# APPROACHES TO ARTIFICIAL INTELLIGENCE

by

Anthony F. Albright

A Thesis Submitted to the Faculty of the

DEPARTMENT OF SYSTEMS ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 6 2

## STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in The University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in their judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Anthony F. Albright

## APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

A. Wayne Wymore — April 26, 1962

A. Wayne Wymore
Professor of Systems Engineering

Date

APPROACHES TO ARTIFICIAL INTELLIGENCE

by

Anthony F. Albright

---

ABSTRACT

The field of artificial intelligence attempts to simulate human intelligence in mechanical devices. It is an infant field for which much enthusiasm and optimism currently exists. Great amounts of money, talent, and time have been spent in research, developments, and trial and error applications. Yet, the field is often marked by modest accomplishments, varied and independent efforts, lack of coherence, and uncertain direction for the future.

Review of this situation reveals that two primary approaches to artificial intelligence appear to have been adopted. The "Simulative Approach" investigations attempt to copy or model the biological processes which underlie the intellectual operations of man. The "Synthetic Approach" investigations make no attempt to copy the means or methods of man, but seek to equal or surpass the intellectual output or results of man for a prescribed task. Representative studies under both approaches are presented.

The review brings out that the field is seriously handicapped by the lack of a generally accepted theory of intelligence. A definition and method of measurement of one aspect of artificial intelligence is developed.

# ACKNOWLEDGEMENTS

The author wishes to express his sincere thanks to Dr. A. Wayne Wymore and Dr. Roger J. Weldon for their support and advice during the preparation of this thesis. He also wishes to acknowledge the patience and understanding of his wife without whose assistance and encouragement this work could never have been completed.

TABLE OF CONTENTS

Chapter | Page

# CHAPTER 1
# INTRODUCTION

## 1.1. Historical Background

With the completion of the first large digital computers shortly before 1950, the subject of artificial intelligence took on new meaning. The topic of the mechanization of human functions is not a new one. Most machines can be thought of as being designed to replace or extend human actions.

One of the first men to state clearly that machines could also carry on the human function of thinking was Jean La Mettrie.[1] La Mettrie, a French physician by profession but a philosopher by preference, wrote in 1746 of the possibility of matter being endowed with the faculty of thought and also of a machine so constructed that it could emit sounds (not necessarily a human language) by which a basis of communication with the human being might be established.

In 1812 Charles Babbage outlined his plans for the first digital computer. Since the British government invested 17,000 pounds (close to a million dollars in present value) in the project, the ideas apparently were very enticing even at that time. The "Analytical Engine," as the completely mechanical machine was to be called, was to have been logically similar to most of our modern computers, but it was never finished. But even then, it is interesting to note, a critic of the period stated that on the basis of the "Analytical Engine", it may be possible to construct electrical equipment which will think.[2]

Samuel Butler wrote in 1871 of the fictitious land, Erewhon, in which a civil war had occurred between the "Machinists" and the "Anti-machinists" over the issue of keeping or destroying their machines. The powers of the machines were so developed that there was fear of the subordination of man by the machine. Butler wrote, "There is no security against the ultimate development of mechanical consciousness in the fact of machines possessing little consciousness now."[3] Butler apparently feared this development for in his story the "Anti-machinists" won and the machines were destroyed.

An early design for a learning machine is that of J. M. Stephens.[4] He considered protoplasmic organisms as very complicated machines and in 1929 described an electric circuit which would exhibit simple conditioning. By using a simple switching circuit, he was able to select one of two outputs according to the presence or absence of "reinforcement." He stated that no new principle would be involved in constructing a machine to make any number of responses to any number of stimuli, such responses to be governed by the reinforcement situation. Stephens generally believed that mechanical simulation of neurological operations was possible, and particularly felt that all concepts of learning are capable of being extended to machines.

In 1943, Kenneth J. Craik[5] wrote of the striking similarity of the functioning of the human nervous system and the operation of various machines. He then attempted to outline a symbolic theory of thought, consisting of three essential thought processes of the human being, in order that possible mechanization of these processes might be accomplished.

By 1950, when the digital computer can be said to have been perfected as a working instrument, the subject of artificial intelligence already had some history. Even as the popular press and periodicals began to speak of "giant brains" and "ingenious machines", A. M. Turing,[6] Claude Shannon,[7] and others were writing their views on the question: "Can machines think?" A new science called Cybernetics had been created; it was defined as the science of control and communication in the animal and the machine. The field of artificial intelligence was beginning to be developed in earnest.

Today, 1962, the field has developed extensively. The complexity and capability of computers have grown. Concepts have expanded and at the same time have become more precise. There have been numerous and varied investigations. Some of these appear to hold great promise, while others have been trivial. It seems clear that mankind could reap tremendous benefits if some of the anticipated potentialities of machines possessing artificial intelligence were to be realized. Man is a slow, often careless, but sometimes brilliant thinker. On the other hand computers are extremely fast and accurate. At one time computers were considered to be very stupid, but this judgment is beginning to be modified.

## 1.2. Objectives of Thesis

The thesis advanced in the following pages is that a review of the work of the past decade demonstrates a powerful development of new concepts and new techniques which take artificial intelligence out of the realm of fancy and into the borderlands of realization. Furthermore, an attempt is made here to extract from the studies of the past a basic concept which underlies the emergence of artificial intelligence. The concept will provide a definition of artificial intelligence and perhaps a valid measure of it.

In summary, the objectives of this thesis are:

1. To present and categorize many of the primary investigations which have been made during the past decade.

2. To analyze and evaluate these studies as bases for a more precise concept of artificial intelligence.

## 1.3. Classification of Approaches

The approaches which investigators have adopted in their efforts toward realization of an artificially intelligent machine fall readily into two general categories, which will be referred to as the "Simulative Approach" and the "Synthetic Approach".

The "Simulative Approach" utilizes the existing information of human processes. Duplication of these processes through copying or modeling is then attempted, through the design and construction of some form of artificial neuron, other mechanical devices or theoretical design. Probable hypotheses are introduced and tested using these models in order to gain better understanding of such abstract entities as "intelligence" and "mind". Hypotheses successfully tested on the models are then tested on humans, whenever possible, in order that new probable hypotheses might be generated and the cycle may be continued.

The advocates of this approach are primarily biologists, psychologists, neuro-physiologists, psychiatrists, etc. They often capitalize on certain striking similarities between the functioning of the human nervous system and the digital computer, and attempt to extend these similarities. Doesn't the human employ his brain to remove desired items of information from his mental storage, use them as desired, and then replace them in storage for future utilization? Isn't this identical to the storage and use of information in the digital computer? Isn't the basic structure of the computer that of a relay or switching circuit perfectly employing binary arithmetic in that each element is capable of only two conditions, i.e., "on" or "off"? Similarly, isn't the basic element of the nervous system the neuron whose function is to act as a relay with essentially two states of activity, i.e., firing and repose?

Occasionally, their studies are marked by research on the brain or nervous system of animals with the anticipation of applying this information toward the understanding of some aspect of human intelligence and the ultimate mechanical modeling of this aspect.

Further, this approach is often characterized by the building of different machines to simulate the various human processes.

This then is the "Simulative Approach" to artificial intelligence. Its primary obstacles to success are the limited knowledge of human thought processes and a frequent redesign of machines, necessitated by the growth in complexity of the processes to be simulated.

The "Synthetic Approach" also is concerned with duplicating mental activity, but the duplication of results is the only matter of importance. It is felt that the information processing systems of the human and the computer are dissimilar, and, therefore, attempts at duplication of the methods or means of the human thought processes are not even considered. This "Black-box" attitude is in perfect accordance with the famed "Turing Test". The late British logician, A. M. Turing,[6] advanced the following test of artificial intelligence. If a qualified observer is unable to distinquish between machine outputs and results produced by humans, with more than chance success, the machine is considered to have achieved intelligence. Quite expectedly, engineers and mathematicians have adopted this "Synthetic Approach". They have sought to assign specific tasks to machines, tasks for which the successful completion had in the past been strictly associated with human intelligence. In some studies, the limitation of the machine to one particular function, as contrasted with human versatility and flexibility, is attacked by attempting to extend a machine to a variety of applications.

The "Synthetic Approach" is characterized by the utilization of standard computers. Very rarely are specific machines designed and constructed. However, this approach is primarily distinquished from the "Simulative Approach" in that the "Synthetic Approach" makes no attempt to copy what is known or believed of human behavior, but possesses complete freedom of method in its attempts to produce outputs which are comparable to human intellectual behavior.

These then are the general descriptions of the two basic approaches to the study of artificial intelligence. Although these approaches are quite distinct, it will become apparent that specific projects may sometimes overlap. However, these are exceptional cases and the classification, which probably has been made possible due to motivation, is most convenient. Specific investigations undertaken under both classifications will now be discussed in detail. The discussion will include many representative studies, and will be sufficiently thorough to permit a critical analysis of the existing "state of the art".

## CHAPTER 2
## SIMULATIVE APPROACH INVESTIGATIONS

For ease of understanding, the "Simulative Approach" investigations will be considered under three categories: General, Pattern Recognition and Learning. The first category will include studies of human capacities which have a definite analogy to machines, comparisons between the computer and the human nervous system, and the creation of artificial neurons and subsequent employment of groups of these constructed elements to form systems. The section on Pattern Recognition will contain descriptions of investigations which seek to discover better perceptual mechanisms in machines. This is a most difficult problem. Computers are only just beginning to be designed to understand the simplest spoken instructions or perceive a letter of the alphabet, tasks which a mere child performs with no difficulty, although it is capable of manipulating millions of alphanumeric symbols per second. Finally, the section on Learning will include models and mechanical simulation studies, usually based on reinforcement principles, which seek to imitate human learning processes.

### 2.1. General

John Von Neumann was scheduled to present the Silliman Lectures at Yale University in 1956. He chose as his subject to compare the computer and the brain, but unfortunately, due to his contraction of cancer and subsequent death, he never made the presentation. His unfinished lectures were published and the resulting book is a particularly interesting "Simulative Approach" to the study of artificial intelligence, a subject which he felt was fascinating and certainly attainable.[8] Von Neumann wrote of the digital character of the acceptance and emittance of pulses by the neuron and, after extensive study, made several comparisons of the basic element of the human nervous system with the basic element of the computer. These comparisons are actually the essence of his work and are tabulated below.

| Category | Neuron | Vacuum Tube or Transistor | Advantage Factor |
|---|---|---|---|
| Speed (Reaction time in sec.) | $10^{-2}$ | $10^{-7}$ | $10^{5}$ (transistor) |
| Size (cm.) | $10^{-5}$ | $10^{-2}$ | $10^{3}$ (neuron) |
| Energy Dissipation | $10^{-9}$ $\frac{\text{watts}}{\text{neuron}}$ | $10^{-1}$ $\frac{\text{watts}}{\text{transistor}}$ | $10^{8}$ (neuron) |
| Memory Capacity (bits) | $10^{20}$ (human) | $10^{7}$ (computer) | $10^{13}$ (human) |

Based on these quantitative evaluations he concluded that:

1. Natural componentry performs its tasks with more but slower organs, whereas its artificial counterpart uses fewer but faster organs.

2. The human thus tends to simultaneously pick up as many items as possible (highly parallel), while a machine is more likely to successively perform actions (highly serial).

3. The natural componentry has a factor advantage of $10^{4}$ in the number of actions which can be performed by active organs occupying the same volume in the same time interval. (This is computed by dividing the volume advantage of the neuron by its speed disadvantage.)

The pioneer of neural simulation studies was probably R. S. Lillie.[9] He employed an iron wire immersed in nitric acid to demonstrate the properties of refraction and all or none response. The acid causes a thin film of oxide to form around the wire. An applied electric shock creates a current which propagates along the wire by destroying the film. A period of refraction then occurs while the acid reforms the oxide film.

A classic study of artificial neurons was conducted by McCulloch and Pitts.[10] These two investigators employed feedback principles to exhibit long term memory in their theoretical model. In particular, an output was fed back as an input in order that the model, once excited, would continue to excite itself, or once inhibited, would fail to excite itself. McCulloch and Pitts believed that, in theory, a machine could be constructed of a sufficient number of these elements to duplicate any behavior which could be completely specified.

L. D. Harmon[11] of Bell Laboratories has recently constructed an artificial neuron. The electronic model is a four transistor device which is capable of simulating many of the operational functions of living nerve cells. In particular, the natural functions which have been incorporated into his artificial neuron are:

1. Inhibition-a particular energized input connection to a neuron is capable of preventing the firing of the neuron by other inputs.

2. Excitation-certain adequately energized input connections to a neuron will always result in the firing of the neuron if the following conditions are satisfied:

    a. Threshold-firing of the neuron will not occur unless the input pulse exceeds the threshold value of the neuron within a definite period of time.

b. Refractory period-a period of fatigue which the neuron endures immediately after firing and an interval in which no firing can occur since the threshold value has temporarily risen to infinity.

c. Summation-two or more pulses may jointly cause firing despite the fact that each are individually incapable of exciting the neuron.

d. Output-firing either occurs and produces a standard pulse of a specified duration or there is no firing and the state of repose remains.

The model is so constructed that threshold values and refractory time constants may be easily altered by making minor circuit adjustments. The circuitry of the artificial neuron is simple and its total cost is less than \$10.00. Experiments on larger network systems, with this model as the basic element, are now being conducted.

D. G. Willis[10] is of the opinion that the ultimate solution of the complex problem of "pattern recognition" and "learning" lies in the initial study of simplified neuron models. Since the number of neurons in the human body is approximately $10^{10}$, and a machine exemplifying human behavior would require that number of the McCulloch-Pitts neuron model, Willis is seeking to develop a more efficient model. He considered only two of the characteristics of human memory and attempted to show how these might be exhibited in a plastic neuron model. The first property was the "random access" nature of memory, whereby vast amounts of stored information may be instantaneously recalled through the proper association. The second characteristic was size, a figure which varies depending upon the investigator. The model is certainly not novel in that it "fires" or "does not fire" as a certain threshold value is exceeded or is not exceeded. It is unique, however, and derives its name in that the influence of one neuron upon the other is not constant but has a variable "synaptic" value which is determined by previous excitations of the neuron. Having the facility to assume many different sets of "synaptic" values, the neurons can be in any of a large number of states at a given time. Thus the plastic neuron does not have the earlier limitation of storing at most one bit per neuron.

Willis has set up a system composed of 288 plastic neurons and has been conducting simulation studies on a Univac 1103 AF computer. Initially, it appears that the model is adequate to account for human memory and the input-output properties are in accord with the "random access" characteristics of human memory. It is hoped, eventually, to extend the concept to the solution of "pattern recognition" and "learning" problems.

Some study is now being devoted by McCulloch and his associates[12] to the problem of constructing networks which give highly reliable output as is found in the nervous system, in spite of the fact that the basic components themselves are not so highly reliable. The human nervous system itself is not totally reliable. For example, errors may be introduced due to a fluctuating signal or a varying threshold, causing a neuron to fire when it is not supposed to. Similar difficulties arise in the design of artificial networks due to the presence of noise or unreliable circuitry. Of primary importance is this problem of reliability in computer networks. Of course, the broad field of information theory is attempting to find solutions to these difficulties. However, some study of the reliability problem in natural networks is being conducted and the information extended to artificial componentry. The primary mathematical tools which are being utilized are Venn diagrams, probability theory, and symbolic logic.

## 2.2. Pattern Recognition

To gain understanding of the pattern recognition faculty of humans, Lettvin, Maturana, McCulloch and Pitts[13] of M.I.T. studied the optic nerve of the frog, hoping that they later might apply any acquired information toward a better understanding of human optic nerves. By using microelectrodes to record impulses in thousands of individual nerves less than 1/25,000 inch in diameter, they discovered that there are four types of neurons, with each type firing upon the occurrence of a different pattern of events at the retina of the eye. Thus, there are four distinct kinds of neurons by which the frog's eye informs its brain of the patterns it currently is watching. The four patterns are: (a) sustained contrast; (b) net convexity; (c) moving edge; and (d) net dimming.

The "sustained contrast" neurons respond to changes in light distribution. "Net convexity" neurons (or bug detectors) detect positively curved or convex edges, such as the front end of an insect. "Moving edge" neurons respond to any distinguishable edge moving through its receptive field. "Net dimming" neurons respond to sudden reduction in illumination.

Further study revealed the important fact that neurons responding to the same pattern were similiarly constructed, but were quite different from neurons responding to other patterns. Thus there exists a definite relationship between the structure of neurons and their particular functions.

Ulric Neisser[14] has studied the information processes of the human being in the hope of utilizing this information in the development of a pattern recognition system. His experiments consisted of measuring the time required for subjects to scan a random list of letters and find particular items. Varying the requirement he demonstrated the flexibility earlier noted by Von Neumann of man's processing system in that it readily adapts to sequential, parallel, or a combination of both modes of operation. The discovery that certain letters are more difficult to find than others prompted the suggestion that different identifications involve different operational features. This explanation of possibly different "shape recognizers" was supported by the subjects who spontaneously reported that the lists were handled completely different for certain requirements. "This result, more than any other, emphasizes the extraordinary adaptiveness of human information processing. It's unlikely that any artificial device we will know how to make in the near future will be as efficient..."[14]

In a very recent article,[15] Neisser has introduced the concept of "functional simultaneity", by which he hopes that some of the parallel processing capabilities of man may be incorporated into the computer. He feels that man has the option either of thinking sequentially or thinking in a multiple fashion, whereby many processes may go on together. The computer, of course, may only operate in a sequential manner, due to the physical limitations of its construction. However, Neisser believes that a computer may perform in a "functionally simultaneous" manner, in the sense that all operations are completed before a decision is made. He suggests the following advantages of a computer with such a capability:

1. There would be a certain degree of nonchalance about internal errors, whereas one misstep under sequential operation is often fatal.

2. Learning and adaptation would be easy to incorporate, whereas a sequential program is difficult to improve.

3. Minor adjustments of weights to various features may produce improvement and not upset the entire system.

However he emphasizes that multiple processing will be justified only if a correct sequence of operations cannot be established. Otherwise sequential operation will be much more efficient. Also, in contrast to sequential operation, one will not be able to determine exactly what the machine is presently doing and what it will do next.

The pattern recognition program, "Pandemonium", is an application of Neisser's theory and it is hoped that it is a beginning to the incorporation into machines of the powerful parallel processing capabilities of man.

A widely known early study was conducted by Clark and Farley.[16] They constructed a neural network, whose operation was patterned after the neural interaction of the human being. The net was arbitrarily divided into an input and output group. The output was subdivided in two and an output for any given time "t" was defined to be the difference in the number of "firing" elements in the subgroups at that time. The input was subdivided into two subgroups and two fixed input patterns, "$p_1$" and "$p_2$" were provided. Input "$p_1$" greatly excited all the elements of one input subgroup and did nothing to the other. Input "$p_2$" operated reversely. Output characteristic of input was obtained by having "$p_1$" cause more "firing" to take place in one output subgroup and "$p_2$" cause more firing in the other. Inputs "$p_1$", and "$p_2$" were then made to drive the output in a specified direction by adding to or reducing the value of a property of each element known as its "weight", which determined whether it fired or not. At a given time, some of the weights may be altered incorrectly, but in the long run a favorable result always occurred. Since a system which always demanded an identical input pattern would be completely impractical and unrealistic, they allowed the input to vary randomly but still experienced considerable success.

Their system, which was simulated on the Memory Test Computer of Lincoln Laboratory, had consisted of only sixteen elements. Although it is possible that fewer failures would occur in larger networks, such networks are still to be constructed.

L. A. Kamentsky[17] of Bell Labs is engaged in the study of spatial pattern recognition or, that is to say, the translation to machine language of spatial symbols. He has built neuron-like elements, which he terms "speurons", which exhibit the neural properties of "all or nothing" output, excitation and inhibition. An actual model, consisting of thirty of these elements, was able to correctly identify simplified inputs of angles, end-points, and closed loops. The model has been extended by the introduction of a scanning device, capable of producing varying input patterns, and simulation studies on the IBM 704 are being conducted.

J. R. Singer[18] of the University of California has devised a system which can presently recognize letters and which will be extended to the recognition of words, people, etc. using identical operational procedures. "When humans learn to read, they are given a simultaneous set of inputs, i.e., they are shown letters and asked for a specific response to those letters. After a time, the human learns to internally program his learning.

That is, by trying out various responses to stimuli the desired response is reached using a nebulous error-correcting signal."[18] The system was based on an electronic analogy of this hypothesis of human pattern recognition and consists of both a learning and a reading mode. During the learning phase, images appear and are centered on a photoreceptor matrix, which has been designed to function as the human retina does. The radial segments in which the image falls then determine a describing matrix which in turn is transformed into a word representation and stored in the computer memory. In the reading phase incoming images are tested for a specified degree of matching with stored patterns. Left hand rotation is investigated by removing matrix rows from the top and placing them on the bottom. The reverse holds for right hand rotation. If an image is not identified within the prescribed limits, the machine automatically goes into the learning mode and stores the word representing the new image. This naturally leads to redundancies, but Singer believes that as additional information is processed, this will not be too relevant. For example, it is much easier to distinguish between "U" and "V" when they are contained in words than when they are alone.

Frank Rosenblatt[19,20,21] of Cornell Aeronautical Laboratory is actively employing the information gained from artificial neuron studies. He is attempting to develop and perfect his "perceptron", a simplified model of the biological brain, consisting of neurons, similiar to those of McCulloch and Pitts, randomly connected and subject to certain laws of growth and organizational constraints. The primary objective of the study is to begin to determine how the brain performs its functions in terms of its structural components. However, a prime application of the system appears to be the solution of the problem of "perceptual generalization". A machine with this facility would be able to recognize any member of a given class (e.g., a triangle in any position, a man in any posture, etc.) after being first exposed to a limited number of members of the class in question. In contrast to the neuron versus transistor approach, which employs the language of symbolic logic and Boolean algebra, Rosenblatt has adopted an entirely different attack. Since no perfect wiring diagram exists for the brain and only its general organization can be characterized, he has based his "perceptron" on probability theory.

The organization of the "perceptron" duplicates the organization of a biological brain as far as physiological and anatomical facts have been established. The "perceptron" basically consists of three major parts: sensory system, association system, and response units. Initially, when the sensory system is exposed to stimuli, the responses are random and have no meaning. In time, however, the association cells of which the association system is composed, cause responses to build up well defined classes or categories as people, individual geometrical figures, trees, etc. The response units cause feed back signals which reinforce the classes in the association system, thereby increasing the probability of correct identification the next time that the stimulation is perceived.

Simulation studies of the model on the IBM 704 computer have been encouraging. Problems such as economical design, introduction of a decay rate to "forget" previous reinforcements and yet maintain a stable learning system, and development of suitable sensing devices, are but a few of many trouble areas which remain to be solved.

## 2.3. Learning

Bush and Mosteller[22,23] have presented mathematical models of learning behavior. A simple system which they simulate mathematically consists of an environment, trainer, and reinforcement machine. The environment presents a stimulus to the reinforcement element, which then must make one of several responses. However the reinforcement machine does possess the ability to remember what decisions were made in arriving at a response. After each response, the trainer, which selectively knows how to determine success or failure of the decisions, sends a positive or negative reinforcement signal to the reinforcement machine. This results in an increase or decrease of the probability that those decisions will be made in the future.

In this simple model of reinforcement system, a reinforcement operator "Z" is expressed in terms of a parameter "θ" so that positive and negative reinforcement are given by:

$$1.\quad P_{n+1} = Z_{+}\ (P_n) = \theta P_n + (1-\theta) \qquad 0<\theta<1$$
$$2.\quad P_{n+1} = Z_{-}\ (P_n) = \theta P_n$$

Thus each desired response is reinforced causing the probability of its future selection to approach unity. Similarly, negative reinforcement of each undesired response continually lowers its future selection probability. The choice of "θ" is then varied according to the desired speed of learning. A near unity value of θ gives slow learning which a small value will cause learning to be quite rapid.

Although this mathematical model was developed only as a model for learning in human organisms, it could readily be extended to machines.

Specifically, the parameter "$\Theta$" could be used as a weighting factor within a program which would alter the decisions which the computer would make.

W. Grey Walter[24,25] has designed and built machines which exhibit similarities to the conditioned learning behavior of animals. The "turtles", as the machines are called, consist of two motors, a photoelectric cell, a touch contact, and an electric circuit of a few vacuum tubes, amplifiers, tuned circuits, capacitors, inductors, and resistors, all contained in a tortoise-shaped shell. Through its three simple characteristics of attraction to moderate light, repulsion by bright light, and repulsion by material objects, the "turtles" were demonstrated to simulate animal learning characteristics. For example, a whistle is blown and then a moderate light is flashed. After about twenty repetitions, the "turtle" has "learned" that the sound means light and is attracted to the whistle as if it were a light. When the light is withheld, and consequently no reinforcement occurs, the response is soon extinguished and the whistle is disregarded. In another example, the device was conditioned to a whistle so that it underwent a withdrawal and avoidance reaction, even though no obstacle was present. As in an animal, this avoidance response is harder to extinguish and is retained much longer. A form of experimental neurosis was induced and readily compared to the neurotic behavior in humans caused by conflicts or inconsistent education. The cure for the "turtle"--switching off or disconnecting one of the circuits--was humourously compared to the sleep and shock remedies of the psychiatrist.

J. K. Hawkins[26] has investigated the possibilities of devising a learning machine which would be modeled after the known characteristics of animal behavior. Since the mechanical duplication of learning processes is in such a rudimentary state, he does not describe an actual system, but confines himself to the concepts upon which such a system should be based. His learning machine would improve its recognition capabilities solely as the result of a repeated joint occurrence of an input and the desired response. It would be based on the creation of a network composed of elements exhibiting properties of synchronous operation, binary operation, and linear weighted summation of inputs. The first property creates discrete time intervals which may be handled mathematically; the second insures that each element is either emitting a pulse or not; the third permits threshold and synaptic values to be mathematically treated. These elements comprising the network will function as either input or output elements and in such a manner as to exhibit both arithmetic ability and storage capacity. When an input appears, the network will continue trying various outputs until the desired one occurs. The network then remains in this desired response position long enough for the recognition parameters to increase. For example, the system might be compared to a language teacher instructing a pupil to say a particular word. He pronounces the word and the pupil attempts to repeat it until he is correct. Then he repeats it over again, thereby strengthening his recognition probability in the future. The system, it must be recalled, is only an idea and has yet to reach the mathematical model or computer simulation stage.

P. M. Milner[27] of the Department of Psychology at McGill University in Montreal has studied learning in neural systems. His belief is that the learning process of the human being may be divided into three phases: exploration, problem-solving, and rote learning (that which has become routine and may be performed without understanding). In the exploratory phase the organism has had no previous experience with its current situation and consequently has no specific goal. In the problem-solving phase, the organism has been in similiar situations and, therefore, does have a goal. In the final phase, the organism has so often repeated the behavior required by the present situation that it has the correct moves by rote. Feeling that the exploration and rote-learning phases can easily be duplicated by machines, he developed a model of the problem-solving phase which employs, as far as possible, the same general principles of the nervous system.

Milner's model has been designed on a functional rather than anatomical basis. Specifically, the model is composed of two primary sections. The first section is responsible for the information processing and storage capabilities, whereas the second contains the goal-seeking drives of the model. In the first section are the cells which respond to stimuli in various patterns and also the basic units which are intended to furnish the additional storage space required for conditions of reinforcement, reward or punishment. The second section contains the reward or positive drive system, the punishment or negative drive system, and a complex motor system, which is connected to these previous drive systems, that possesses the capability to generate the basic acts of the model.

Milner hopes that his model employs the same operating principles as the human, but even if it does not, he feels it should still work. He hopes to eventually conduct computer simulation studies of his model, which would require a storage capacity of approximately $10^{12}$ bits, if and when a machine of such capacity is available.

# CHAPTER 3
# SYNTHETIC APPROACH INVESTIGATIONS

Studies which follow the "Synthetic Approach" are those which specify that only the input and the output of a proposed device need to resemble the inputs and outputs found in human intellective behavior. The processes that are to mediate between input and output of the device can be put together in any manner and out of any materials. This is the "Black Box" approach. These processes are thus "synthesized" and need not be similar to human processes in any way other than that there shall be a successful translation of specified input to specified output. The various studies to be reported fall rather well into the following four classes: 1. Game Playing, 2. Pattern Recognition, 3. Learning, and 4. Problem Solving.

## 3.1. Game Playing

When the study of artificial intelligence was being introduced to the computer field, prospective investigators were confronted with the problem of how to begin to work on the development of a machine which possessed such ill-defined abilities as thinking, reasoning, and perceiving. How is one going to mechanize abstract thought processes which one does not begin to understand? A starting point that was proposed was that of developing programs to play games which are known to require considerable intelligence of human players. It was also considered necessary that the game have specifically defined rules and goals that could be put into a computer program. The most appropriate game meeting these conditions is chess, the intellectual game "par excellence." The working hypothesis was that is a successful chess playing machine could be devised, then considerable progress would have been made toward the understanding and construction of artificially intelligent machines.

The study of chess playing machines actually began somewhat earlier than might be expected. In 1769, a Hungarian inventor, Wolfgang von Kempeler, was astounding large European audiences with his device called the Maelzel Chess Automaton. However, its fame was short-lived. A number of experts quickly and correctly concluded that the automaton was operated by a dwarf chess master concealed inside.

In 1914, a Spanish inventor, Torres y Quevedo, constructed an end game device. The machine would employ a rook and king against its opponent's king and force a checkmate in a few moves, regardless of how its human opponent played. Since a definite set of rules can be set up for making moves in such an end game, the problem was fairly simple.

Relevant applications to the study of approaches to artificial intelligence began with the work of Claude Shannon in 1950.[7] He did not write a specific program, but his thorough analysis of the problem has been widely used by subsequent investigators. The problem was basically divided into three parts:

a. Choose a code to represent positions and pieces as numbers.

b. Find a strategy for choosing the moves to be made.

c. Translate this strategy into a program.

The first part is arbitrary in nature and any satisfactory code should be acceptable. Shannon suggested positive and negative values for opposing pieces and a numerical grid system for position identification.

The second part, calculation of a reasonably good move for any given position, naturally presents the most difficulty. Since there are $10^{120}$ possible variations in an average game, it is impossible to play perfect chess and consider every variation to the end of the game. Thus, chess is always played by making partially complete studies of alternatives. The problem arises as to what depth to go in studying alternatives, and depth of exploration must be in accordance with the speed and operating time of the computer or the player. Selection of moves should be based not only on evaluation of pieces, but also mobility, piece positions, defense of king, and the structure of pawn formation.

Shannon suggested a procedure for translating the strategy into a program in which a "master" program would control nine subroutines. Six of the nine subroutines would deal with movement of individual pieces. The final three are concerned with listing possible moves for a given position, making of moves in memory, and evaluation of proposed moves.

Shannon believed that a machine programmed in this manner could play a brilliant game. It would not be prone to obvious blunders, laziness, and nerves which humans suffer, but on the other hand it would not possess the flexibility, imagination, and learning capacity of the human mind.

A. M. Turing[28] was the first to write an actual program. His program considered all alternatives and explored to whatever depth was necessary to reach a "dead position", that is, one in which no capture, recapture, or checkmate could be made in the next move. Value of material was dominant in the evaluation of a board position. The program was only hand simulated and was beaten by a very weak player in its only game.

In 1956, the Los Alamos Group programmed the Maniac 1 almost exactly after the pattern outlined by Shannon. To gain speed, however, they made certain concessions in the game. The board was reduced to a 6 x 6; number of pawns were reduced from eight to six; and bishops were removed from the game. Castling, two square pawn moves, and "en passant" captures were also eliminated. By making these simplifications, the program was written in only six hundred steps. All alternatives were investigated to a depth of four moves and the average playing time was twelve minutes per move. The program was said to be the equivalent of a player of twenty games experience and was easily defeated by a good player.

In 1957, Bernstein[29] wrote the first complete chess program to be run on a computer (IBM 704). It remains the only program of complete chess, which requires something more than a novice to defeat it. The program determines its best move by first considering every possible square and asking such questions as: Is it occupied? By whose men? Is it being attacked? Can it be occupied? This information is then compiled in tables for later utilization. The seven best possible moves are then selected on the basis of such questions as: Am I in check? If so, can I capture checking piece, block, or must I move? Can I gain material in an exchange? Can I castle? Can I develop a minor piece or occupy an open file? Can I create a pawn chain? Having then selected its seven best moves, each is evaluated to a depth of four moves. The score of a move is measured by successive considerations of gain of material, defense of king, mobility, and control of important squares. In this manner, the machine evaluates the $7^4$, or 2401, variations and uses a minimax procedure to select its best possible move. The program decides on a move in only eight minutes. To consider only one more move and explore it to the same depth would nearly double this time. Bernstein's program often plays like a master, making moves which an expert would consider the only satisfactory ones. However, for a complete game, it is not a master. Newell, Shaw, and Simon[28] have worked periodically on their program since 1955, but it has yet to be completed. Their program is the most ambitious attack on machine chess in that it involves greater complexity through more selective heuristics. It also utilizes the study of human thought processes in chess to gain insight on the design of their program. The program organization is based on a set of goals, whose order of importance is king safety, gain in material, center control, development, king-side attack, and promotion. Their study is unique in that no numerical evaluation function for comparing alternatives is used, but a leading goal merely takes precedence over one that follows it. It is not novel because it employs Turing's "dead position" concept and Shannon's minimax procedure. The program is being written in IPL-IV, a language the authors developed during their work on theorem proving programs. As of 1958, it contained 6,000 steps and it was estimated that it will increase to 16,000 by completion with an expected time per move of up to ten hours. Many problems remain but Newell and Simon predict that a machine will be chess champion of the world by 1968. This prediction, it might be noted, was termed conservative by L. I. Gutenmacher, director of the Russian laboratory for the modeling of human mental processes.[30]

Chess has not been the only game which has been investigated. An IBM scientist, Arthur L. Samuel,[31,32] has a checker playing program which refutes the argument that a machine can never outplay its designer. The program learns from past experience and has reached a state of proficiency in which it does defeat Samuel. The program operates on an initial exploration to a depth of three and a minimax procedure for making the decision which will gain the maximum advantage. The efficiency of the program grows in that it remembers old positions and their values. When it is confronted with a familiar position, it immediately accepts its stored value of the board position and analyzes three additional moves in depth, and stores this new evaluation in memory for the board position. After a few hundred games, the program actually is exploring some positions to a depth of twenty one moves and has remembered billions of positions. Furthermore, it requires approximately thirty seconds for a move, a faster speed than the average human player.

Bridge bidding[31] has been programmed on the IBM 650. The program makes an opening bid or a response in less than fifteen seconds and will respond to one club, short club open, Stayman, and Blackwood conventions. As the criterion of good partners is a common understanding of the systems and conventions, it appears that two programs operating as partners could be quite compatible.

Other games have been programmed. One is the regular, 3 x 3 Tic Tac Toe. Another is a 4 x 4 version of Tic Tac Toe which is sufficiently complex to require a heuristic program. A third is the ancient Chinese game of Nim.

3.2. Pattern Recognition

Bernard Gold[33] of Lincoln Laboratories at MIT directed the development of Maude, a program for machine translation of Morse code sent by human operators. There would be no problem if telegraphers sent ideal code. Marks and spaces would be transmitted in perfect intervals and a machine could process the signals. Human variability, of course, prevents this from being the case and Gold was confronted with the problem of producing a machine, capable of duplicating man's ability to detect patterns of letters in wholes. Maude is heuristic in that it does not memorize words, but knows only standard Morse characters. Actually, the machine receives the marks and spaces in the form of electrical pulses and measures their duration. Since no Morse character is more than six marks long, the machine records the longest and shortest duration in every possible group of six, applies plausible dividing criteria, and identifies the characters. Error rate has been found to be only slightly higher than that of a highly proficient operator and the program is an example of machine recognition of variable input patterns. Of course, the division rule of Morse Code makes the pattern recognition capabilities very specialized.

William C. Dersch[34] of the IBM Advanced Systems Development Division is investigating the mechanical recognition of speech patterns. After investigating various approaches, he feels that his system has better chance of success if its logic is not required to duplicate human functions and use the same information that humans do. The problems with which he is confronted are best described by stating some human capabilities. Man readily adapts to the fact that speech varies with the speaker. Within limits, the rate at which words are spoken do not affect his comprehension and he can distinguish one voice when several people are simultaneously speaking. The task is indeed great and Dersch has initially simplified his problem by imposing certain limitations:

a. Words are artificially separated in time.

b. Words are natural and not shouted, whispered, etc.

c. Machine is adjusted to the voice characteristics of the speaker.

d. The speaker is briefly trained to accommodate the machine.

e. Vocabulary is limited.

On the side of realism, the machine would recognize any sequence of the words in its vocabulary, recognize words as they were spoken rather than an hour later, and would not be disturbed by room noises. A computer called "Shoebox", because of its minute size, recently was demonstrated. Having an approximate vocabulary of twenty words, numbers zero through nine and computational words such as "add", "minus", etc., it received verbal instructions and successfully performed the commands to operate arithmetically on groups of numbers. Current emphasis is being placed on extending its vocabulary and increasing its ability to respond to variations of voices.

G. P. Dineen[35] has investigated the recognition of simple visual patterns. He has considered only the block capitals, "A" and "O", and the geometric figures, squares and triangles, and sought to discover how these might be correctly identified despite variations in orientation, thickness, and size. His theory was that any given configuration from a definite class (A, O, etc.), subjected to a sequence of operations, would reduce to the same set of numbers. The first problem of having the computer "see" the image was solved by making each of 8100 cells of a 90 x 90 array either white or black. The visual image described was then read into the machine and stored. The unknown image is then operated upon until it is identified by matching it to one of the stored set of numbers. The first operation is an averaging operation which smooths irregularities in the image. A resulting more homogeneous image is produced by considering each element of the matrix as the center of the 5 x 5 square which surrounds it. If the number of black elements in the 5 x 5 square exceeds a certain threshold, then the element under consideration is black in the new image. If the threshold is not exceeded, the element is white in the new image. An edging operation picks out edges, corners, junctions, and end points, and reduces the input image to a new image where only the outline remains. Some initial success has been achieved, but it remains to extend the limited input and to introduce additional operations, especially one capable of measuring curvature.

A machine, which they hope will eventually read handwriting, is being investigated by Grimsdale, Sumner, Tunis, and Kilburn.[36] The latest known progress of the machine was successful recognition of capital and small letters of the English alphabet, the Greek alphabet, and numerals, but an inability to recognize script or other symbols possessing a large amount of detail. The system is based on obtaining a description of the shape of the figure and does not merely involve a geometrical comparison of the unknown image with a stored set of standards. Furthermore, the system readily processes distorted patterns with breaks or ragged edges, and compensates for dirt smudges on the paper. Basically, the operation begins with a build up in the computer memory of a library of standard patterns, variations of which the computer will have to recognize. Recognition is then accomplished in three phases: scan, assembly, and comparison. During the scanning process, divisions of the figure into segments (e.g. vertical and horizontal lines comprising an "L") and "cleaning" of figure imperfections is accomplished. In the assembly phase, detailed analysis of length, shape, and curvature of the segments, and reduction of the original two dimensional image to a one dimensional pattern, is undertaken. From this analysis a simplified coded representation of the pattern, independent of orientation and size, is formed. Recognition then occurs by searching the small portion of the library, which has been subdivided into classes according to common features, for the favorable comparing standard. If the machine is unable to recognize a pattern, it will indicate this, and if given the name of the pattern, it will store it as a standard for subsequent recognition. The program is approximately 4,000 instructions and recognition time is about 60 seconds, an interval of time which could be reduced by at least $10^3$ on special purpose equipment. Extension to recognize handwriting remains the primary goal.

Selfridge and Neisser[33] are studying the development of the "Pandemonium" system of recognizing hand printed letters of the alphabet. Input to the system is accomplished by flashing an image of the letter onto a bank of photocells, the output of each cell controlling a binary device in the computer. Identification by the machine is then accomplished through analysis rather than matching previously stored digital forms. Two different possibilities, sequential and parallel processing, have been considered as methods for conducting this analysis. In the sequential mode branching is used to eliminate all possibilities except one, which then is a letter identification. For example, to identify a letter, such questions as "Is there a vertical line?", "Is there a crossbar?", etc., might be asked with "Yes," or "No," branches continuing until the identity of the letter is assured. In the parallel mode all questions are asked in a manner that is called "functionally simultaneous", which was described in Chapter 2. It is felt that "functional simultaneity" may be the beginning to the incorporation within computers of the powerful parallel processing of man. Specifically, Selfridge and Neisser have favored parallel processing in their "Pandemonium" system for the following reasons.

a. To compensate for common distortions of letters in slant, shape, and size, and resulting ease of error, elaborate sequential mode back tracking procedures are necessary to correct erroneous decisions.

b. Some features (e.g. Does "B" have a crossbar?) cannot be answered by only two alternatives but require some type of relative value system, which is easily incorporated into the parallel mode.

c. Small changes in a parallel processing system for experimental purposes are easily accomplished.

d. Humans seem to use a parallel system for handling pattern recognition.

Such a system which employs 28 tests of features and a 32 by 32 matrix for a simulated input has been investigated for distinguishing between ten hand-printed letters. A learning phase, during which a few samples of each of the letters are presented and identified, naturally precedes the test for recognition. The letter with the highest total probability after undergoing all tests is selected as the correct identification. The program currently makes only 10% fewer correct identifications than humans, and changes are being made to improve its performance.

### 3.3. Learning

Claude Shannon[37] has constructed a simple electrical machine which exhibits a modest degree of learning ability. The sensing device, which is called the "mouse", learns and remembers the correct route through a maze. The maze consists of a large square which has been partitioned into several small squares. A number of sides are opened so that numerous paths through the maze are formed, but there exists only one route to a particular goal square. Powered by two motors for horizontal and vertical movement, the "mouse" moves through the maze with two relays for each square storing the direction which the mouse departed its square. If the mouse returns to a square before reaching its goal, this information is cancelled and the new departure direction is stored. By a trial and error procedure, the goal is eventually reached and all relays have been fixed. On his next attempt, the "mouse" then moves without error through the maze to its goal.

It should be noted that Shannon's "mouse" appears to be an extension of Stephens' electric learning machine which was described earlier. Each of the small squares closely corresponds to Stephens' basic circuit concept and the maze is a set of several of these individual circuits.

Friedberg,[38,39] assisted by Dunham and North, has attempted to devise a machine which would learn to accomplish tasks for which it is given no specific instructions. This learning procedure was tested for program writing tasks by simulation studies on the IBM 704. The hypothetical computer is given an extremely simple task (e.g., insure that the same bit is in two specific storage locations or compute the sum of two bits and store it in a particular location). Having a few simple instructions at its disposal, it then randomly generates a program, consisting of 64 commands. The success or failure of each program is noted in order that instructions appearing in successful programs will be most likely to be used in later trials. This is accomplished by having a success number associated with each instruction. Operating on these principles the machine did generate many successful programs but it never even attained the success that pure chance would expect it to have, despite many modifications of the generation and reinforcement principles. In some cases, success numbers would be retained from an old task for the beginning of a new one; in others random success numbers were generated and utilized. An instruction from a program which failed was dropped on a random number basis or on the success number principle. Dropping of many instructions after failure was compared to a dropping of one. The program was given the ability to partition a problem into parts and then deal with the parts in order of difficulty. Despite these and other tests, the success of the program remained limited. In a final effort to improve the program's performance, a "priming and reset" operation gave the machine useful starting instructions to accomplish its prescribed task. Still the results remained disappointing and caused the investigation to be at least temporarily discontinued.

Mary E. Stevens[40] of the National Bureau of Standards has devised a model of a machine which is capable of both learning and forgetting material. "IQ", as the computer program is called, is basically a modest information retrieval program, which, in response to commands such as MATCH, REJECT, DEFINE, etc., searches its 768 word vocabulary of proper nouns and adjectives for answers to queries. The learning of new information is accomplished through a LEARN command. This causes an unknown term, which has been presented with known terms, to be stored and assigned descriptive references common to the known terms. However it is initially labeled provisional and if given as an answer during this provisional period, it is preceded by the word "maybe". It remains in this interim status until a REWARD operation following a response indicates to the machine that it has been correctly used and the term then becomes a permanent part of storage. Forgetting must occur when the addition of a new term would cause overflow in the vocabulary storage. Internal provision within the program causes highly specific words to be dropped in order that a general vocabulary is retained.

Kilburn, Grimsdale, and Sumner[41] have devised a program generating machine which displays two types of learning: learning by experience and learning by teaching. The former is exhibited in that the machine uses programs it has developed in the past to generate new programs. Learning by teaching occurs when the machine is initially given simple problems to assist it in solving more difficult problems later. In general, the machine is described as operating similar to the manner of a human inventor. Confronted with a problem, it looks to its memory for ideas, which are then tested to determine if any of them are a solution. Specifically, the system is composed of a master computer, which can generate programs attempting to satisfy certain criteria, and a sub-computer which is capable of testing the programs. The sub-computer is composed of an accumulator, several storage registers, and an arithmetic performing device. There are six basic instructions from which the programs are produced:

$$A' = A + Sr$$

$$A' = A - Sr$$

$$A' = A \times Sr$$

$$A' = A \div Sr$$

$$Sr' = A$$

$$A' = Sr$$

in which:

A is current number in accumulator.

Sr is current number in the $r^{th}$ storage register.

A' is new number being stored in the accumulator.

Sr' is new number being stored in the $r^{th}$ storage register.

The manner in which the programs are constructed depends upon the criterion involved. Since simple convergence is satisfied by a variety of different programs, convergence was selected as their initial test criterion in the form:

$$|A_4 - A_3| < |A_2 - A_1|$$

The system was not random in that instructions were chosen dependent upon previous successful choices and the program was tested against the criterion after each additional instruction. Several successful programs were generated and presented some new ideas concerning convergence. As the machine learned and remembered all successful programs, the rate of production and complexity of new programs increased. The system was improved in that irrelevant instructions were eliminated from a completed program, and a loop instruction was added for repeating sets of instructions any number of times. The criterion was changed to sequences and again considerable success was experienced. The authors feel that applications to particular fields such as electronics and nuclear engineering are very likely. Their work continues with the current emphasis being placed on a system which would produce its own criteria and then invent a program to satisfy it.

### 3.4. Problem Solving

A. M. Turing[42,43] described a machine (basically similar to recent computers) which he claimed could complete any problem-solving task that man himself could carry out. The Turing machine was extremely slow but it demonstrated an ability to solve very complex problems. Basically, the machine was composed of a writing or erasing device, scanner, a motor to move a tape, dial with pointer, and a logical control. Completing the system was a tape of presumably infinite length which had been divided into successive squares. The machine operated on each of these squares in turn by performing one of the following items:

a. Write an "X"

b. Write a "1"

c. Erasure of either of these marks

d. Move the tape one square to the left.

e. Move the tape one square to the right.

f. Stop.

Which of the six commands to be performed was determined by the pointer setting on the dial and the present square before the scanner. From these two conditions, the logical control would give the machine its instruction from a prepared table of commands. In this manner, this "programmed" machine was able to complete complex computational problems, despite the extreme simplicity of each operational step. Turing also demonstrated his concept of a universal machine which would be versatile enough to accomplish any given task. By enumerating the special purpose Turing machines, the universal machine would read a specific number and then solve the problem that that Turing machine would have solved if it operated on a blank tape.

In seeking to design a machine whose behavior exhibits more of the characteristics of human intelligence, H. L. Gelernter[44] of IBM has written a program which discovers proofs in elementary Euclidean plane geometry. The method is not algorithmic but proves the theorems heuristically in the manner of a clever high school student. The functions of the program fall into one of three divisions: heuristic, diagram, and syntax. The heuristic computer controls the operation. It contains the intelligent criteria by which the most appropriate or most likely possibilities are selected. The remaining two computers merely follow its instructions and answer its questions. The syntax computer contains the formal system (symbols, theorems, axioms, rules of formation, etc.) and its objectives are to establish the proof. It verifies the correctness of sequences of lines of proof which have been submitted to it. The diagram computer makes constructions and measurements by means of coordinate geometry and floating point calculations. The geometer, as the machine is called, readily proves theorems in well under five minutes, but basically its operation is static. Its only learning ability, and this is modest, is the addition of newly proved theorems to its formal system for future utilization.

Two other studies have developed theorem proving programs. Wang[44] has written an algorithmic program which is capable of proving all the symbolic logic theorems in the Principia Mathematica of Russell and Whitehead. Newell, Shaw, and Simon[32] developed their Logic Theorist which successfully proved 38 of 52 theorems in the Principia through heuristic methods. One of their proofs, it might be noted, was completely original and was considered to be more elegant than the existing proof for that theorem.

Newell, Shaw, and Simon,[45] went on to develop a program to find solutions to problems in general. The 10,000 step program, General Problem Solver, was written in IPL (Information Processing Language) and was designed to simulate human problem solving ability in detail. Data was collected by having college students think aloud while solving problems in symbolic logic. GPS was written and revised until its trace favorably compared with the protocol recording of the students. Basically, the program operates on problems which can be readily expressed in terms of objects and operators. The executive program consists of three specific instructions:

a. Find a way to transform object "a" into object "b".

b. Apply some operator "q" to object "a".

c. Reduce the difference between object "a" and object "b" by modifying "a".

GPS attempts simplification by reducing the problem to several intermediate objectives. When these are achieved, by repeated use of the above rules, they are then utilized to solve the original problem. The program has already solved a substantial range of problems which includes proving trigonometric and algebraic identities, discovering proofs for logic theorems, and performing integration and differentiation. The possibility of developing intelligent learning in the GPS is currently being investigated. However, only rough hand simulation studies have been conducted and nothing more than initial observations have been published.[46]

Green, Wolf, Chomsky, and Laughery[47] have attacked the tremendous problem of devising a computer-centered system in which man communicates with computers in natural language. As an initial step toward this goal, they have recently written a program which answers questions presented in ordinary language. "Baseball" as the program is called, was written in the IPL-V language and basically consists of two parts: linguistic and processor. The linguistic part reads the question from the punched card, analyzes it to ascertain its meaning, and classifies the information as known or unknown. The processor searches through the data for the appropriate answer and prints it. The stored data consists of the day, month, city, teams, and scores for every game played in the American League during one season. For simplicity, the input questions are restricted to a single clause, and do not possess such connectives as <u>and</u>, <u>or</u>, and <u>not</u>, comparative adjectives as <u>least</u> or <u>most</u>, and sequential statements as <u>in a row</u>. If a submitted question contains an undefined word, the unknown word is printed out in order that the questioner may revise his question. Barring violation of these limitations, the program successfully answers all questions which might vary in difficulty from "Who did the White Sox lose to on July 16?" to "Where did each team play in June?". Ambitious plans lie ahead for the program. Attempts will be made for response in sentence rather than outline form. The program will be augmented to include multiple-clause questions, and comparative words. A direct-entry keyboard will be utilized for solving ambiguities and increasing vocabulary by permitting the computer to query the questioner. Finally, the generality of the program will be tested by changing the context from baseball to possibly voting records and measuring its effectiveness.

# CHAPTER 4
# A PROPOSED DEFINITION AND SYSTEM OF MEASUREMENT FOR ARTIFICIAL INTELLIGENCE

A basic problem in the study of artificial intelligence is to determine that which is the object of study. What is artificial intelligence? The problem is solved in a practical way, as we have seen in the previous chapters, by pointing to a specific human activity, declaring it to be intelligent behavior, and then concluding that artificial intelligence had been produced when a non-biological device reproduces to a reasonable degree that behavior. But this kind of answer does not provide a basic and definable concept of what artificial intelligence is.

## 4.1. Need for a Definition

A great deal of time and effort has been spent by psychologists and others in searching for a definition of intelligence. Some have defined it as the ability to learn by experience. Others have defined intelligence as a global characteristic of persons who exhibit skills in various measurable ways. Still others have said that the measure of intelligence is directly associated with the amount of exhibited selectivity; that is, a system exhibits intelligence when it processes a large quantity of information through appropriate selection. Another concept of intelligence is found in Turing's test of intelligence, the imitative game mentioned earlier. None of the existing definitions of intelligence which we have seen are specifically applicable to computer programming, although Thorndike's concept of the altitude of intelligence has some similarity to the concepts proposed herein.

Recent investigations of artificial intelligence have raised not only new possibilities for research but also new demands for specific definitions of concepts and goals. In general it seems that a number of definitions of intelligence can be used, each fitting into its proper sphere of research and application. It seems proper to develop a concept, and a definition of that concept, that will be of service in the field of programming for the automatic digital computer. Because there is some difference of opinion in this limited field it is necessary to recognize that what follows is a tentative definition.

As was made apparent in Chapters 2 and 3 the majority of artificial intelligence investigations have employed the digital computer. One important reason for the use of the digital computer is that it is much easier to construct new and complex programs than it is to construct new and complex machines. It is entirely possible that even those "Simulative Approach" studies, in which specified hardware was designed and constructed, could have been simulated with a computer program. The definition, which will be developed in the succeeding paragraphs, will not define intelligence in an absolute sense. Rather, the definition will generate a measure of artificial intelligence which seems to be implicit in the attempts to simulate intelligence on digital computers.

## 4.2. Implications for a Definition

There seems to be a promising approach to a useful definition of artificial intelligence which is implicit specifically in the concept of computer subroutines. Numerous standard subroutines have been programmed and these are treated as units in developing more complex routines and programs. Although a subroutine is composed of many machine operations it can be put into another routine or program as a unit. By the use of such units, larger and more complex programs can be made with relative ease.

In a sense these computer subroutines and routines within programs are quite similar to the hierarchy of composition that is found in nature. Much of the complexity of matter may be discussed through the establishment of some form of hierarchical order. For example, elementary particles form atoms, which in turn form molecules, and which in turn form living cells. In other words, there seems to exist an hierarchical explanation of levels within levels. It seems appropriate to apply this general concept to the development of a scale of hierarchical orders of artificial intelligence.

Hierarchical orders are found not only in more complex programs, but also in the automatic digital computer itself. A computer is a highly complex machine, made of metals, these being formed into wires, dialectrics, frames, etc., these in turn being formed into components and circuits and these finally being developed into a machine that can perform a small number of basic operations upon suitable command. The basic operations are performed very rapidly, always in series, in any order specified by the program, and for an indefinite time limited only by the patience, ingenuity and resources of the programmer. Although these operations may vary from one computer to another, most machines have the built in capability to add, subtract, multiply, divide, shift digits a specified number of locations, store and retrieve information, and branch upon the occurrence of a specified event, such as the appearance of zero in the accumulator. These operations of the machine are controlled by humans through the medium of a program which informs the machine of the sequence in which the operations are to be performed.

Hierarchical orders are also found in the program, and it is among these programmed hierarchical orders that artificial intelligence begins to appear in a form that can be recognized as "intelligence" without too much imagination. A program can be thought of as a series of commands which the programmer gives to the computer through an input media of either punched cards, paper tapes, magnetic tapes, or typewriter. The program may be very elementary, consisting of a few instructions, each of which is performed only once. On the other hand the program may be quite complex. It may contain many elements which form a unit that will be called a <u>recursive routine</u>. The recursive routine is of particular importance in the concept of artificial intelligence and will be defined in the next paragraph. A subroutine may or may not be a recursive routine. A generic term is needed to explicate the concept of artificial intelligence; the term "recursive routine" has been selected for this purpose. The presence of these recursive routines will determine the order, or altitude, of artificial intelligence in the proposed concept.

## 4.3. Proposed Definitions and Rules

The following two definitions and four rules are given to define and quantify the altitude aspect of artificial intelligence. It is recognized that other dimensions of artificial intelligence may be quantified in other ways.

Definition 1. A recursive routine is defined as a group of at least two computer operations beginning with some operation a and ending with a branch operation b, one side of which branch returns the computer to operation a.

Note: A recursive routine may contain one or more recursive routines within it.

Definition 2. A recursive routine properly contains another recursive routine when the contained recursive routine and at least one other non-trivial operation or routine lie between a and b.

Rule 1. A first order recursive routine is one which does not properly contain any other recursive routine.

Rule 2. An N order recursive routine is one which properly contains one or more N-1 order recursive routines, N being a positive integer.

Rule 3. A program is of 0 order artificial intelligence if it does not properly contain any recursive routine.

Rule 4. The order of artificial intelligence of a program is that of the highest order recursive routine found within it.

## 4.4. Illustrative Flow Diagrams

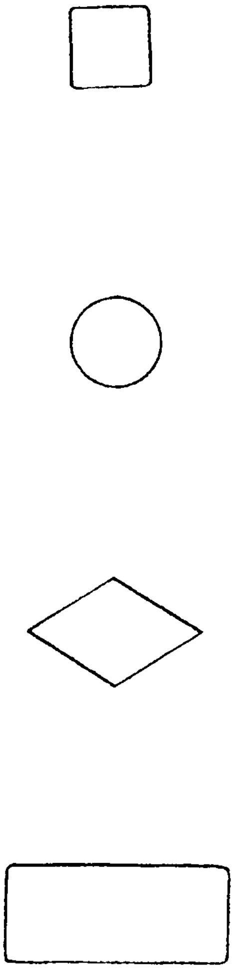
The above definitions and rules comprise the basic structure of the proposed measure of artificial intelligence. Several example programs are shown in Figure 2 through Figure 10 to illustrate the application of the definitions and rules. For clarity the programs are shown in the form of flow diagrams. Figure 1 gives the meanings of the symbols used in the flow diagrams.

Figure 2 illustrates a program of zero order artificial intelligence. The flow diagram shows that there are no recursive routines in the program. Rule 3 therefore applies, and the order of artificial intelligence is zero. Several operation boxes are shown in the flow diagram to emphasize that according to the rules given, the addition of any number of machine operations does not increase the resulting order of artificial intelligence.

Figure 3 illustrates another program of zero order artificial intelligence. In this flow diagram there are no recursive routines and Rule 3 is applicable. The point emphasized in this example is that a branch operation alone does not increase the order of artificial intelligence.

Figure 4 illustrates a program of the first order of artificial intelligence. Because the flow diagram shows that the program properly contains one first order recursive routine, Rule 1 and Rule 4 are to be used. The program is thus of the first order of artificial intelligence.

Figure 5 also illustrates a program of the first order of artificial intelligence. The flow diagram shows a program in which there is



Beginning of a Program

End of a Program

Branching Operation

Arbitrary Number of Basic Machine Operations Excluding Branching

Figure 1

Symbol Notation

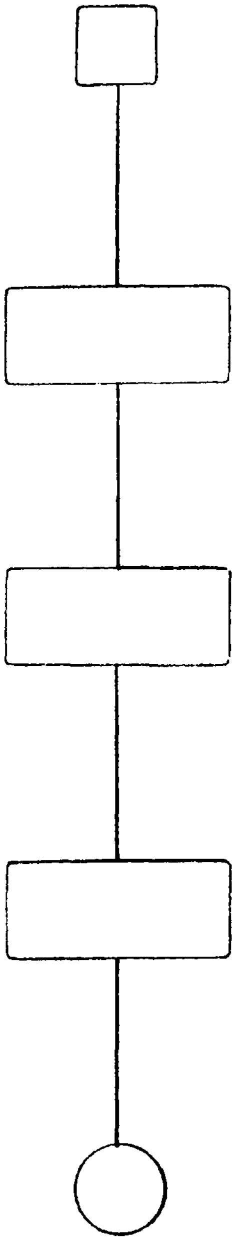


Figure 2

Program of 0 Order Artificial Intelligence

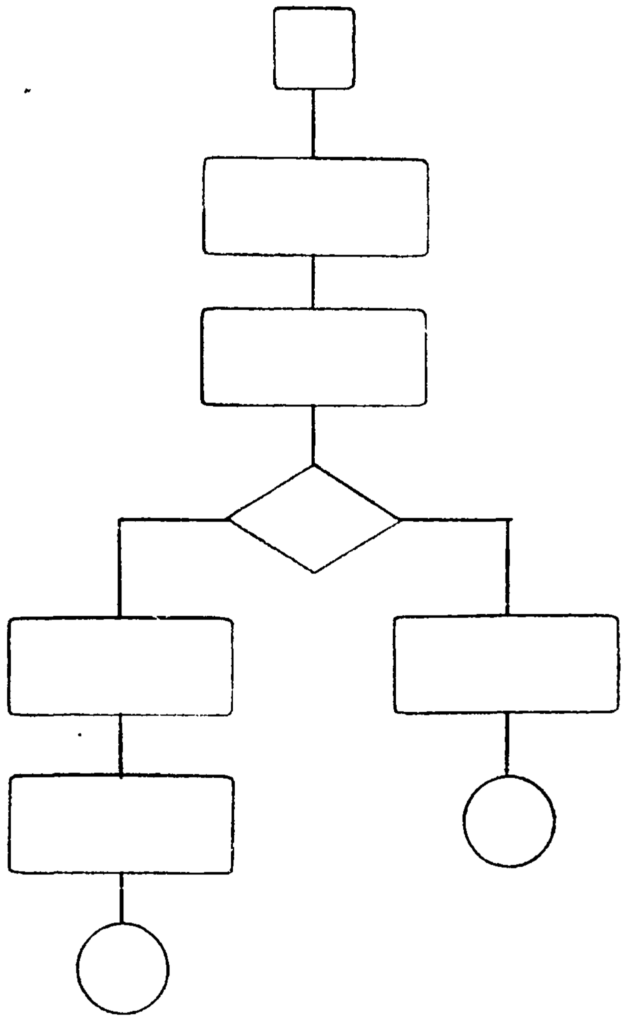


Figure 3

Program of 0 Order Artificial Intelligence

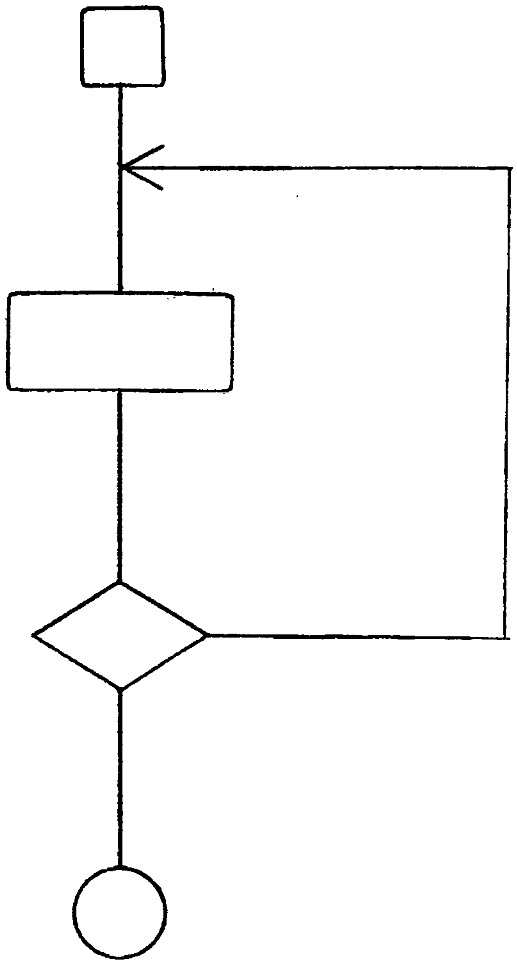


Figure 4

Program of 1st Order Artificial Intelligence

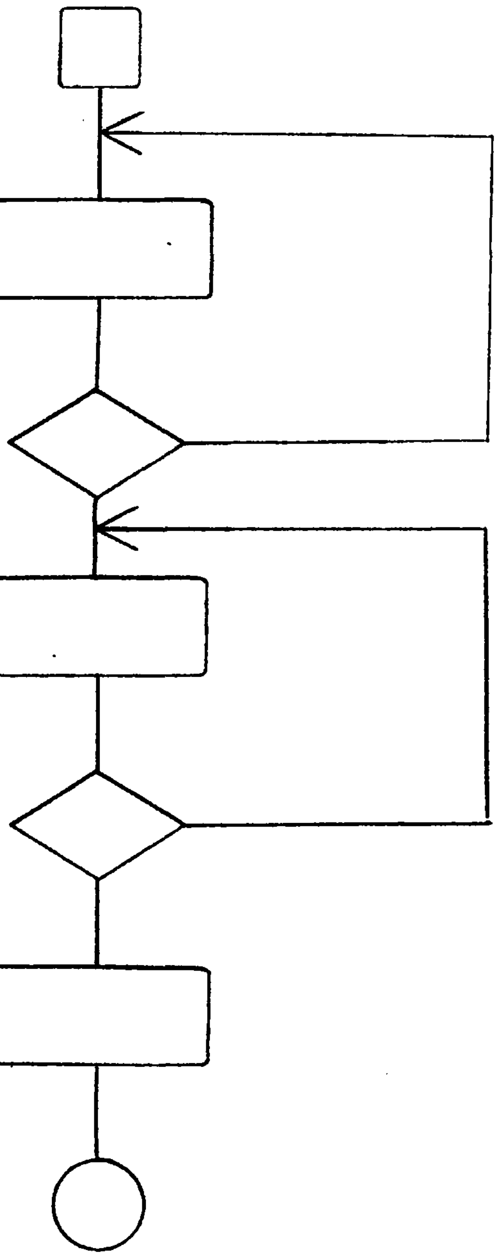


Figure 5

Program of 1st Order Artificial Intelligence

a succession of first order recursive routines. Applying Rule 1 and Rule 4, it is determined that the program is of first order artificial intelligence.

Figure 6 illustrates still another program that has first order artificial intelligence. The flow diagram shows an initial branch with two alternative paths in the program. The left path at A has a recursive routine; the right path has no recursive routine. According to Rule 4 the highest order recursive routine in the program determines the order of artificial intelligence, and this is found in the left path after the initial branch. Thus the program is at the first order of artificial intelligence. The illustration shows that neither the initial branch nor the simplicity of the alternative path on the right affect the order of artificial intelligence.

Figure 7 illustrates a program of second order artificial intelligence. The flow diagram shows a simple first order recursive routine, A, nested within a second order recursive routine, B. Since there is nothing else in the program, it is obviously of second order artificial intelligence.

Figure 8 illustrates another program of second order artificial intelligence. The flow diagram shows that the program is the same as that given in Figure 7, except that a first order recursive routine, C, is added after the second order recursive routine, B, is completed. This illustrates the point that the addition of recursive routines does not affect the order of artificial intelligence unless such successive recursive routines are of higher order than those that have preceded.

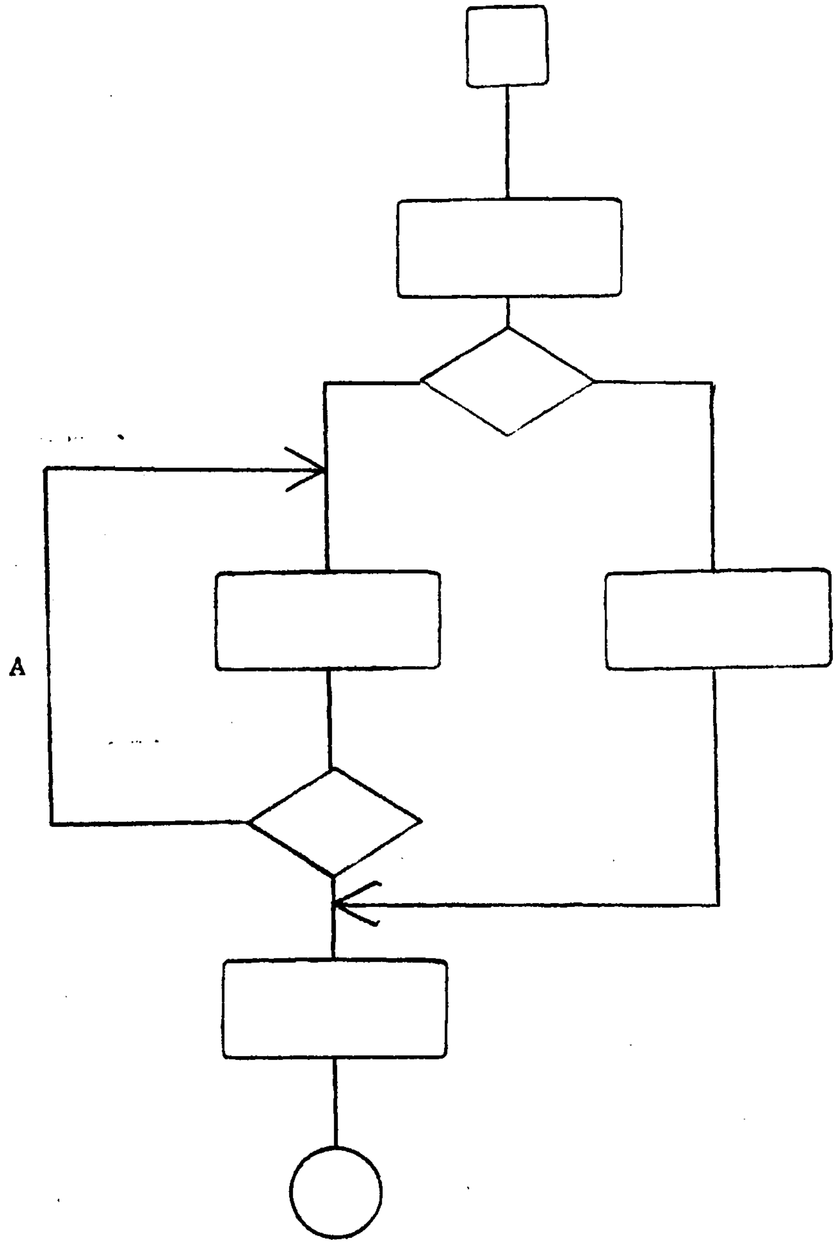



Figure 6

Program of 1st Order Artificial Intelligence

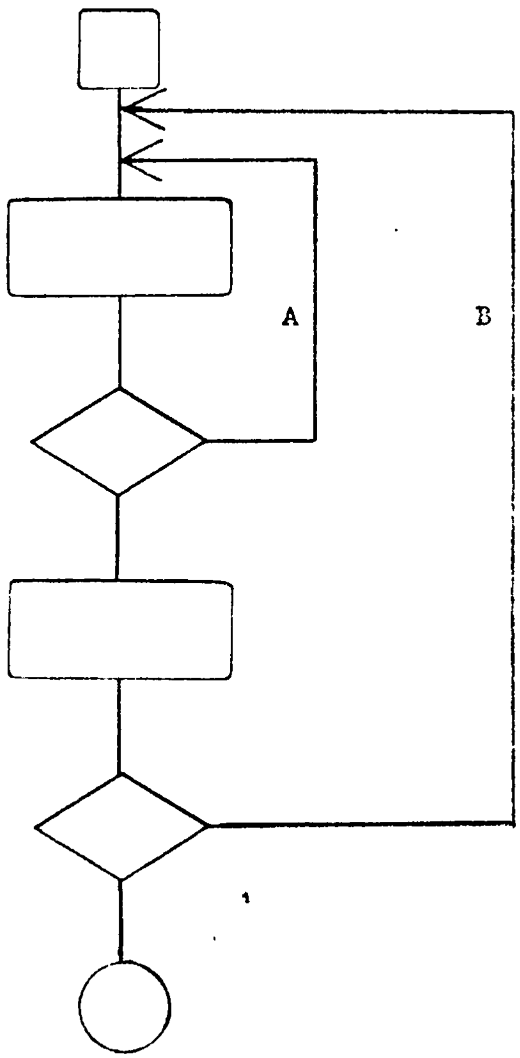



Figure 7

Program of 2nd Order Artificial Intelligence

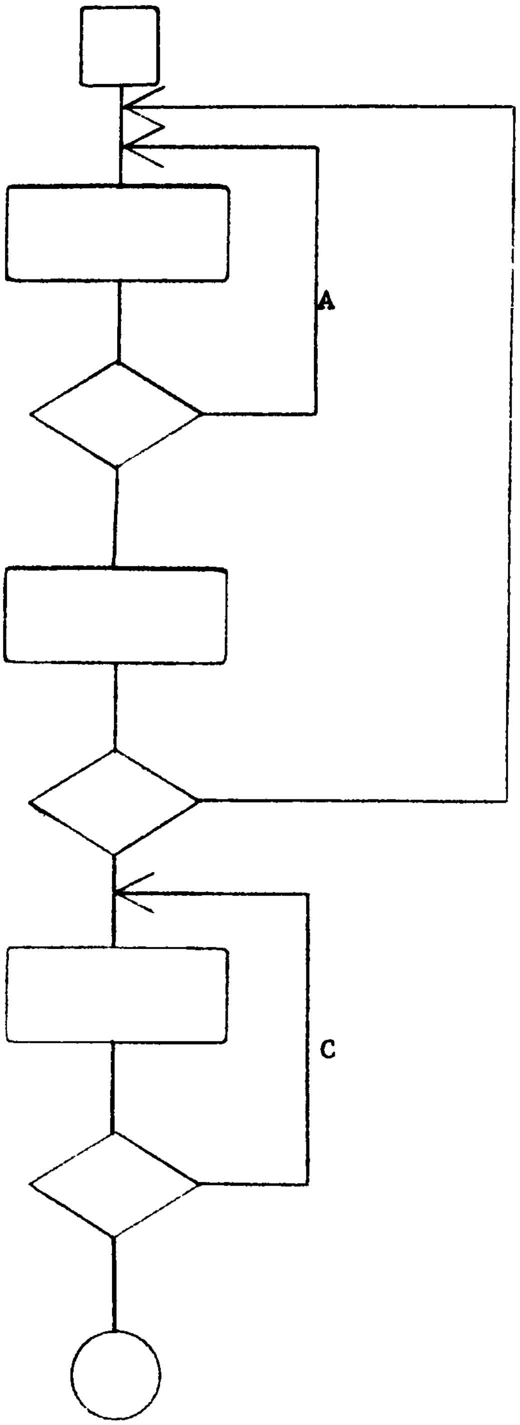



Figure 8

Program of 2nd Order Artificial Intelligence

Figure 9 illustrates a program of third order artificial intelligence. It is the simplest possible program of this order. A first order recursive routine, A, is properly contained in a second order recursive routine, B, and it in turn is properly contained in a third order recursive routine, C.

Figure 10 illustrates a slightly more complex program which nevertheless by the rules given above and illustrated in the preceding figures is of third order artificial intelligence. The order of the largest recursive routine in the program is three, although it contains more operations and more smaller recursive routines than the program in Figure 9.

## 4.5. Examples of Actual Programs

Figure 2 through Figure 10 have illustrated in detail the application of the rules proposed to quantify artificial intelligence with respect to the dimension of altitude. These are formalized programs that bring out the manner of appliaation of the rules. The next programs shown in Figure 11 and Figure 12 are actual programs used on computers. The following paragraphs analyze these programs according to the proposed rules in order to determine the order of artificial intelligence found in them.

Figure 11 is the flow diagram of a program in Runcible language to find the square root of any non-negative number. The method is based upon the Newton-Rhapson formula. The program moves down to the first branch where a test is made for a non-negative number. If the number is

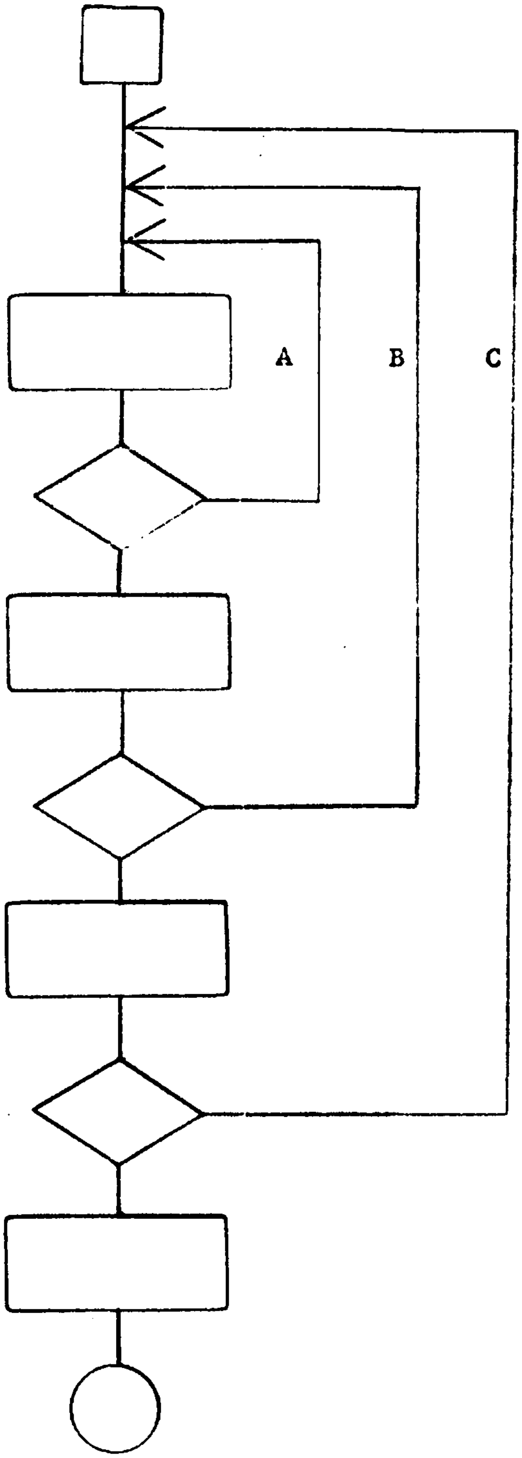



Figure 9

Program of 3rd Order Artificial Intelligence

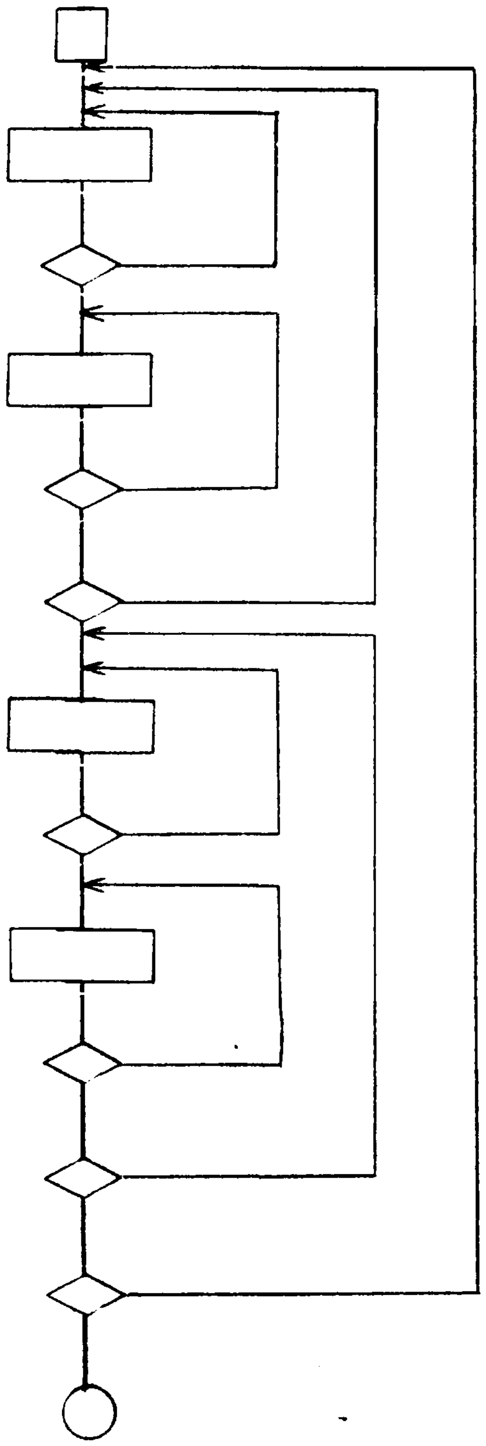


Figure 10

Program of 3rd Order Artificial Intelligence

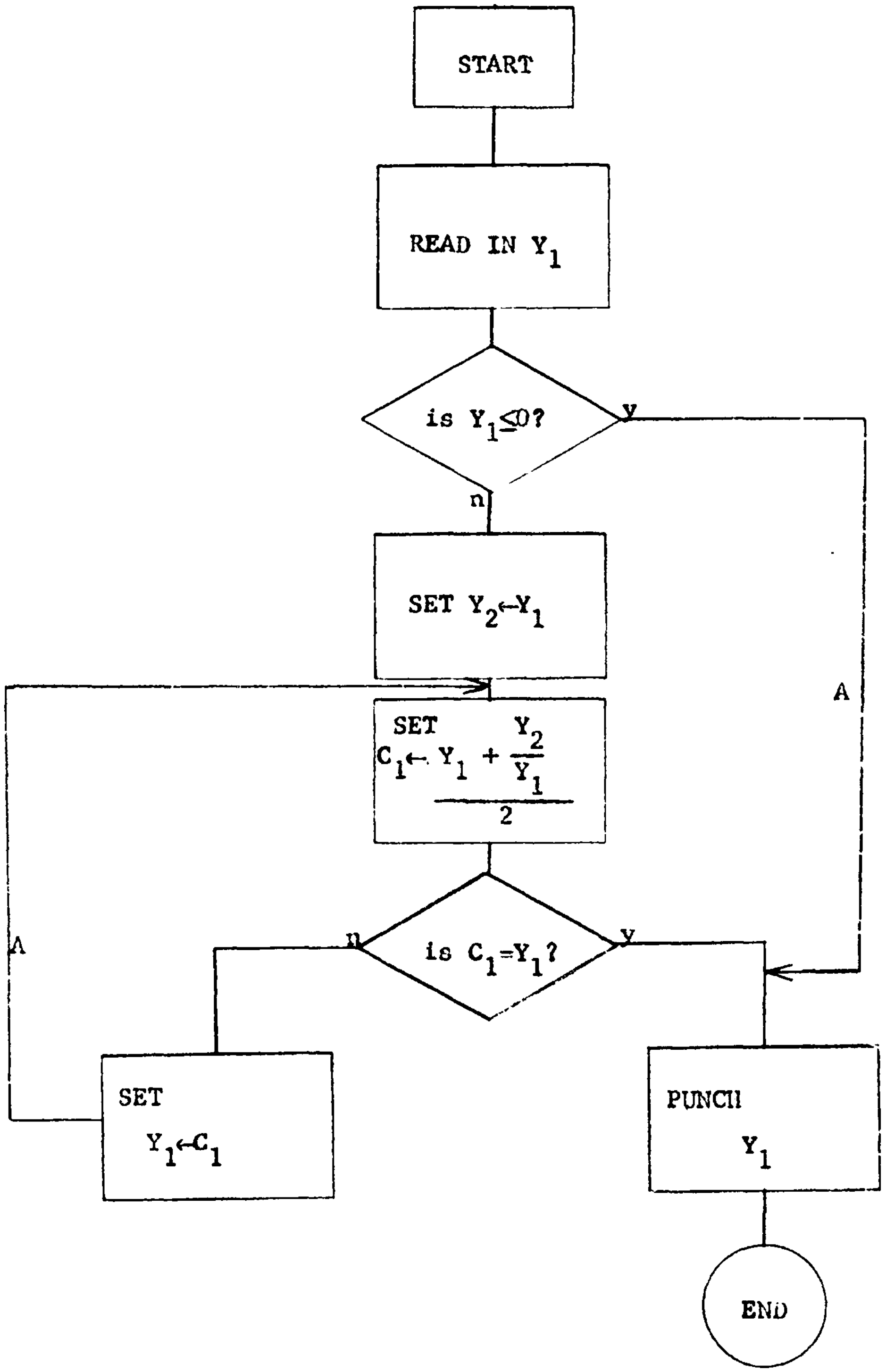



Figure 11

Actual Program of 1st Order Artificial Intelligence zero or negative, the program branches to the yes side and a card is punched giving the value of $Y_1$. If the number is positive the main line of the program is followed, branching to the no side at the second branch and returning along route A recursively, until the square root is obtained as accurately as the machine can obtain it. Thus there is one recursive routine and the order of artificial intelligence of the program is one.

Figure 12 illustrates a Fortransit program for performing a multiplication of two conformable matrices. To investigate the order of artificial intelligence of the program one observes that there is a recursive route at A. It is a first order recursive routine because it does not contain another recursive routine. Since B is also a recursive routine and properly contains A, it is a second order recursive routine. Then it can be seen that C is a third order recursive routine because it properly contains B. This is the highest order in the program and therefore the program is of third order artificial intelligence.

It should be emphasized that the proposed rules for measuring artificial intelligence are tentative. In a sense they are arbitrary as are most rules of measurement. Nevertheless they are intended to be a means to determine unambiguous and consistent measures of artificial intelligence for any program. Although a number of illustrative and actual programs have been analyzed, it will be necessary to study many more before the adequacy of the proposed rules can be tested with some approach to finality. A serious difficulty in doing this is that few complex programs are published in detail, and the proposed rules demand a detailed analysis of a program.

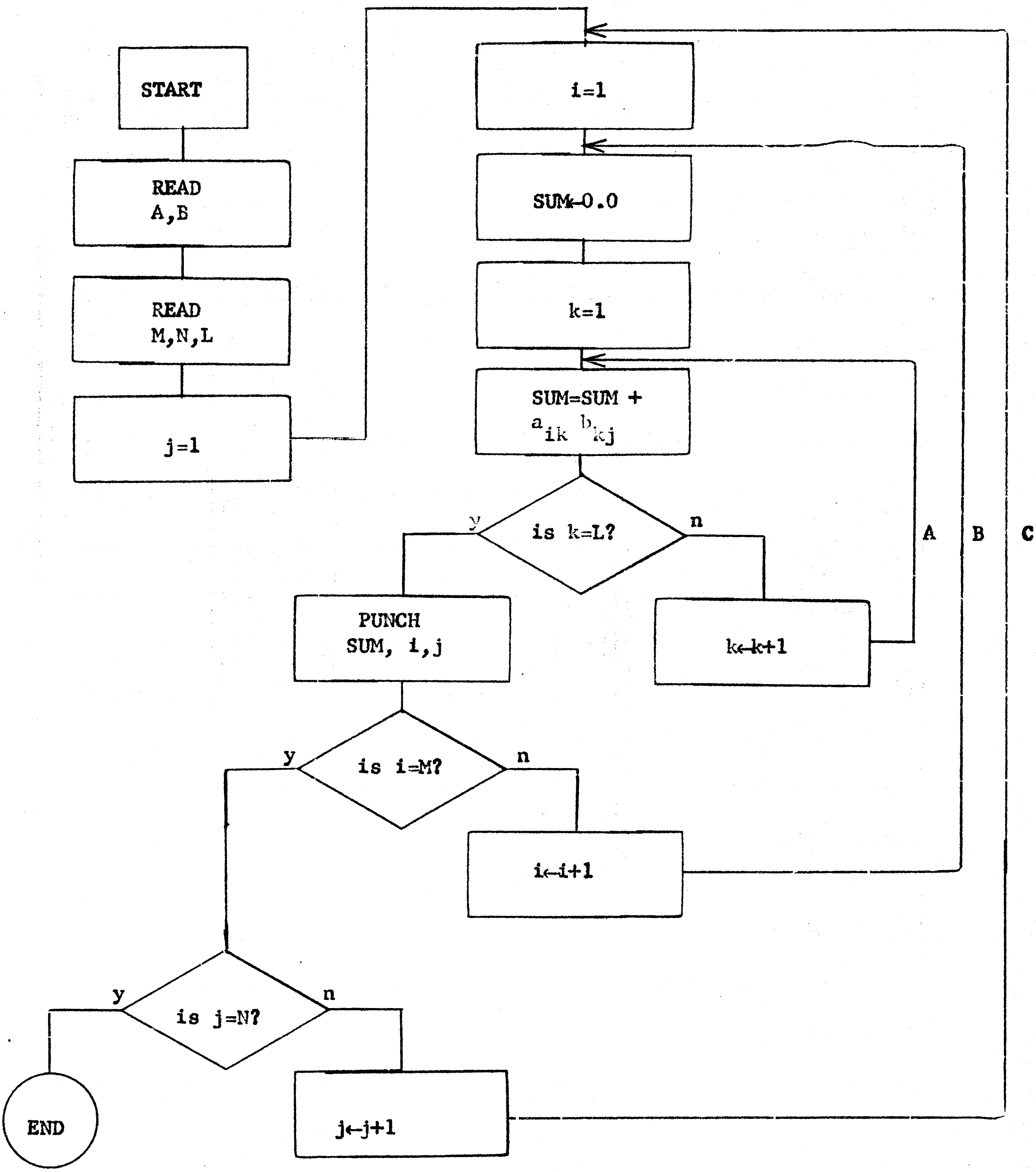



Figure 12

Actual Program of 3rd Order Artificial Intelligence

A complex program was published in Communications of the A C M, November, 1959. It is a program of the Runcible algebraic compiler. Although it is shown in considerable detail, the detail is not sufficiently complete to make an unambiguous determination of the order of artificial intelligence of the program. For example, a set of four branch operations are shown only as one symbol, a five way switch. To ascertain the artificial intelligence this switch should be decomposed into its several branch operations. A general problem arising out of this and other situations in the program is whether reversing the sequence of recursive routines in a program will change the determined order of artificial intelligence of the program. For example, if there are two recursive routines, A and B, it is sometimes possible that A will occur first and thus be included in B, or on the other hand that B will occur first and thus will be included in A. So far as the present study has gone such a reversal of sequence of routines will not produce different orders of artificial intelligence, but the question has not been completely answered.

A determination was made of the order of artificial intelligence of the above mentioned Runcible compiler, although it could not be done rigorously. By making some arbitrary decisions at points where ambiguity arose, it is estimated that it has an eighth order of artificial intelligence. However, the recursive routines found in the program are not iterative routines like those found in computational programs, and this also raises a problem which must be left open. The problem is, whether computational and non-computational recursive routines can be given equal value in determining the order of artificial intelligence.

## 4.6. Summary and Conclusions

The aim of this chapter has been to introduce a set of rules that can be applied to a digital computer program of any complexity and thereby determine an unambiguous order of artificial intelligence for the program. It is essentially a set of simplifying rules. Operations which do not add to the number of recursive routines are ignored or telescoped together, so that the program can be simplified to only the important essentials. These essentials can then be readily counted to determine the order of artificial intelligence.

In conclusion, it is interesting to speculate as to the highest meaningful order of artificial intelligence that a program might attain. There appears to be no logical upper bound to this order. It may be that the limits of man's ingenuity will establish the upper limit of artificial intelligence. Or, it may be that computer characteristics and size will impose an upper limit. It may be that the upper limit of artificial intelligence, unrestricted by the limits of biological inheritance, will in some respects far outreach that of natural intelligence. Speculations of this kind will not be completely fanciful if precise measurements of intelligence can be made. It seems quite possible that artificial intelligence, whether it be closely analogous to natural intelligence or not, will be the first to be measured with precision.

CHAPTER 5
REFERENCES AND BIBLIOGRAPHY


1. La Mettrie, J. O. D. Man A Machine (Chicago: Open Court Publishing Co., 1927).

2. Stibitz, George R. and Jules A. Larrivee, Mathematics and Computers, New York: McGraw-Hill Book Company, Inc., 1957.

3. Butler, Samuel, Erewhon; or, Over the Range, New York, E. P. Dutton and Co., 1901.

4. Stephens, J. M., "A Mechanical Explanation of the Law of Effect," American Journal of Psychology, 41: pp. 422-431, July, 1929.

5. Craik, K. J. W., The Nature of Explanation, Cambridge: The University Press, 1943.

6. Turing, A. M., "Computing Machinery and Intelligence," Mind, LIX: pp. 433-460, October, 1950.

7. Shannon, C. E. "A Chess-Playing Machine," Scientific American, 182: pp. 48-51, February, 1950.

8. Von Neumann, John, The Computer and the Brain, New Haven: Yale University Press, 1958.

9. Boring, Edwin G., Sensation and Perception in the History of Experimental Psychology, New York: D. Appleton-Century Company, Inc., 1942.

10. Willis, D. G. "Plastic Neurons as Memory Elements", Proceedings of the International Conference on Information Processing, Munich: R. Oldenbourg, 1960, pp. 290-297.

11. Harmon, L. D. "Artificial Neuron", Science, 129: pp. 962-963, April, 1959.

12. McCulloch, W. S. "Background", in Joan C. Robinette, ed. Bionics Symposium, WADD Technical Report 60-600, 1960, pp. 51-152.

13. Lettvin, J. Y., Maturana, H. R., McCulloch, W. S. and W. H. Pitts, "What the Frog's Eye Tells the Frog's Brain", Proceedings of the Institute of Radio Engineers, 47: 1940-1951, November, 1959.

14. Neisser, Ulric. "Time Analysis of Logical Processes in Man," Proceedings of the Western Joint Computer Conference, New York: The Institute of Radio Engineers, 1961, pp. 579-585.

15. Neisser, Ulric. The Multiplicity of Thought, Lincoln Laboratories, Massachusetts Institute of Technology. To Be Published.

16. Clark, W. A. and Farley, B. G., "Generalization of Pattern Recognition in a Self Organizing System," Proceedings of the Western Joint Computer Conference, New York: The Institute of Radio Engineers, pp. 86-91, 1955.

17. Kamentsky, L. A. "Pattern and Character Recognition Systems," Proceedings of the Western Joint Computer Conference, New York: The Institute of Radio Engineers, 1959, pp. 304-309.

18. Singer, R. J. "A Self-Organizing Recognition System," Proceedings of the Western Joint Computer Conference, New York: The Institute of Radio Engineers, 1961, pp. 545-554.

19. Rosenblatt, Frank, "The Design of an Intelligent Automaton," Research Trends, Cornell Aeronautical Laboratory, Inc, VI: pp. 1-7, Summer, 1958.

20. Rosenblatt, Frank, "Perceptual Generalization over Transformation Groups", in Yovits, Marshall C. and Scott Cameron, eds. Self-Organizing Systems, London: Pergamon Press, 1960, pp. 63-100.

21. Block, H. D., "Analysis of Perceptrons", Proceedings of the Western Joint Computer Conference, New York: The Institute of Radio Engineers, 1961, pp. 281-287.

22. Bush, Robert R. and Frederick Mosteller, Stochastic Models for Learning, New York: John Wiley and Sons, Inc., 1955.

23. Minsky, Marvin, "Steps Toward Artificial Intelligence", Proceedings of the Institute of Radio Engineers, 49: pp. 8-30, January, 1961.

24. Walter, W. Grey, "An Imitation of Life," Scientific American, 182: pp. 42-45, May, 1950.

25. Walter, W. Grey, "A Machine That Learns", Scientific American, 185: pp. 60-63, August, 1951.

26. Hawkins, J. K. "Self Organizing Systems - A Review and Commentary," Proceedings of the Institute of Radio Engineers, 49: pp. 31-48, January, 1961.

27. Milner, P. M. "Learning in Neural Systems," in Yovits, Marshall C. and Scott Cameron, eds. Self-Organizing Systems, London: Pergamon Press, 1960, pp. 190-204.

28. Newell, A., Shaw, J. C. and Simon, H. A., "Chess-Playing Programs and the Problem of Complexity," IBM Journal of Research and Development, 2: 320-335, October, 1958.

29. Bernstein, A. and M. de V. Roberts. "Computer vs. Chessplayer", Scientific American, 198: pp. 96-105, June, 1958.

30. Armer, Paul, "Attitudes Toward Intelligent Machines," in Joan C. Robinette, ed. Bionics Symposium, WADD Technical Report 60-600, 1960, pp. 13-39.

31. Samuel, Arthur L. "Programming Computers to Play Games," in Alt, Franz L., ed. Advances in Computers, New York: Academic Press, Inc., 1960, pp. 165-192.

32. Pfeiffer, John, "Problems, Too, Have Problems," Fortune, LXIV: pp. 144-148, 154-168, October, 1961.

33. Selfridge, Oliver G. and Ulric Neisser, "Pattern Recognition by Machine," Scientific American, 203: pp. 60-68, August, 1960.

34. Dersch, William C. "A Decision Logic For Speech Recognition," in Joan C. Robinette, ed. Bionics Symposium, WADD Technical Report 60-600, 1960, pp. 287-306.

35. Dineen, G. P. "Programming Pattern Recognition," Proceedings of the Western Joint Computer Conference, New York: The Institute of Radio Engineers, 1955, pp. 94-100.

36. Grimsdale, R. L., Sumner, F. H., Tunis, C. J. and T. Kilburn, "A System for the Automatic Recognition of Patterns," Proceedings of the Institution of Electrical Engineers, 106: pp. 210-221, March, 1959.

37. Gilstrap, L. O. and R. J. Lee, "Learning Machines," in Joan C. Robinette, ed. Bionics Symposium, WADD Technical Report 60-600, 1960, pp. 437-450.

38. Friedberg, R. M. "A Learning Machine: Part 1," IBM Journal of Research and Development, 2:2, p. 13, January, 1958.

39. Friedberg, R. M., Dunhan, B. and H. J. North. "A Learning Machine: Part 11," IBM Journal of Research and Development, 3: pp. 282-287, July, 1959.

40. Stevens, Mary E. "A Machine Model of Recall," Proceedings of the International Conference on Information Processing, Munich: R. Oldenbourg, 1960, pp. 309-315.

41. Kilburn, T., Grimsdale, R. L. and F. H. Sumner, "Experiments in Machine Learning and Thinking," Proceedings of the International Conference on Information Processing, Munich: R. Oldenbourg, 1960, pp. 303-309.

42. Shannon, C. E. and J. McCarthy, eds. Automata Studies, New Jersey: Princeton University Press, 1956.

43. Kemeny, John G. "Man Viewed as a Machine," Automatic Control, New York: Simon and Schuster, 1955, pp. 132-146.

44. Gelernter, H. "Realization of a Geometry Theorem Proving Machine," Proceedings of the International Conference on Information Processing, Munich: R. Oldenbourg, 1960, pp. 273-282.

45. Newell, A., J. C. and H. A. Simon, "Report on a General Problem-Solving Program," Proceedings of the International Conference on Information Processing, Munich: R. Oldenbourg, 1960, pp. 256-264.

46. Newell, A., Shaw, J. C. and H. A. Simon. "A Variety of Intelligent Learning in a General Problem Solver," in Yovits, Marshall C. and Scott Cameron, eds. Self-Organizing Systems, London: Pergamon Press, 1960, pp. 153-189.

47. Green, Bert F., Jr., Wolf, Alice K., Chomsky, Carol and Kenneth Laughery. "Baseball: An Automatic Question-Answerer," Proceedings of the Western Joint Computer Conference, New York: The Institute of Radio Engineers, 1961, pp. 219-224.

## SELECTED BIBLIOGRAPHY


1. Ashby, W. Ross. An Introduction to Cybernetics, London: Chapman and Hall Ltd., 1961.

2. Ashby, W. Ross. Design for a Brain, London: Chapman and Hall Ltd., 1960.

3. Ashby, W. Ross. "What Is an Intelligent Machine?" Proceedings of the Western Joint Computer Conference, New York: The Institute of Radio Engineers, 1961, pp. 281-287.

4. Licklider, J. C. R. "Man Computer Symbiosis," IRE Transactions on Human Factors, HFE-1: pp. 4-11, March, 1960.

5. McNaughton, Robert. "The Theory of Automata, A Survey", in Alt, Franz L., ed. Advances in Computers, New York: Academic Press, Inc., 1961, pp. 379-421.

6. Minsky, Marvin. "Descriptive Languages and Problem Solving," Proceedings of the Western Joint Computer Conference, New York: The Institute of Radio Engineers, 1961, pp. 215-218.

7. Neisser, Ulric and Paul Weene. "A Note on Human Recognition of Hand-Printed Characters," Information and Control, 3: pp. 191-196, June, 1960.

8. Reitman, W. R. "Information Processing Languages and Heuristic Programs: A New Stage in the Bead Game," in Joan C. Robinette, ed. Bionics Symposium, WADD Technical Report 60-600, 1960, pp. 409-417.

9. Uhr, Leonard. "Intelligence in Computers: The Psychology of Perception in People and in Machines," Behavioral Science, 5: pp. 177-182, April, 1960.

10. Wiener, Norbert. Cybernetics, New York: John Wiley and Sons, Inc., 1949.