for approval. The experimenter then introduces a number a_t of appropriate magnitude and sign into the machine. Given a_t, the machine proceeds to form $S_{i,t+1}$ from S_{it}, for all i. When R_i has occurred at time t,

$$S_{i,t+1} = S_{it}+a_t+1+N_{it}-N_{i,t-1}-d(S_{it},t);$$

the terms of this expression will be described in turn.

By increasing, decreasing, or leaving S_{it} constant, the addition of the factor a_t correspondingly modifies the probability that $S_{i,t+1} > S_{j,t+1}$, $i \neq j$, and hence the probability that the scanning process will stop at $S_{i,t+1}$. Adding unity to the threshold state of the response which has been initiated at time t increases the probability that this response will occur again at time $t' > t$. This device accounts for the habit-forming effect described earlier. It is this effect which, together with the chance selection of R_3 at t = 17, accounts for the S-machine's delayed learning of the response R_1. Attempts to teach this response were begun at t = 16, but were unsuccessful until t = 29, after R_3 had been effectively discouraged.

N_{it} and $N_{i,t-1}$ are both pseudo-random numbers*. In each interval (t,t+1) a pseudo-random number N_t, $-5 \leq N_t \leq 5$, is added to one S_{jt} selected at random, and N_{t-1} is subtracted from the S_{kt} ($k \neq j$, or $k = j$) to which it had been added in the interval (t-1,t). In this fashion, random fluctuations are superimposed on the average level of the threshold states. Because of these fluctuations, the machine can make mistakes, that is, it can occasionally make a response other than the one it has been taught to make. More important, provided that no S_i is excessively large, each S_i has a reasonable probability of being greater than the others at some time. This makes possible the teaching of a new response, or, when no response is favored over the others, produces an interesting variety of responses.

The last factor, $d(S_{it},t)$, produces a decay trend of all threshold states toward 1. d is different from zero only in the intervals between t = 5n and t = 5n+1, where n = 0,1,2,... In these intervals d = +1 when $S_{it} > 1$, d = -1 when $S_{it} \leq 0$, and d = 0 when $S_{it} = 1$. The effect of $d > 0$ is self-explanatory. The negative decay is provided when $S_{it} \leq 0$ for the purely practical purpose of preventing the "death" of the response learning machine. As illustrated in Table 2, the decay introduces some lethargy into the behavior of the machine, by causing all S_i to drop, hence requiring ever-increasing stimuli to

*"Pseudo-random" numbers good enough for these experiments are generated by squaring a certain constant, and by selecting a number of digits from the result. The middle digits of the square then serve as a new constant.

produce a response. Were all S_i to become so strongly negative that $S_{it} + s_t < T$ for all i, given the most favorable random effect and stimulus, no further response could be elicited from the machine. The use of negative decay effectively makes 1 the average minimum level of the S_i. The effect of decay and of the random variations on the threshold states can best be observed in Table 2, when $t > 15$.

For all those R_j which did not occur at time t,

$$S_{j,t+1} = S_{jt} + N_{jt} - N_{j,t-1} - d(S_{jt}, t).$$

In this expression a_t and the habit forming term 1 do not appear. The essential features of the preceding description are thus summarized by the three relations which govern the operation of the response learning machine:

1. To initiate a response at time t:

$$S_{jt} + s_t \geqslant T \text{ for some } j;$$

2. Where R_i has occurred at time t:

$$S_{i,t+1} = S_{it} + a_t + 1 + N_{it} - N_{i,t-1} - d(S_{it}, t);$$

3. Where R_i has not occurred at time t:

$$S_{i,t+1} = S_{it} + N_{it} - N_{i,t-1} - d(S_{it}, t).$$

It must be remembered that, while he is training the response learning machine, the experimenter does not know the threshold state numbers, and must rely on his recollection of his own past actions and of the responses the machine made to them. This is the data given in the first three columns of Table 1.

The behavior pattern of the response learning machine is sufficiently complex to provide a difficult task for an observer required to discover the mechanism by which the behavior of the S-machine is determined. By examining the data of the first three columns of Table 1 such an observer could easily find regularities in the response pattern, and he might even develop empirical rules for predicting responses with tolerable accuracy. He would find it very difficult by this means to obtain a good approximation to the description of the response learning program given above. Switching the machine off to dissect it would be of limited value only, since this action would make the response learning program vanish from the EDSAC's store.

4. The Digital Computer as a Model

The readiness with which the EDSAC, and other digital computers, can play different roles at a moment's notice is one of their important properties. When the role a digital computer is called upon to play is markedly different from those its designers had in mind, a necessarily large proportion of the orders in a program must be devoted to specifying this role. In the response learning program, an aggregate of elementary EDSAC orders is required, for example, to initiate the firing of a response, since this is an operation which does not correspond to any single order. This drawback is balanced by some important advantages. Given the EDSAC, the only additional equipment required to turn it into a learning machine is the length of teleprinter tape which holds the learning program before it is introduced into the machine, and changes in learning machine design and the rectification of errors require at most the preparation of a new program tape. Only a few seconds of input time are required to turn the EDSAC from an orthodox computing machine into an experimental learning device. For these reasons, a flexible digital computer could serve with advantage as a proving ground for a wide variety of models.

Acknowledgments

The author wishes to thank Dr. M. V. Wilkes for suggesting this study. He is indebted to Dr. M.V. Wilkes, Mr. J. W. S. Pringle, Mr. S. Gill, and Mr. M. F. C. Woollett for many valuable suggestions, and to Dr. J. C. P. Miller for helpful advice. This paper is part of a study carried out while the author was the holder of a Henry Fellowship.

References

Oettinger, A. G. (in press), _Phil. Mag._, "Programming a Digital Computer to Learn"

Turing, A. M., 1950, _Mind_, 59, 433-460, "Computing Machinery and Intelligence"

Wilkes, M. V., 1951, _Spectator_, No. 6424, 177-178, "Can Machines Think?"

Wilkes, M. V., Wheeler, D. J., and Gill, S., 1951, "The Preparation of Programs for an Electronic Digital Computer," Cambridge, Mass., Addison-Wesley

Table 1

A typical response learning experiment

t	1 s_t	2 R_{1t}	3 a_t	4 S_{1t}	5 S_{2t}	6 S_{3t}	7 S_{4t}	8 S_{5t}
	02	X		03	03	03	03	03
	02	X		04	03	03	03	03
	02	2	00	03	07	03	03	03
	02	X		03	04	03	01	03
5	02	3	00	03	04	07	03	03

Table 1 (Cont't)

A typical response learning experiment

	1	2	3	4	5	6	7	8
	02	1	00	06	03	03	02	02
	02	X		03	03	02	02	02
	02	X		02	03	03	02	02
	02	3	00	03	03	08	02	02
10	02	X		03	04	04	02	02
	02	X		02	02	03	01	01
	02	X		02	02	02	01	01
	03	2	00	02	05	03	01	01
	03	5	00	02	03	03	01	05
15	03	3	00	02	03	08	01	02
	03	1	01	06	02	03	01	01
	04	3	00	03	02	03	01	01
	03	3	00	03	02	04	01	02
	03	5	00	03	02	05	01	06
20	03	3	00	03	02	05	01	-02
	01	X		02	01	05	01	01
	03	3	-01	01	01	05	01	01
	03	X		02	01	03	01	01
	03	3	-02	02	01	05	01	01
25	03	3	-03	02	01	04	01	01
	03	X		01	01	01	01	01
	03	1	02	06	01	01	01	01
	03	3	-01	04	01	05	01	01
	03	1	02	04	01	01	01	01
30	02	1	01	07	01	02	01	01
	02	1	00	08	01	01	05	01
	02	1	00	11	01	01	01	01
	01	1	00	10	04	01	01	01
	01	1	-03	11	01	05	01	01
35	01	1	-03	09	01	01	01	01
	01	1	-04	06	01	01	01	01
	01	X		04	01	01	01	01
	02	X		04	01	01	01	02
	03	1	-04	04	01	01	01	00
40	03	4	02	01	01	01	04	01
	03	4	01	01	01	02	04	01
	02	X		01	01	01	04	01
	03	4	02	01	01	02	06	01
	03	4	01	00	01	01	09	01
45	02	4	01	01	01	02	11	01
	01	4	-01	01	01	01	10	01
	01	4	-01	00	01	01	13	01
	01	4	00	-01	01	01	13	01
	01	4	-02	01	01	-03	14	01
50	01	4	-02	06	01	01	13	01
	01	4	-04	01	01	01	12	05
	01	4	-04	00	01	01	09	01
	01	4	-04	01	01	01	06	01
	01	X		01	01	01	03	01
55	02	X		01	01	-04	03	01
	03	X		02	01	01	02	01
	03	X		00	01	01	02	01
	03	X		01	01	01	02	03
	03	X		01	02	01	02	01
60	03	2	00	01	06	01	02	01

Table 2

The effect of decay and random fluctuations on the threshold state numbers

	1	2	3	4	5	6	7	8
t	s_t	R_{1t}	a_t	S_{1t}	S_{2t}	S_{3t}	S_{4t}	S_{5t}
	02	X		03	03	03	03	03
	02	X		04	03	03	03	03
	02	2	00	03	07	03	03	03
	02	X		03	04	03	01	03
5	02	3	00	03	04	07	03	03
	02	1	00	06	03	03	00	02
	02	X		03	03	02	02	02
	02	X		02	03	03	02	02
	02	3	00	03	03	08	02	02
10	02	X		03	04	04	02	02
	02	X		02	02	03	01	01
	02	X		02	02	02	01	01
	03	2	00	02	05	03	01	01
	03	5	00	02	03	03	00	05
15	03	3	00	02	03	08	01	02
	03	1	00	06	02	03	01	01
	01	X		02	02	03	01	01
	01	X		02	02	03	01	01
	01	X		02	02	03	01	02
20	01	5	00	02	02	03	01	06
	01	X		01	01	02	01	-02
	01	X		01	01	02	01	02
	01	X		00	01	02	01	02
	01	X		01	01	00	01	02
25	01	X		01	01	02	01	02
	01	X		01	01	01	01	01
	01	X		01	01	01	01	01
	01	1	00	06	01	01	01	01
	01	X		02	01	05	01	01

AN ANALYSIS BY ARITHMETICAL METHODS OF A CALCULATING NETWORK WITH FEEDBACK

By

L. C. Robbins
Burroughs Adding Machine Company, Philadelphia

At the Wayne University meeting of this Association in the spring of 1951, George W. Patterson presented a paper on Reversing Digit Number Systems. Although these systems can be stated with respect to any base, we will be concerned only with the special case of base 2. The reversing representation for base 2 is also called the Gray code, symmetric binary code, and the reflected binary code. It appears like this:

Decimal	Reversing	Normal
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010

where the normal base 2 representation has been shown for comparison. Note that for