WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

THE MEANING AND MECHANICS OF INTELLIGENCE

Frederick Hayes-Roth

July, 1971

569-71

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

THE MEANING AND MECHANICS OF INTELLIGENCE

Frederick Hayes-Roth

July, 1971

569-71

Acknowledgment

I would like to thank everyone in the Managerial Information for Planning and Controls group at the Sloan School of Management for the support and stimulation they provided for this research.

This work was performed while I was a research associate at the Sloan School, January through June, 1971. It is part of a research effort in Intelligent Systems within the MIPC group and has been supported by the Ford Foundation Grant #690-0143 under the direction of Zenon S. Zannetos.

F.H-R

June 30, 1971

TABLE OF CONTENTS

PART TWO (Unfinished)

THE MEANING AND MECHANICS OF INTELLIGENCE

1.0 INTRODUCTION

By this time, many people have lost interest in the controversial question, "Can a machine think?" For one thing, the question stubbornly resists attempts at operationalization. Turing, for example, suggested the alternative question, "Can a machine pretend it is a person of specified sex as well as a person of the opposite sex who lies about its identity?" Others have argued that the important question is, "What are the common features of biological and artificial computational networks that account for the production of accurate solutions to problems?" To these questions can be added a myriad of possible distinctions between human and machine intelligence. Among these are issues of intelligence, learning, motivation, rationality, induction, synthesis, conceptualization, analogy, understanding and consciousness.

These are undoubtedly among the most prominent and intractable questions which we face. It is my principal thesis that even the most abstract of these ideas, such as consciousness, can be operationally defined in a suitable framework. In this

monograph, a machine will be described which will exhibit some of these "intelligent" properties. It is my hope that those capacities which remain unattainable by this particular machine will, however, become explainable and comprehensible by contrast.

It is important at the outset to clarify exactly what my purposes are. First, I intend to introduce a level of abstraction into more practical research on intelligent systems. That is, a single machine (which may helpfully be considered as a programmed computer) is discussed in order to facilitate the investigation of all machines that exhibit similar properties. Thus, a machine capable of learning about its environment provides a comparative basis for understanding other intelligent systems, including people. Secondly, I hope to demonstrate that induction by machine is not only feasible but is quite simple in certain contexts.

Moreover, it will be suggested that the activity of induction can be accorded the qualities of spontaneity and higher-order intelligence only in a somewhat contrived sense. Induction, like problem-solving, is shown to be a purely mechanical and natural effect of more basic operations on the part of the organism.

Finally, by considering the properties of a prototypic thinking machine, inferences can be drawn about many interesting problems. The ramifications of this work extend to machines and organisms in general, and these generalities will occupy the remainder

of the paper. In the end, many of the seemingly refractory issues related to intelligence, rationality, and thinking should become soluble and comprehensible in the context of our framework.

1.1 An Outline of the Paper

In such an effort as this, potential semantic pitfalls abound. Thus, the next section will introduce several definitions upon which we will rely throughout the paper. We will construct somewhat unusual definitions for concepts like knowledge, learning, and concept. From these, we will be led ultimately to hierarchically defined notions of conceptualization, intelligence, and consciousness.

Armed with these operational statements of the properties of intelligent systems, we will describe the operation of one such machine in section three. It will be shown that a machine equipped with a few rudimentary features can learn to solve many problems which transcend the capabilities with which it is originally endowed. Moreover, the machine provides a demonstration of the mechanized genesis of predictive models. Without specific pre-programming, the machine discovers salient attributes of symbolic sequences and proceeds successfully to solve many diverse prediction problems.

In section four, we will address the question, "Does this machine think?" It will be possible to consider this problem directly and operationally. The classes of problems which the limited thinking machine can and cannot solve are considered. Notions of limited intelligence and mechanisms whereby such a

machine could expand its knowledge are introduced.

In section five, the induction problem is reconsidered. It is shown that the question, "Can a machine perform induction?" is of limited interest and importance. Instead, it is the construction and integration of concepts into the conceptualizing framework of the machine which are of significance. The class of behaviors typically regarded as inductions or syntheses, I will argue, are not distinguishable from many other logical behaviors. Thus, the original induction question is robbed of whatever significance it may have been presumed to possess.

The topic of the sixth section is the macroscopic pattern of behavior of intelligent organisms. The relations among behavior, intelligence, and personality are explored. In particular, we will focus on the way in which the behavior of an organism is structured and organized by its intelligence mechanisms. Some common but complex behavioral patterns, both personal and interpersonal, are considered in this way. It is in this section also that the arbitrariness of ideas is demonstrated. As corollaries to this significant finding, we will consider the possibility of boundless knowledge and the patently fallacious positivism underlying the common intelligence test.

In the seventh section, consideration is given to the state-of-the-art in intelligence science. Especial attention is paid to the significance of our prototypic thinking machine and to its limitations. Other related topics which were overlooked are considered in this section in a somewhat broadened perspective. A

formal disclaimer is made of the accuracy or appropriateness of this particular machine as a descriptive model of human intelligence. Nevertheless, I will suggest that any machine which exhibits intelligence and which may effectively simulate some human activities is a provocative datum in the study of human psychology.

At the conclusion of the monograph, in section eight, a brief summary is presented. Those issues which emerge from this study as particularly salient are recommended for further investigation, and some of the prospects for future achievement are informally sketched.

2.0 CONCEPTS AND DEFINITIONS IN THE SCIENCE OF INTELLIGENCE

Anyone who has pursued study in cognition, artificial intelligence, or the philosophy of knowledge must share the common frustration stemming from the inadequacy of the terminology with which processes of thinking and intelligence are discussed. It is not the circularity of definitions relating these terms which alone is confounding. Such circularity of meaning is intrinsic in language. Rather, it is the concomitant lack of precision and of referents in these terms which precludes achievement of operational and testable descriptions of intelligent processes.

In order to rectify the serious condition of our terminology, it will be necessary to set forth new and precise meanings for the important concepts which we must negotiate. In doing this, some friction with pre-existing habits is to be expected. It is not easy to employ familiar words in narrow and novel ways, but it is essential to this endeavor. To the extent that we can agree upon the meanings of our words will we be able to argue profitably about the observations we describe. In what follows, it is not my intention to suggest that other nuances and connotations associated with these terms are without significance. I wish simply to restrict these words to particularly precise meanings for the duration of this study. I have no doubt that as the processes of intelligence become more fully understood, the language will be properly extended and refined to accommodate these needs. In the interim, the

best approach is to put our words and meanings in the open in a way in which all can agree upon what is being said.

In the definitions and discussion which follow, certain assumptions are made. First, it is assumed that all machines and organisms discussed occupy the same environmental space, wherein all events and attributes which can be perceived are objectively confirmable. The motivation for this assumption is to avoid difficult epistemological issues which are not particularly important here. Second, the environment of an organism (machine) includes all things which are theoretically perceptible or denotable in the total space, including internal states of the machine itself or of other machines. Any perceptible or denotable condition can be considered an attribute of the environment. Within the real-world context provided by these assumptions, we can consider now the major concepts vital to our study.

2.1 Knowledge

Definition: Knowledge. Knowledge is a capacity to predict the value which an attribute of the environment will take under specified conditions of context.

Figure 2.1 here

Let me clarify this definition before offering examples. Knowledge is evidenced whenever an organism or machine produces information or reduces a priori uncertainty about its environment which is synthesized from other data and not simply contained within them. An attribute of the environment is any measure or

method of scaling which theoretically can be applied to the environment. Or, recursively, an attribute may be a measure based, itself, on one or more attributes. An attribute value is the specific measurement obtained in one assessment of an attribute. To predict is to produce an estimated value of an attribute which can subsequently be confirmed by direct measurement. Finally, conditions of context are the environmental attributes used as the basis for prediction.

Examples of knowledge are certainly plentiful. I know the name of my wife. Therefore, I am capable of predicting the name Fox from the attribute value my wife. Knowing how to add means a capacity to predict the sum of a set of numbers. A prediction of the next number of the sequence 0,1,2,3,4,5 reflects, at least, some knowledge of the natural numbers. The ability to predict the identity of the famous politician and President who was shot in Dallas is also knowledge.

From these few examples, several inferences can be drawn. Prediction, in the sense in which it is being used here, does not imply the foretelling of the future in any usual sense. Instead, the predicting organism simply presages an unknown attribute in advance of its confirmation. Thus, you can have knowledge of the past as well as of the present or future. This leads directly to our next definition.

Definition: Types of Knowledge. Knowledge can be divided among several types according to the relationship it shares with each. Types of knowledge are necessarily neither exclusive nor

denumerable. Each type of knowledge is an equivalence class of one or more capacities for predicting events. Each capacity in such a class relates to every other by virtue of a single common attribute.

Figure 2.2 here

What this definition suggests is that clusters of knowledge can be organized into distinct types whenever salient attributes arise which can act as the basis for classification. For example, knowledge relating to past, present, and future predictions about an organism can be divided among ontogenetic, diagnostic, and prognostic types of knowledge. The distribution of university courses and researches among different departments reflects a similar organization with respect to types of knowledge, e.g. mathematics, physics, and literature. As another example, the specialization of labor in a production process is related to the development and exploitation of diverse types of knowledge.

Less obvious means of organizing knowledge into types are possible. Because any attribute of the environment is a priori suitable as a basis for the formation of a type of knowledge, an individual's knowledge can be partitioned in numerous ways. For example, we may consider his ability to predict the behavior of others with whom he interacts as one type of "interpersonal" knowledge. His ability to predict how he himself will behave during interaction is another. The Freudian system of id, ego,

and superego provides an example of a particular taxonomy of knowledge about human behavior. Of course, both the validity and utility of this system are problematic. These considerations motivate the following definitions.

Definition: Event. Consider a specific attribute value prediction X_p with knowledge K. Suppose that the occurrence of an n-tuple E of attribute values $(X_1, X_2, \ldots, X_n)$ is sufficient to support this prediction by K. Further, suppose that no m-tuple $(X_1, X_2, \ldots, X_m)$ entirely contained within E is sufficient to produce X_p with K. In this case, the attribute set E is an event under K.

Figure 2.3 here

Definition: Perception. Consider, as above, a prediction X_p made with knowledge K. Consider the set of events P of all events E_i which support the same prediction X_p with K. The set P is a perception under K.

Figure 2.4 here

Definition: Error. Error is a measure of the difference between a prediction made with some knowledge and a measured observation of the environmental attribute being predicted. Any measure of the discrepancy is a priori suitable for this purpose.

Definition: Scope (Domain). The scope (domain) of a knowledge is a measure of the events or perceptions which can affect predictions made with that knowledge. The scope of knowledge is precisely the union of all events or perceptions under that knowledge. If all predictions made with a knowledge are independent of the environment, the scope of that knowledge is nil.

Figure 2.5 here

Definition: Range. The range of a knowledge is a measure of the universe of predictions which could potentially be made with that knowledge. The range of knowledge is simply the union of all predictions made under all events in the scope of that knowledge. When the knowledge is theoretically capable of predicting (without consideration of accuracy) all possible events of the environment, we say the knowledge is universal. Otherwise, the knowledge is limited or bounded.

Definition: Domain of validity. The domain of validity of a knowledge is the subset of events D in the scope of that knowledge for which any error of prediction is always within a prespecified tolerable class. Conversely, we say knowledge is valid over a domain of events if no prediction which it supports could result in an error which exceeds the bounds of some prespecified tolerable error class. The tolerable error may be specified as zero or none.

Figure 2.6 here

Definition: Degree. The degree of knowledge is a measure of the relative perfection of the knowledge. The measure is a proportion of two terms. The second term, the denominator, is a measure of the range of the knowledge. The first term, the numerator, is a measure of the subset of the range which includes those predictions which can only be supported by events in the domain of validity. When these two sets are equivalent, we say the degree of knowledge is perfect. Otherwise, the degree of knowledge is imperfect. When the degree of knowledge is imperfect, the difference between the two terms is a measure of the set of predictions supported by that knowledge which are always or may be occasionally in excess of the tolerable error.

Figure 2.7 here

Definition: Precision. The precision of a knowledge with respect to a specific purpose P is a measure of the relative adequacy of K for the purpose for which its predictions may be utilized. Consider a second knowledge T which is defined as the minimal or simplest knowledge required to predict perfectly the activity of the purpose P. That is, T is a logical equivalent of P and is isomorphic to the machine that executes the purpose P. Now consider the scope S of all perceptions of T. Each perception

represents those events that can lead to a distinct behavior of the machine associated with P. Now consider a mapping between S and the range R of predictions of K. This mapping shows which predictions by K can lead to which perceptions by T. If each prediction in R can be associated with at most one perception in S, we say K is precise with respect to P and T. Otherwise, K is not precise. The measure of relative precision is a proportion of two terms. The second term, the denominator, is a measure of the scope S of T. The first term, the numerator, is a measure of the subset of perceptions of S which are exclusively associated with predictions in the range R of K which meet the one-to-one requirement from R to S. In other words, the precision of K with respect to P is a measure of the proportion of perceptions which can be made without ambiguity in the performance of P or in its prediction by T.

Figure 2.8 here

At this point, we have introduced all of the essential concepts pertaining to knowledge. Several of these are relatively complex. The most outstanding feature of these definitions, however, is the quality of relativity which they reflect. In the study of knowledge, the only absolutes are the assumed objective measurements of rela attributes in the environment. Knowledge is then definable as an ability to predict one or more of these attributes from some others. There is no absolute measure of the

value of knowledge, because each prediction may carry significance for each of many different puposes. In fact, the precision of knowledge itself is a function of the use to which it must be put. To someone without taste buds, the distinction between lemons and limes is without import with respect to eating. The same person, however, may attach great significance to the same distinction in the laboratory or when purchasing fruit for a table decoration.

Many interesting relationships can be exhibited among these measures of knowledge when various assumptions about the source and distribution of events are made. These questions, however, lie beyond the scope of the current paper. Nevertheless, these defintions will prove useful in considering our special problem, the nature of intelligence.

2.2 Logic and the Concept of Rationality

In this section, I wish to introduce the major concepts pertaining to logic and rationality. It will be interesting to consider how several knowledges may be combined by a logic, and how, together, they may comprise a rational system.

Definition: Attribute and Environment.

An attribute A is a transformation applied to the current environment which produces a single valuation. The attribute may be considered as a functional mapping $A: \theta \rightarrow \theta$, where θ is the state of the environment. The domain of this mapping is called the scope of the attribute. The range is called the range or scale of the attribute.

Thus every attribute is based on the environment and is part of the environment. The current environment θ is, in fact, definable by the current value of all attributes: $A_1(\theta)$, $A_2(\theta)$, ... That is, an attribute of the environment is any possible mapping from some elements of the environment to some others. The environment is logically equivalent to the collection of all attribute values at any moment in time.

These definitions are, of course, tautological. I have said that the environment is the set of attributes of the environment and that an attribute of the environment is any function which maps the environment into itself. The definitions are not worthless, however. They lead to the following important theorem.

Theorem 2.1. An environment θ is identical to the set of all definable functions.

Proof. Assume the opposite is true. This implies the existence of at least one function $B: A \rightarrow A$, where A is some arbitrarily defined space and where B is not included in the environment. But by definition B is an attribute of all environments which include A. Meanwhile, A is an attribute of those environments which contain it, because we can define a reflexive function $A: A \rightarrow A$ where A(x) is defined as:

$$A(x) = \begin{cases} A \text{ if } x=A \\ \text{undefined, otherwise.} \end{cases}$$

Thus, the existence of the space A implies that B is an attribute of the environment, which contradicts the assumption and completes the proof.

Corollary 2.1.1. Any definable function is suitable as an

attribute of all environments.

A word should be said on the term definable. The whole structure which we are building depends on the assumption of an observable and measurable reality. Definable, in such a reality, is equivalent to operationalizable. Thus, any function which can be computed by any means is a potential attribute of the environment. This implies the possibility of hierarchically defined attributes which are compositions of several operations.

Definition: Procedure. A procedure is a rule for computation, i.e. it is the full specification of a transformation from one state of the environment to another.

It is possible to have a complete knowledge of a procedure. This means the capacity to predict every action of the procedure under all condtions of context.

Definitions: Equivalent and Subsumes. Consider a procedure $P:\psi \rightarrow \psi$ where ψ is part of the environment. Consider, also, some knowledge $K: \theta \rightarrow \theta$ which is capable of predicting every attribute of P and no more. In this case, K and P are equivalent. If K is in addition capable of predictions about θ which are not attributes of P, we say K subsumes P.

These definitions lead to the additional observation,

Fact: Any procedure $P: \theta \rightarrow \theta$ or any knowledge $K: \theta \rightarrow \theta$ of the environment θ is itself an attribute of θ. Conversely, any attribute of the environment is equivalent to at least one procedure.

Definitions: Operator and Operation. An operator is any

procedure. An operation is the transformation of the environment resulting from the application of a procedure.

Definition: Logic. A logic L is a set of knowledges which is equivalent to a set of attributes and procedures of the environment which are closed under the operations of the set. A logic is thus equivalent to a set of connected graphs. Each node represents an attainable attribute value. The connection between two nodes represents a possible transition from one state of the environment to another under a specific operator.

Figure 2.9 here

This definition of logic is somewhat unusual and deserves further consideration. Within our framework, any set of attributes is sufficient to define a logic if no transformation of an attribute leads to another attribute which is not included in the logic.

It is interesting to consider what would be required of a logic for it to be "rational" or "logical." Note first that each achievable attribute value (node) of the logic can be considered the end result of some procedure operating on other attributes. Further, each transformation between two nodes of the logic is equivalent to a knowledge that predicts it. If each of the knowledges thus implied is valid over the attributes which they transform, then the logic which embodies them is rational.

Definition: Rationality. Consider a knowledge K which is equivalent to a logic exclusively comprised of tranformations which

occur within the domains of validity of their associated knowledges. Such a knowledge K is said to possess rationality.

Several implications can be drawn from this notion of rationality. Rationality is a quality of logics which transform the observed environment into new attributes. The transformations of the logic must be operationalizable and their results confirmable. No transformation can result in a deduction which lies outside of the logic itself. As a measure of validity, an arbitrary degree of error is tolerable for each transformation within the logic. Thus, rationality is a quality of those logics which consistently avoid excessive errors. To the extent that rationality is violated by any knowledge, we may ascribe the quality of irrationality to it.

Definition: Recursion. Consider a logic L and an original attribute state S_1 contained in L. Suppose the tranformation of S_1 by an operation ϕ_1 leads directly to a new attribute state S_2 in L. Suppose however that the transformation of S_1 to S_2 also results in the occurrence of a new attribute S_1 owing to the specially changed conditions of context. The logic may in this case be applied recursively to the new occurrence of S_1. This is the process of recursion. Alternatively, the logic may transform S_2 by other available operations without recursion.

In essence, this definition allows a logic to transform a fixed portion of the environment at a time. When the conclusion of such a transformation is reached, the logic is recursively applied to another portion of the environment. This second portion

of the environment includes the conclusion of the earlier application as one attribute of the larger context.

Definitions: Proof and Premise. Consider a logic L and an unbroken path from one attribute state S_1 to another S_2. The initial state S_1 is called a premise of the proof of S_2. The proof of S_2 is an ordered set of operations $(\emptyset_1,\emptyset_2,\ldots,\emptyset_n)$ which result in the tranformation of S_1 into S_2. That is, $S_2 = \emptyset_n(\emptyset_{n-1}(\cdots(\emptyset_1(S_1))\cdots)$. Each state which is attained between S_1 and S_2 by a partial series of tranformations, including S_2, is a conclusion by deduction from S_1 in L. Any set of attributes T which are transformed to a set U by recursion in L is called the set of premises of the proof of U in L. The proof of U is the ordered set of operations formed by the concatenation of the individual transformations of each application of L in the recursion.

To illustrate the notions of logic and proof we will discuss a logic for Boolean algebra or the propositional calculus. This logic may be considered to operate on an ordered set of symbols including 0 (false), 1 (true), & (conjunction, and) and ~(negation). Any problem in this logic can be expressed in suffix functional form, where an operator (& or ~) is applicable to the operands 0 or 1 which immediately precede it. For example, 00& is equivalent to "false and false" which is false. The problem, 00&~1&~ is also equivalent to 0. The truth of this statement can be seen in the following stages of a proof:

$$\underbrace{00\&\sim 1\&\sim}$$
$$\underbrace{0\ \sim 1\&\sim}$$
$$\underbrace{1\ 1\&\sim}$$
$$\underbrace{1\ \sim}$$
$$0.$$

The complete recursive logic required to compute all problems which are expressed in this form is given in Figure 2.10. In this

---

Figure 2.10 here

---

logic the attributes which are sought constitute the set of states. Each attribute means "transform the first occurrence of this pattern, moving from left to right." For example, the pattern 00& means the first such set of symbols, found while going from left to right, in the problem environment. It can be seen that the proof of the state 0 from the premise set $00\&\sim 1\&\sim$ is $(\phi_2,\phi_5,\phi_4,\phi_1)$. That is, $0 = \phi_1(\phi_4(\phi_5(\phi_2(00\&\sim 1\&\sim))))$.

2.3 Models

An important concept in most discussions of intelligence is the notion of a model. Science abounds with models (of the atom, of homo economicus, of social systems) and these all share common defining properties.

Definition: Model. A model is a knowledge.

In what ways can it be seen that model and knowledge are synonyms? A model is supposed to be employed in making predictions about certain attributes.[1] For example, economists create a model,

homo economicus, with certain attributes and premises, in order to deduce conclusions about possible eventualities. In its use as a tool for judging which alternative outcomes seem reasonable or likely, the model is being used for prediction. Thus the model is part of some knowledge. It also embodies knowledge because it is itself a capacity for prediction. Thus, a model is fully equivalent to a knowledge.

Definition: Types of Models. Synonymous with types of knowledge.

When people discuss types of models, they are usually classifying knowledges on special types of attributes which are historically accidental in that role. For example, we consider the mathematical school of model building which provides a taxonomy of models closely paralleling the taxonomy of problems of simultaneous equations of several unknowns. Thus, there are linear, quadratic, and polynomial models; there are first-order (like industrial dynamics), second-order, and higher-order differential models; there are interaction and non-interaction models, of both stochastic and non-stochastic variety.

In psychology, we can see a similar preeminence of models which reflect discriminations among diverse academic views in that

---

1. The equivalence of these terms is evident in the following quotation from Minsky, "We use the term 'model' in the following sense: To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A." From M. Minsky, "Matter, Mind, and Models," in M. Minsky (ed.), Semantic Information Processing. Cambridge, Massachusetts: The MIT Press, 1968.

field. We do not find frequent disagreements about which mathematical type of model is most appropriate, because that is an unasked question in most psychological arguments. The really significant question for these people is which model--developmental, gestaltist, physiological, behavioral, cognitive, or psychodynamic--predicts what and how. Similarly, these psychological diversities have never made a significant impact on the development of mathematics.

The general principle which I have been illustrating above is fairly important. Fields of knowledge develop in order to predict particular attributes of importance to the people in that field. The models which they talk about are simply representations of the knowledge which they have constructed.

The study of functions is within the scope of mathematics. Thus, it is not surprising that different types of mathematical models have developed reflecting the major differences that have been specifically assigned to individual functions. The extrapolation of significant attributes from one problem to another seems characteristic of human problem-solving behavior, in general.

It is a fallacy to hold that models provide information which is not intrinsic in the events which they represent. A model, as a means of making predictions, is only as valid as its component predictive relationships. If a model makes a particular prediction and the premises on which it operates are even slightly erroneous, the prediction may be utterly incorrect.

From this can be seen at least two sources of value in models.

There is a familiar sense in which models are valuable. That is, a model is valuable if it makes predictions which are useful for some purpose. But there is a concomitant issue related to the scope and structure of the model. In this sense, models may be valuable if they are simply interesting or provocative by virtue of the attributes on which they operate. For example, the model of the steam engine played a significant role in the development of other models, e.g. man, calculators, and institutions. In this case, the value of a model of the steam engine derived in part from the other sets of attributes which could be considered in a similar way to the set of attributes of the steam engine. That is, the potential for analogy between the steam engine and other engine systems increased the value of that model.

The consideration of the structure or interrelationships within a model is the aim of all fields of study. However, no field of study has a monopoly of useful models. If the steam engine is a useful model of the workings of a man, then a man is a useful model of the workings of a steam engine. The genesis of models is an important issue which we consider often throughout this monograph.

2.4 Motivation and Drive

To this point, little has been said of the apparently universal aspect of organisms related to their purposiveness or goal directedness. It may have seemed that organisms were supposed to possess complete discretion of choice and behavior. It might seem that such an organism could be designed. These ideas motivate the

following definitions.

Definitions: Motivation and Drive. Consider an organism O whose behavior is completely predictable by some knowledge K. For any state of the environment θ, the organism will behave in a determinisitc way and execute a certain behavior $B(\theta)$. Consider any other potential behavior of a similar organism, $B'(\theta)$. To the extent that O acts in way $B(\theta)$ as opposed to $B'(\theta)$, we say that O is motivated to act according to B as opposed to B'. Any function of the difference between B and B' can be called O's drive to achieve B or O's aversion to B' with respect to B.

As an example of drive we may consider the commonly cited sex drive. For some psycholanalysts, the sex drive is considered inherent in a person and is held responsible for all sorts of behaviors, including lust for one's mother or father. For ethnologists, sex drive is tantamount to the tendency for reproductive and nest-building activites. For organized religion, a sex drive is something that is alternatively recognized, denied, condemned or ordained. In all of these, the notion of a drive for sex is a measure of the supposed tendency of organisms to behave sexually vis-a-vis a particular sexless model.

Definition: Reinforcement. Again, consider the organism O predictable by K. Suppose O perceives some attribute $A(\theta)$, meaning that $A(\theta)$ is a perception of K. Further, suppose that O behaves in such a way as to repeat a behavior $B(\theta)$ which occurs whenever $A(\theta)$ is perceived. In this case, we call A a reinforcement for O. The perception of $A(\theta)$ reinforces the behavior B.

I wish to clarify the word repeat used in this definition. Repeating a behavior means exhibiting two behaviors which are measurably similar in two contexts separated necessarily by a lapse of time. In this definition, attention is also to be paid to the role of the perception of $A(\theta)$. The perception need not immediately precede the execution of $B(\theta)$ or coexist with it or follow it. Much of the work of experimental psychology is concerned specifically with determining the nature of the relationship between $B(\theta)$ and $A(\theta)$. $A(\theta)$ may, for example, be an internal state of the organism.

Definition: Punishment. Again, consider O predictable by K. Suppose O perceives some attribute $A(\theta)$. Further, suppose that O behaves in such a way as to repeat a behavior $B(\theta)$ whenever $A(\theta)$ is not perceived. In this case, $A(\theta)$ is a punishment for O.

Definition: Goal. A goal is a reinforcement.

Definition: Incentive. An incentive is any attribute $A^I(\theta)$ which is perceptible under some conditions as a reinforcement $A(\theta)$, that is, is equivalent to a perception of $A(\theta)$ for some values of θ.

This definition clearly embodies the common meaning of incentive. It is useful in an explanation of the failure of some incentives to have the desired reinforcing effects under all circumstances.

Definition: Drive for Predictability. Once again, consider O predictable by K. If O is reinforced by a perception in K that O itself has a made a valid prediction, O can be said to possess

a drive for predictability.

Definition: Drive for Complete Predictability. Consider O predictable by K. If O is reinforced by a perception that O itself has validly predicted all perceptible events, O possesses a drive for complete predictability.

Definition: Drive for Economy. Suppose O, as above, is predictable by a knowledge K which is, in turn, representable as a particular computational procedure P. Let O exhibit some behavior A' which is measurably identical to another behavior A for some observer E (A and A' are perceived identically by E). If A' can be executed by P in less time than A and if O is reinforced by the execution of A', O can be said to possess a drive for economy of time with respect to A, A', and P.

The notion of a drive for economy is extendible to other attributes than time, such as memory, energy, and weight. However, it should be clear that the notion of economy is relative. It is very difficult to imagine even generally desirable drives for economy in an organism--consider love-making or reading activities. As many organisms are combined to form a society, legislated societal drives for economy are more easily comprehended if intra-organism differences are ignored.

2.5 Concepts, Ideas, and Ideogenesis

I have given considerable attention elsewhere to the nature of concepts in human activities[1], but the notion of concept in the

---

1. F. Hayes-Roth, The Structure of Concepts. Working paper of the Sloan School of Management, Cambridge, Massachusetts, April 1971.

abstract framework we are now creating is somewhat different.

Definition: Concept. A concept is a procedure for transforming the environment into a perception of the environment or for transforming one perception of the environment into another. The scope of the concept is the scope of an equivalent knowledge.

We have already shown in section 2.2 that the environment is the set of all computable functions. Consider a knowledge K which is incapable of predicting all attributes of the environment. A concept in K is a rule which explains how perceptions in K relate to and are produced by other aspects of the environment. For example, my concept of large is a rule which explains my perception of large. This explanation may involve interactions among aspects of the environment which are individually imperceptible to me.

Also, concepts may be reflexive transformations. For example, I have many concepts of concepts. The concepts noun and verb are applicable to specific verbal concepts, such as to the nouns apple and speed and to the verbs fructify (to cause to bear fruit) and speed (to move very quickly). Because verbal concepts--of which noun, verb, apple, speed, and fructify are examples--are the only relatively precise tools of daily interpersonal communication, they have a significant impact on human experience which is not possible for non-verbal concepts. However, it would be an error to assume that significant conceptual transformations depend on the existence of corresponding words. That supposition, although frequently encountered, is total unreasonable in my opinion.

Definition: Idea. An idea is either a concept or the perception

of an event in the scope of the concept.

This definition of idea is intended to cover its two most frequent meanings. The meaning of idea as concept has already been explained. The meaning of idea as a perception is easily illuminated. Consider an organism O which is predictable by K. If an environmental event occurs which is in the scope of a concept of O, then O will transform the perceived event by utilization of the concept. Thus, if a dog appears in the environment, O can have the idea dog if he perceives it. This does not mean that he must see or hear it, but that he has invoked one of the same procedures (concepts) which he would if he did actually see or hear it. Thus, reading the word "dog" for an organism which reads, sees, and knows the concept (like the reader, for example), results in an experience which is identical in some respects to the visual recognition of a dog as dog.

We consider now the issue of the generation of concepts. In this definition, we will establish a critical and salient attribute of intelligent systems.

Definition: Ideogenesis (Concept Formation). Consider an organism O predictable by K. Ideogenesis (concept formation) is the creation of a new concept in O.

Ideogenesis is the process whereby an organism changes the way it responds to the environment. When a new concept is formed, the organism may have perceptions which are totally unlike those it had before. Remember that a perception is the set of all environmental events that are transformed in an identical way.

Thus, when a child forms the concept dog he has developed a capability which would allow him to respond identically to all dogs. That he does not respond identically to all dogs indicates only that the knowledge which would predict his activity must be more complex than simply predicting whether or not something is perceived as dog.

Definition: Ideogenetic Drive. Again, consider O predictable by K. If O can be observed to exhibit ideogenesis, it is possible to define and measure the various attributes of this behavioral tendency. These operations can constitute operational definitions of ideogenetic drive.

The relevance of ideogenetic drives can be seen in a simple hypothetical problem. Suppose you were required to choose between two machines X and Y for application to an industrial process. Both machines are originally programmable for a specific simple task, and one of the machines, say X, is capable of ideogenesis regarding its task environment. Clearly, if X is capable of generating valid and useful concepts to correct its performance, it is preferable to an unchanging Y. This is just the case when X and Y are thermostatic controls, and when X is additionally capable of maintaining a valid temperature calibration independent of any erroneous readings on its dials. The alternative thermostat Y may be thought of as gradually and unalterably losing its calibration.

The variety of concepts which people possess suggests a wealth of questions about ideogenesis. For example, we may wish to discover how people form specific concepts or concepts in general,

which concepts facilitate or inhibit the learning of other concepts, and what the developmental concomitants of ideogenesis are. All of these questions are studied currently in various fields of psychology, although frequently indirectly.

For my purposes, it will suffice to define three additional terms related to ideogenesis. In these definitions, the major psychological issues can be found, albeit somewhat transformed.

Definition: Abstraction. Consider an organism O predictable by K. Consider also a set of O's perception P = $(P_1, P_2, \ldots)$. If each of the perceptions P_i in P is transformed into a common perception Q under a series of conceptual operations, we say Q is an abstraction of P. We may say that O has abstracted the property Q from P.

Definition: Developmental context of ideogenesis. Consider O as above. Consider, further, the set of all possible environmental events C which could accompany, simultaneously, the formation of a new concept I. This set C is called the developmental context in O of the ideogenesis of I.

Definition: Induction. Consider O as above. Suppose that O has abstracted some attribute Q from a set of perceptions P which is potentially more extensive than P--i.e., the set of possible perceptions S exhibiting Q properly contains P, the set from which Q was abstracted. The ideogenetic formation of a concept which transforms S into Q is called induction by O. We say that O has induced Q about S from P.

I worry that this proliferation of precise terms may be confusing

the relatively straightforward relations which they are intended to signify. In the case of induction, it is essential that the reader grasp the basic processes involved. I will illustrate these with an exaple.

Suppose a child knows the concept of number only to the extent that it can properly identify the single digits 0, 1, 2, or 3 as cases of number. Suppose, further, that when queried by an observer the child first denies that the pairs 00, 01, 10, and 11 are numbers. Now, suppose the child is provided with corrective feedback from the observer to the effect that, "You are wrong; 00, 01, 10, and 11 are all cases of number." If the child can subsequently generate a concept of number that permits him to identify all of the symbols 00, 01, 02, 03, 10, 11, 12, 13, 20, 21, ... as numbers, we say he has induced the allowable property of two-digit-ness of numbers.

We cannot, however, deliver a certain explanation of what he has induced without discovering what all the ramifications of his ideogenesis are. One might have alternatively said in explaining his behavior that "he has induced the fact that two numbers placed side by side constitute a number." Actually his behavior in recognizing novel two-digit numbers as numbers is consistent with this explanation, but it is insufficient to justify it. If this alternative explanation were valid, the child would be able to recognize four-digit numbers also, because they are formed from juxtaposed two-digit numbers. By the same reasoning, the child should be able to recognize any number of adjacent digits as a number. That is,

of course, the correct concept of number which is ultimately sought. No one who has studied children, however, would be surprised if the child could not recognize all possible numbers correctly after his first induction. In many cases, induction does not lead to complete and valid knowledge even if it produces a few correct responses to novel situations[1].

2.6 Conceptualization and Problem-Solving

Definition: Conceptualization. Conceptualization is both the process and result of the transformation of the environment (including perceptions) under concepts. Conceptualization is the application of conceptual procedures to the environment.

Definitions: Thinking and Thought. Thinking is that part of conceptualization which is restricted to the process of transforming the environment under concepts. A thought is the result of thinking. An organism which thinks is said, alternatively, to conceive.

Definition: Problem-Solving. Suppose an organism O conceives of a possible attribute of the environment $A(\theta)$ which is theoretically realizable although not currently perceptible by O. O may think of a means for realizing that attribute or causing $A(\theta)$ to become perceptible. O's thinking in this case is called problem-solving. The series of procedures which constitute the means of

---

1. For a detailed analysis of the stages of development of a concept, with regards to such partially valid inductions, I recommend Vygotsky's wonderful research. See L.S. Vygotsky, Thought and Language. Cambridge, Massachusetts: The MIT Press, 1962.

instrumentally altering the environment to cause the attainment of A(θ) is called the solution to the problem A(θ).

From this definition it is evident that many organism, other than people, think and solve problems. For example, my dog thinks about where her ball is. I observe this because she executes a rather extensive household search (limited, of course, to her usual haunts) when I present the problem to her, "Go get your ball." The remarkable chimpanzees of Kohler's insight experiments must also think a fortiori[1].

It follows from these definitions that problem-solving behavior is sufficient evidence of both thinking and the existence of concepts. Whether or not the problems which organisms solve or the concepts with which these are solved are in some way restricted to a particular type was a motivating question behind Gestalt psychology. That is, a Gestalt principle is nothing more than an attempted statement of those properties of the whole context which are prerequisite to successful problem-solving with the problem parts. In this sense, a chimpanzee may be able to use a stick as a tool in one particular setting, but unable to conceive its use in a similar role in another context. This Gestalt attribute of the tool-in-setting may, by similar reasoning, be defined in terms of a failure on the part of the chimpanzee to abstract the utility of the stick from the stick-in-setting.

---

1. See W. Kohler, Intelligenzprufungen an Menschenaffen. Berlin: Springer, 1917. Translated, The Mentality of Apes. London: Kegan, Paul, 1924.

2.7 Learning

Many definitions of learning already exist. Learning is one of the few topics we cover that has been extensively studied from a comparative and scientific viewpoint. Nevertheless, we will find a slightly more abstract definition than the common one quite useful.

Definition: Learning. Consider an organism O predictable by an equivalent knowledge K. Any change in behavior directly related to the application or withdrawal of reinforcement or punishment is learning. Such learning will be manifested by changes in subsequent predictions in K.

Reinforcement and punishment were defined (section 2.4) as perceptions which supported and negatively supported repetitions of behaviors, respectively. Learning then is any measure of the tendency to repeat a different behavior after punishment or a similar behavior after reinforcement.

Excluded from this definition are changes in the organism which are not related to reward and punishment. These may include changes produced through growth, cell mutation, or simple behavioral aberrations which do not tend to be repeated.

One might wonder, given the circularity of these definitions, why learning is significant. Clearly, the knowledge K which perfectly predicts O's behavior can predict any change in behavior which O will undergo. Why, then, specially characterize those which relate to reinforcement and punishment? The answer lies in our wish to build, design, and comprehend teleological (purposeful)

systems. That is all. It is not difficult to see that a teleological organism is also a physical system and that all learning could therefore be explained as mechanical adaptation without reference to the organism's goals which are essentially intermediate variables in our model. This type of explanation is actually preferable in some ways. Nevertheless, the concepts of goal and purpose can help clarify the issues faced when considering alternative mechanisms of feedback in the design of such a system.

From this definition of learning, it can be seen that there are numerous possible measures of learning behavior. Changes in accuracy, speed, latency, duration, and frequency of behaviors are all measures of learning. Notice that each of these measures may, without difficulty, be directly transformed into a measure of repetition or probability of occurrence of some attribute of the responding organism. For example, increases in accuracy at a task may be defined in terms of the increased probability of issuing a perfectly accurate response.

2.8 Intelligence

We have reached the first major objective. Armed with the system of concepts just introduced, we can set forth the following important definition.

Definition: Intelligence. Consider an organism O predictable by an equivalent knowledge K. First, suppose O is at any moment capable of computing some attributes of the environment. That is, suppose O perceives through senses which provide measurements of

the environment. Second, suppose O is theoretically capable of performing any and all abstractions possible within the limitations of its environment sensing apparatus. Third, suppose O is capable of an ideogenesis incorporating any abstraction. That is, O must be capable of performing and applying induction. Fourth, suppose O possesses a drive for complete predictability. In this case, we say O possesses intelligence and that O or K is intelligent.

This definition immediately motivates the ones which follow.

Definition: Components of Intelligence. The components of intelligence are: (1) perception, being sensation plus contingent conceptual behavior; (2) a capacity for abstraction; (3) a capacity for ideogenesis; and (4) a drive for complete predictability without error. The second and third components may be considered, together, as the capacity for induction.

Definition: Endowment. Consider an organism O. The combination of any of the four components of intelligence which are present in that organism at some moment, usually at its inception, is called its endowment.

From these definitions it can be seen that there are many possible endowments which do not qualify as intelligence. Some of these may be supplemented during the life of the organism in such a way as to transform an unintelligent organism into an intelligent one. For example, we can imagine an organism incapable of any ideogenesis at its inception subsequently being altered in such a way as to overcome this deficiency.

It is debatable, of course, whether this definition of intel-

ligence is the best for all purposes. I would hold only that it is both useful and testable. It is valuable as a guide in an inquiry into intelligence, because it isolates attributes of systems which are frequently observed but rarely found in full complement. We may thus pinpoint the deficiencies and strengths of alternative systems. Further, we are now in a position to ask other related questions which depend utterly on a precise knowledge of intelligence.

Before exploring other implications of this definition, it may be desirable to relate some of the previously discussed topics to intelligence. Note that any intelligence contains some logic, because it is a closed system of transformations on perceptions. The operators of the logic are the concepts of the organism. The scope of an organism's knowledge is limited to the set of perceptions which it can have. The range of its knowledge is the set of all predictions it can make.

An intelligent organism must be capable of performing all abstractions and ideogeneses: it is capable of universal inductions. Every function of the environment may be considered as just that property which is induced from a set of observations of that function over a limited domain. Thus, an intelligent organism is capable of predicting all computable functions of the environment. For this reason we state that the knowledge which is potentially attainable by an intelligent organism is universal and unbounded. If the environment of an intelligent organism includes finite sequences of zeroes and ones which can be sensed, altered,

and shifted one digit at a time by the organism, that organism can compute all computable functions of all Turing machines and subsumes a universal Turing machine. Any knowledge that subsumes that of such a universal Turing machine is also universal.

Definition: Cause. Any set of attributes, A, the observation of which is sufficicient for the prediction of an event E is a cause of E.

Definition: Explanation. Consider an organism O predictable by an equivalent knowledge K. Suppose O discovers by induction a set of events E each of which is predictable by an attribute A. In this case, O induces that A causes E. We say that A is an explanation of E in K.

Definition: Hypothesis. Consider an organism O predictable by an equivalent knowledge K. Suppose A is an induced explanation of E in K. In this case we may say that O hypothesizes that A causes E. An hypothesis is an explanation. This hypothesis is denoted by $A \rightarrow E$ or $E \leftarrow A$.

One of the implications of this discussion is that intelligent organisms may have many explanations for the same event. These explanations may or may not be mutually consistent with respect to the range of predictions which they support. For example, the organism may need to predict the next term in the sequence 0,1,2,... . If the organism explains the set (0,1,2) as a series of integers beginning at zero and increasing by 1, it will subsequently predict a 3. However, if the organism thinks that the sequence 0,1,2 has a periodicity of length three, it will predict another 0. If it

thinks alternatively that the sequence is a representation of increasing natural numbers which are not trivially factorable (prime numbers including zero), it may also predict a 3 or perhaps a 5, 7, or 11. Needless to say, since these are all effectively computable sequences, each is an explanation that an intelligent organism could possess and employ in this problem. Each explanation is an induction from the set (0,1,2).

The resolution of this problem lies not in a limitation of the possible explanations which an organism can possess. The fact is that each of these explanations is equally valid although mutually inconsistent with respect to the next predicted digit. An intelligent organism learns to make correct "guesses" in such situations by abstracting other properties of the problem context which are of relevance. In this particular problem, most humans in our culture will respond with the digit 3, which is produced in accordance with the first suggested explanation. If the problem is restated in such a way that the objective becomes the prediction of the most common response among humans in this culture, 3 is the correct answer.

Theorem 2.8.1. An intelligent organism may hold an infinite number of inconsistent hypotheses which are valid over the entire domain of its experience. These hypotheses will lead to different predictions of the future.

Let us suppose that the capacity of an intelligent organism to retain hypotheses is limited. We know that the organism is

sensitive to reinforcement, because it has a drive for complete predictability. Thus, let us suppose that the organism behaves in a way which maintains over time any of its finite number of explanations which are consistent with its observations. Suppose also that it replaces those which are no longer consistent with its observations by others which are. Suppose also that we directly control the environment which provides this organism with observable data.

Now it is interesting to consider the learning behavior of this organism as viewed over time. Clearly, what it learns and possesses as hypotheses at any moment is a function of the events we have presented to it, the reinforcing rule we have employed, and its internal structure. If we had employed either a different sequence of events or another reinforcement schedule, we would as likely as not have produced an organism with different hypotheses, hence with different knowledge.

Definition: Training Sequence. The continuous set of observable events in the environment including the reinforcement associated with these constitute the training sequence for an organism.

It follows from what has been said that the knowledge K possessed by an intelligent organism O with fixed endowment is a function of the training sequence only. This holds whenever the endowed capacities are fixed at its inception and are subsequently never altered exogenously.

2.9 Consciousness, Mind, and Time

The last concepts to be discussed are those most closely pertinent to the human experience of existence or awareness. It was implicit in the previous definitions that awareness of oneself was unnecessary for the acquisition of intelligence. Nevertheless, in this section I will attempt to briefly describe how awareness is operationalized. The implication, that a machine can be built with similar properties, is straightforward.

Let us consider first the concept of time. We have relied upon various qualities and existence of time implicitly in some of what has already been said. For example, the concept of reinforcement was dependent upon the measurement of a behavior repeated over time. But what is time? I propose the following definition.

Definition: Time. Consider an organism O, in environment θ, which is predictable by an equivalent knowledge K. Suppose O possesses two different hypotheses about the environment which are not inconsistent with its training sequence ϕ, where ϕ is one possible value of θ. Call these two hypotheses $A(\theta)$ and $B(\theta)$. The validity of these hypotheses over the domain of the training sequence is sufficient for the condition $A(\phi) = B(\phi)$. That is, both hypotheses fully explain past events consistently. However, suppose these two hypotheses support different predictions for some other states of the environment, that is, $A(X) \neq B(X)$ for some X which is a possible value of θ which is not included in ϕ (i.e., $X \in \theta - \phi$). The domain of validity for

the two conditions (1) if $A(\theta)$ is true then $B(\theta)$ is false and (2) if $B(\theta)$ is true then $A(\theta)$ is false is called time in the framework of K. Time is thus a partitioning of the environment with respect to the conditions (1) and (2).

This definition states that time is the condition which precludes the possibility that opposing predictions are both true. Thus, time is for us simply a constraint on the validity of knowledges. Knowledges which fail this constraint are invalid.

I suppose this definition is a bit bland compared to the personal experience of time. Time does not appear to have this simple logical structure. Rather, time seems to flow smoothly past my senses like the sand in an hourglass. Of course, my logic is to be preferred to my sensation as a modality of communicating to others. A little reflection will show how the condition of the sand in the hourglass is equivalent to the state of the environment with respect to time. Because the perceptible states of sand in the hourglass are dependent upon the volume of sand in either side, one's knowledge of the sand is contingent upon those relations which can validly describe the pebbles in any specific volume. Change the volume of sand and you accordingly change the world with the hourglass. No two logically complementary measures of the volume can both obtain. The world is divided among those things which are confirmable and their negations which are not. Any organism which perceives this condition can be said to perceive time.

In this respect, we can explain the relativistic effect of time dilation as the product of a change in the logical constraint on a system. Consider a stationary observer O and a system S which is speeding at some velocity close to the speed of light with respect to O. The apparent expansion of time in S with respect to O is representable by a modification of the constraint on mutually exclusive predictions in O. In fact, valid predictions made by O about events in S can be considered invalid for O unless account is taken of the dependence between the validity of a measurement and the frame of reference in which it is used. The relativistic nature of all knowledge is thereby assured.

Figure 2.11 here

Definition: Consciousness. Consider an organism O predictable by an equivalent knowledge K. Suppose O induces a property about O itself and ideogenetically forms the corresponding concept. The application of this concept to the environment is called O's consciousness. When O thinks with this concept, we say O is conscious.

From this definition it can be seen that consciousness is tantamount to self-awareness. These are both terms for the property of an organism which considers itself as part of the problem environment. The existence of consciousness can be inferred for any presumed intelligent organism which is capable

of predicting the consequences of its own instrumental acts which alter the environment.

Theorem 2.9.1. Any intelligent organism O, predictable by an equivalent knowledge K, which alters the environment by an instrumental response predictable in K and perceptible by O will attain consciousness with some particular training sequence. However, there are training sequences which would preclude the attainment of consciousness by O.

I will conclude this chapter with one final definition which may prove helpful.

Definition: Mind. Consider an intelligent organism O predictable by an equivalent knowledge K. If O possesses consciousness, we say K is the mind of O.

3.0 A PRIMITIVE INTELLIGENT MACHINE

In the current chapter, I will describe a machine which I believe is the first designed which possesses intelligence. The machine is of little immediate value to us as a practical device, but I have no doubt that it will serve as a useful model for many inquiries into the subject. The machine is designed for implementation on a computer although the coding as a program has not yet been performed[1]. However, it is not the actual performance of the program that is of interest. Rather, the value lies in the structure and design of the machine. It would, of course, be useful for some purposes to see the program actually perform, but my goal here is to relate the findings which the design itself provides. Therefore, I will introduce the design characteristics first and subsequently discuss the behavior of such a machine with respect to a variety of problems.

The reader will see in the current chapter how many of the concepts explicated in the last chapter reach full operationality in the machine as they are included as programs in its endowment. In this chapter, operationalizable means programmable. It is my belief that all of the concepts of chapter 2 are programmable. Although this particular machine does not embody all of them, the reader should sense the fact that such a goal is realizable.

---

1. As the reader gains familiarity with the machine, it will become apparent that the design makes parallel processing of concepts highly desirable. In fact, an interrupt driven computer would be preferable to most common computers as an implementation device. Alas, these considerations must be put aside temporarily.

3.1 The Problem Environment

I will refer to this machine simply as P. The environment in which P operates must be fully described. Without being too misleading, we could describe P as a computer program to predict terms in symbolic sequences. A sequence of symbols is fed to P via a computer input device. P attempts to develop an explanation of the behavior of the sequence and predicts the next set of symbols. The trainer who has provided P with the original input sequence subsequently "judges" P's behavior by supplying it with the correct response. If P's prediction is the same as the correct response, P adapts in such a way as to increase the probability of identical behavior on its part in future similar circumstances. If P's prediction is incorrect, P also modifies its behavior in an attempt to reduce the probability of error in the next similar stimulus situation.

The problem environment can be represented symbolically by isolating several variables which P can sense. First, there is I, the input sequence of symbols. These symbols are read from left to right, and each "word" is separated from the surrounding ones by an intervening blank space.

In addition to sensing I, P is able to generate and sense its prediction, N, of the next symbols in the sequence. P's prediction can be thought of as a single word or number. P can also sense the time T at which the input I was read and ΔT, the total amount of elapsed time before P made its prediction. The reader can consider T to be the reading of an eternal interior

computer clock when I is read. ΔT can be considered the reading of another short-term clock which is restarted every time I is read.

P must be able to sense the trainer's response R to P's prediction N. If P predicts N=5 and the trainer responds with R=5, P senses that its prediction was correct. If R=4, P can sense a difference between N and R and can utilize R in the development of new behaviors.

There are other properties of P's internal environment which P can sense. However, a full understanding of these can be gained only through detailed study of P's structure and operations. These are the topics of the subsequent sections.

3.2 The Conceptual Endowment (Logical, Mathematical)

The endowment of P is a fairly rich one. Not only does P sense I, N, R, T, and ΔT, but P posesses many additional concepts which guide its behavior. These concepts are the subroutines of the computer program which would be necessary to build P. Furthermore, P possesses an extensive memory and several drives, including drives for predictability and economy. In this section, I will provide a detailed description of P's logical and mathematical conceptual framework. This task will be somewhat arduous, but the clarity of meaning provided by the technicality of these descriptions should assure that the entire effort is a fruitful one.

There is a recurring question of notation which should be introduced at the outset. The foremost requirement of any notation

is that it effectively communicate the operations which it describes. For this reason, I will frequently employ an informal descriptive meta-language based in normal English which is complemented by familiar logical or mathematical symbols where needed. However, both a formal rigorous language for these operations and a comparable programming language are available and will be introduced. Where it is illuminating, the formal description of a concept will be given. Any actual implementation of P would necessitate the development of programming descriptions for all concepts which operated on their formal notations as data. This is not needed for the purposes of the current paper. The reader may be assured that every concept, regardless of the informality of its description, is fully describable in both formal and programming notations.

Each concept to be introduced represents a computer subroutine that recognizes valid occurrences of the principles which it embodies. Some of the concepts, in addition, are pro-active in that they seek out supporting evidence in P's memory. These, too, represent programs but may be considered as goal-oriented procedures which recognize an occurrence of a concept when they find one. We can now proceed to introduce P's endowment.

Concept: Symbol. A symbol is any of the characters that can validly appear on the input I. Thus, a symbol may be a blank, any digit, any letter, or any mathematical symbol. We may write this formally, using =df to mean "is defined to be", as follows:

symbol =df 0/1/2/3/4/5/6/7/8/9/a/b/c/d...

In this formal notation, the slash separates equally possible alternatives. It can be read as "or".

Concept: Set (Ordered Set, List). For the purposes of this machine, a set, list, and ordered set are synonymous. A set is a collection of denumerable elements x_1, x_2, x_3,... . We denote the set S of such elements by writing S =df $(x_1,x_2,x_3,...)$. These elements are ordered and constitute a sequence. The first member is x_1, the next is x_2, and so on. A set may also be a collection of sets. A set of only one element may be written without parentheses.

Formally, set =df $(x_1,x_2,x_3,...)$ where each x_1 =df symbol/set.

Concept: Word. A word is any sequence of symbols, taken left to right, surrounded by blanks on the input I.

Concept: Digits. The digits are a set D =df (0,1,...,9).

Concept: Number. A number N is a finite set of digits taken right to left. For example, the number 103 is the set (3,0,1). The direction is chosen here as a convenience for P only.

Concept: Relation. A relation r is a set of sets $(r_1,r_2,...)$ where r_1 =df $(x_{10},(x_{11},x_{12},...,x_{1n}))$. x_{10} is called the conclusion (attribute value) of the relation r under the premise (condition, argument) which is $(x_{11},x_{12},...,x_{1n})$. It is possible to consider a relation as a set of two-tuples, r =df $((c_1,p_1),(c_2,p_2),...)$ where each c_1 is the conclusion of the premise p_1. Relation is synonymous with rule.

Concept: Well Formed Formula (WFF). A wff W is an ordered set of valid relations where the conclusion of each relation is the

premise of the next one. W =df $((c_1,p_1),(c_2,p_2),\ldots,(c_n,p_n))$. W is a wff if and only if (c_i,p_i) is a member of some valid relation (for all i) and if $p_i=c_{i+1}$ (for $i=1,2,\ldots,n-1$). Describing W we may say that p_n is the premise and c_k is the conclusion ($k=1,2,\ldots,n$). If each (c_i,p_i) is presumed to be a member of some relation r_i, the set $(r_1,r_2,\ldots,r_n)$ is called a proof of c_1 under condition p_n.

Example: Let W =df ((happy,healthy),(healthy,well-fed), (well-fed,wealthy)). If each of the relations in W is valid, it follows that wealthy people are happy. The proof of this statement is that wealthy implies well-fed, well-fed implies healthy, and healthy implies happy.

Concept: Assumption and Axiom. An assumption is a relation presumed to be valid. Assumptions may be defined relations in which case they are called axioms. They may also be relations which are induced from partial knowledge and which are presumed valid in novel circumstances of similar structure. Axioms are assumptions of the first type. Induced relations are assumptions of the second type.

Concept: True. Any axiom S is true. We write this as the wff (true,S) or equivalently (S,true). Induced relations are presumed to be true. If T is an assumption of this type (second), we write (p-true,T) or (T,p-true) both of which are valid wffs.

Any definition is expressible as a valid wff. If P =df Q, both (P,Q) and (Q,P) are valid wffs.

Whenever S is true, we may informally write S as an abbreviation for (true,S). The conclusion of any wff composed exclusively

of valid relations is either true or p-true. wffs are true if composed solely of true relations and p-true otherwise.

Concept: Negation ($\sim$). Negation is a relation which changes the truth value of a wff. We define negation by the following axioms.

$$\sim W =df \begin{cases} \text{false if W is true} \\ \text{true if W is false} \\ \text{p-false if W is p-true} \\ \text{p-true if W is p-false} \end{cases}$$

If any of the component relations in a wff is only p-true, the entire conclusion is p-true as stated above. Therefore, it will not be necessary to define the remainder of the relations both in terms of true and p-true values. In the following definitions, p-true and true are synonymous. The conclusions will differ according to whether or not one of the presumed true premises is actually p-true as explained.

Formally, negation is indicated as ($\sim$,W).

Concept: Conjunction (&). Conjunction is a relation which composes the truth values of two wffs.

$$A \ \& \ B =df \begin{cases} \text{true if both A and B are true} \\ \text{false, otherwise} \end{cases}$$

This can be written in formal notation as (&,A,B) or equivalently (&,B,A) where true =df ((&,A,B),(true,A),(true,B)).

Concept: Disjunction (/).

$$A \ / \ B \ =df \begin{cases} \text{true if A is true} \\ \text{true if B is true} \\ \text{false, otherwise} \end{cases}$$

In formal notation, (/,A,B) or (/,B,A).

Concept: Conditional ($\Rightarrow$).

$$A \Rightarrow B \ =df \begin{cases} \text{true if B is true} \\ \text{true if A is false} \\ \text{false, otherwise} \end{cases}$$

In formal notation, ($\Rightarrow$,A,B).

Concept: Biconditional(≡).

$$A \equiv B \quad =df \begin{cases} \text{true if A and B are both true} \\ \text{true if A and B are both false} \\ \text{false, otherwise} \end{cases}$$

We will use the notation ≡ to mean "if and only if". In formal notation, A≡B is written (≡,A,B) or (≡,B,A).

Concept: Null Set (Ø). The null set is the set of no members.

Concept: Element (Θ).

$$x \Theta Y \quad =df \begin{cases} \text{true if x is a member of the set Y} \\ \text{false, otherwise} \end{cases}$$

If x Θ Y, we write formally (Θ,x,Y).

Concept: Subset (C).

$$X \subset Y \quad =df \quad (a \Theta X) \Rightarrow (a \Theta Y)$$

It is, additionally, an axiom that Ø C X, for any set X. Formally, we write (C,X,Y).

Concept: Equality of Sets (=).

$$X = Y \quad =df \quad (XCY \ \& \ YCX).$$

Formally, (=,X,Y) and (=,Y,X) are equivalent.

Concept: Ordered Equality (=*).

$$X =^* Y \quad =df \begin{cases} \text{true if both ordered sets are identical} \\ \text{false, otherwise} \end{cases}$$

For example, if both X =df (1,2,3,4,7) and X =* Y, we can deduce that Y =df (1,2,3,4,7). Formally, (=*,X,Y) and (=*,Y,X) are equivalent.

Concept: Ordered Subset (C*). X is an ordered subset of Y if it is contained within Y as a subsequence of Y. With reference to X as defined above, (1,2) C* X, (2,3,4) C* X, but not (2,4,7) C* X. Formally, (C*,X,Y) is written when X is an ordered subset of Y.

Concept: Concatenation(//).

$$X // Y \quad =df \quad (X,Y).$$

Concept: Correspondence ($\underline{r}\stackrel{*}{=}$).

Consider a relation r =df $((x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n),\ldots)$. (x_1,y_1) is a wff for i=1,2,...,n,... . Let $X=^*(x_1,x_2,\ldots,x_n)$ and $Y=^*(y_1,y_2,\ldots,y_n)$. We then define a correspondence between X and Y as $X \underline{r}\stackrel{*}{=} Y$. The element pairs x_i and y_i (for each i) are called correspondents. This means simply that there is some relation which associates each element of X with a corresponding element of Y.

Concept: Parameterization (P). A parameterization P is a set of two elements (p,l). p is a list of parameters $(p_1,p_2,\ldots,p_n)$. l is a list of components $(l_1,l_2,\ldots,l_m)$. Some of the l_i contain, as elements, the elements p_j. A parameterization is simply a procedure for indicating where replacements of the p_j can be made in l_i when the values of x_j are substituted for the parameters p_j. Formally, we denote the parameterization (or procedure) as (P,p,l).

Concept: Invocation (I). Whenever a particular set X is to be substituted for the parameters p of a parameterization P, we informally denote this as P(X). X is called the set of arguments of the invocation or execution of P. P(X) is the set which is identical to the set of components of P, except every occurrence of any parameter has been replaced by its correspondent from the set of arguments X, where $X \underline{r}\stackrel{*}{=} p$, $r =^* ((x_1,p_1),(x_2,p_2),\ldots,(x_n,p_n))$. In less formal language, the invocation of P with arguments X produces the set of components of the parameterization where the i^{th} parameter is consistently replaced by the i^{th} argument. Formally, we denote the invocation of P with arguments X as (I,X,P).

An example of a parameterization and invocation will be quite illustrative. Suppose we wish to parameterize a series of five numbers which always end with (3,4,5) so as to make the first two numbers replaceable. We could describe this parameterization as P =df ((x,y),(x,y,3,4,5)). An invocation of this parameterization with the set (1,2) as arguments could be written P(1,2) =df (I,(1,2),P) =* (1,2,3,4,5).

Concepts: Equivalent Parameterizations (E) and Reorderings. Consider two procedures (parameterizations) Q=(p,l) and Q'=(p',l'). If the parameterizations are identical in the structure of l and l' except for differences in the names and ordering of the elements of p and p', they are equivalent parameterizations. In this case, we write Q E Q' or Q' E Q. This may be operationalized as follows. If there exists some set q such that q=p' but not necessarily q=*p' such that the invocation Q(q)=*l', then QEQ'. Further, the set R=$((p_1,q_1),(p_2,q_2),\ldots,(p_n,q_n))$ defines a reordering relation such that $p \overset{R}{=}{}^* p'$. That is, (p_i,p_i') is a valid relation as a member of R. R describes how the parameters of equivalent procedures need to be reordered from one to the other. Formally, we may write (E,Q,Q',R) for equivalent parameterizations.

Concept: Extension (Ɇ). Consider a procedure P=(p,l). The set of all sets X for which P(X) is a wff is the extension of P. Informally, we write Ɇ(P) for the extension of P.

For example, suppose Q is a parameterization (using x) of the statement "x has blue eyes." Let Q =df (p,l) =* (x,(x has blue eyes)). The extension of Q, Ɇ(Q), includes all creatures and objects for which (x has blue eyes) is true. Formally, (Ɇ,P) is Ɇ(P).

Concept: Existential Quantification ($\exists$). $\exists(x)\ P(x)$ means "there exists at least one x such that P(x) is true." This is equivalent to the statement that the extension $\mathbb{E}(P)$ is not empty (the set $\emptyset$). Formally, we write $(\exists,x,P)$.

Concept: Universal Quantification ($\forall$). $\forall(x,P(x))\ Q(x)$ means "for all x such that P(x) is true, Q(x) is true." P and Q are parameterizations. Thus, universal quantification is equivalent to the statement that the extension $\mathbb{E}(P) \subset \mathbb{E}(Q)$. That is, all x for which P(x) is true belong to $\mathbb{E}(Q)$, the set of elements for which Q(x) is true. Formally, we write $(\forall,x,P,Q)$.

Notice that P does not possess any concept of the natural numbers or of arithmetic operations. For these concepts, P will be required to induce relations from the environment which effectively compute the relations desired. What P has been given in this section is a basic complement of logical operations. In the subsequent sections, we will augment this endowment with additional concepts that empower P to manipulate experiences and to perform induction.

3.3 The Conceptual Endowment (Idiological)

In the last section, a set of concepts was introduced which forms the basis of P's logico-mathematical system. In the current section, I will introduce concepts which P possesses about itself. I call these idiological concepts. The first of these definitions, that of concept itself, is without doubt the most important single mechanism for a thorough understanding of P's behavior.

Concept: Concept (K). A concept C is a parameterization

(X,T), where T =df $(t_1,t_2,\ldots,t_n)$. Each t_i is a rule for the transformation of some attributes of the environment into some others.

If we let each t_i be a set of two parameterized component lists (q_i,p_i), we can call p_i the preconditions and q_i the post-conditions of the concept. The meaning of a concept C is easily explained: The value of an invocation of a concept C(X) =df $q_i(X)$ if $p_i(x)$ is true.

Note that every concept is a parameterization and every parameterization a concept.

Let me demonstrate the meaning of concept with a special example. Suppose C is to be the concept which describes types of polygons. The concept can be partially described as follows.

Concept: Type of Polygon (T)

Parameter =df (x)

t_1: If x has 0, 1, or 2 sides, x is no polygon

t_2: If x has 3 sides, x is a triangle

t_3: If x has 4 sides, x is a quadrangle

t_4: If x has 5 sides, x is a pentagon

. . .

Suppose P possesses another concept called # and #(x) equals the number of sides of some object x. The concept T is then expressible as follows.

t_1: $\#(x) \in (0,1,2) \Rightarrow T(x)$ =df (no polygon)

t_2: $\#(x) = 3 \Rightarrow T(x)$ =df (triangle)

t_3: $\#(x) = 4 \Rightarrow T(x)$ =df (quadrangle)

t_4: $\#(x) = 5 \Rightarrow T(x)$ =df (pentagon)

It is possible to express this concept completely as a set of paramterized lists. Because it is just such lists upon which P's concept handling routines operate (theoretically), we will do this formally for the concept type-of-polygon as an example.

T =df $(x,(t_1,t_2,t_3))$

t_1 =df (p_1,q_1)

p_1 =df $(\in,(I,x,\#),(0,1,2))$

q_1 =df (no polygon)

t_2 =df (p_2,q_2)

p_2 =df $(=,(I,x,\#),3)$

q_2 =df (triangle)

t_3 =df (p_3,q_3)

p_3 =df $(=,(I,x,\#),4)$

q_3 =df (quadrangle)

. . .

A concept is, in sum, a named set C denoted formally (K,C), where C =df $(X,(t_1,t_2,\ldots))$, which prescribes how values are assigned to that name according to component transformational rules t_1. Examples of occurrences of concepts are members of the extensions of the corresponding relation. For instance, the extension of the relation type-of-polygon = triangle is seen to be just those elements of $E((x,p_2))$. Thus, any x for which $\#(x)=3$ will qualify as an example of triangle.

Some concepts produce only true or false values. An example of this sort of concept is the pattern recognizing concept dog. Occurrences of dogs are members of the set $E(\underline{dog})$.

It is important that the reader realize that all of the logico-mathematical concepts introduced in the last section are encoded in this formal precise way in P. It is not necessary for us to demonstrate these codings, but we must remember that all concepts are of a common format.

Concept Meta-Language (CML). It will be very valuable, both for the current and future projects, to employ a precise meta-language in the description of concepts. The parameterized notation is one which is feasible, but it is far too complicated and unnatural for the purpose of communication among humans. Instead, by following a short detour en route to my main objective, I will describe a more effective and precise notation for the representation of concepts.

The language which I will use is a natural vehicle for the expression of the two-part parameterizations embodied in conceptual transformations. I call the language CML for Concept Meta-Language. In the following pages, I will give a very brief description of the CML language and demonstrate its utility for our particular problems.

Name. A concept is a named set of transformation rules.

Parameters. Each concept is partially expressed in terms of a set of parameters $X =df (x_1,x_2,x_3,\ldots,x_n)$.

Arguments. Whenever a concept is invoked, a set of argument values is substituted for the set of parameters. If the arguments of the invocation are $Y =df (y_1,y_2,\ldots,y_n)$, y_i is substituted for any occurrence of x_i, for all $i = 1,2,\ldots,n$.

Preconditions. Preconditions are logical expressions which are either true or false depending in part on the value of the arguments.

Postconditions. Postconditions are the related logical expressions which are caused to become true (effected) when the preconditions are determined to be true.

Level. Every precondition is associated with a level number. The first level within the concept is level 1 and is represented by a (1) prefixed to the precondition.

When the concept is invoked, the level 1 preconditions are simultaneously evaluated. If none obtains (is found to be true), the concept cannot be evaluated and simply terminates its activity. When any one of the level 1 preconditions obtains, any associated postconditions is effected.

Subsequently, any level 2 preconditions nested within the obtaining level 1 precondition are simultaneously evaluated. If any one of these obtains, the associated level 2 postconditions are effected.This process continues indefintely until no further nested preconditions exist, at which point the evaluation process begins anew at level 1.

Termination. When a value is finally computed for the named concept being evaluated, that value is returned to the invoking concept and the valuation of this concept is terminated.

Notation. Preconditions are preceded by a parenthesized level number. (0) is prefixed to the concept name, which itself is followed by the parameter list.

The preconditions are succeeded by a colon (:). Any associated postconditions are written immediately following the colon. Several postconditions are separated by semicolons (;).

Variables. Any named set which is utilized in the conceptual procedure is a variable. Those variables which are not included in the parameter list represent either other knownconcept names or temporary sets which are used only for the purpose of computation within the immediate concept. A temporary variable is created for any named set in the conceptual procedure which is not the name of a known concept or a parameter. All temporary variables are initially equivalent to the null set (Ø) at the inception of the concept.

Assignment. When postconditions express an equality which is to be effected when preconditions obtain, this relation is expressed by the assignment notation ":=" as in the ALGOL language. If a concept C(x) is to be evaluated as equal to 1 whenever P(x) obtains, we write this postcondition as C := 1. ":=" means "is subsequently assigned the value of and becomes equal to."

Start. At the inception of the concept evaluation, an automatically constructed variable start =df true. As soon as any level (1) precondition obtains, start := Ø.

It is now possible to illustrate the simple CML language. Let us describe the familiar concept T (type-of-polygon).

```
(0) T(x):
     (1) #(x) ∈ (0,1,2)  :     T := (no polygon);
     (1) #(x) = 3        :     T := (triangle);
     (1) #(x) = 4        :     T := (quadrangle);
     (1) #(x) = 5        :     T := (pentagon);
```

As a more complicated example, I will show the CML program for next(x) which is a concept to produce the number (x+1) without utilizing rules of addition.[1] This concept employs the more rudimentary concept count. Count(x) produces the next number after x but is restricted to x = 0,1,2,....,9. Both count and next are concepts which should be acquired by P during its training. The training of P requires that count be learned by rote but next is to be learned by induction. How these concepts are acquired will be described later. For the moment, the descriptions of count and next will be provided to extend the reader's grasp of CML concept procedures.

```
(0) count(x):
     (1) x = 0        :        count := 1;
     (1) x = 1        :        count := 2;
     (1) x = 2        :        count := 3;
          .                         .
          .                         .
          .                         .
     (1) x = 9        :        count := (0,1);
```

1. My motivation for illustrating this concept is the consideration of developing a machine capable of solving many of the sequence problems discussed by Simon and others. Fundamental to their problem-solving programs which tackle numerical sequences are two concepts, next and periodicity. Periodicity simply examines a sequence and returns the number which equals the length of a repeating pattern in that sequence. Both of these concepts are attainable by P without specific programming. Therefore, P should be able to learn to solve sequence problems of this sort as well as sequence problems of a more general sort (which is exactly its primary task). A. Newell and H.A. Simon, Simulation of Human Processing of Information, Amer. Math. Monthly, February 1965.

Pointers. To describe the concept next(x), it will be necessary to introduce the concepts relating to pointers which P possesses. A pointer is a variable which locates a member of a set ("points" to it). Concepts exist for first, last, forward, and back which produce values of pointers which locate the first, last, next, and previous elements of the argument of these concepts, respectively.

When a variable, say P, is a pointer, it must be used in conjunction with a particular set to extract an element value. We connect the pointer P and any set X for which it locates members by an arrow, as P->X. If $X=(x_1,x_2,x_3)$, the postcondition P := first(X) will change the value of P so that $P\rightarrow X = x_1$. If, subsequently, we write P := forward(P->X), this will produce $P\rightarrow X = x_2$. The other pointer operations are effected in similar ways, mutatis mutandis. Whenever the bounds of a set are exceeded by forward or back operations, the pointer value nil is assigned.

Armed with these additional concepts, we now describe the concept next as a CML procedure.[1] Any precondition at level (i) which is complementary to another at that level may be written as a null precondition ("Otherwise :"). Such preconditions obtain whenever the other preconditions fail. Comments describing each of the conditions in the procedure are included between the symbols /* and */.

---

1. An important point should be made about the uniqueness of the concept to be described. There are clearly an infinite number of mechanisms which could achieve equivalent results to those produced by the concept next which is actually described. In my opinion, each of these constitutes a distinct concept. The equivalence of their outputs is a relation which allows all of them to be considered as members of the same extension, a fact which can be valuable for some purposes.

```
(0) next (x) :                  /* computes x+1 without addition  */
   (1) start :                  /* first obtaining precondition   */
       m := x;                  /* m is a variable where x+1      */
                                /*     can be computed            */
       p := first(m);           /* p is a pointer to first        */
                                /*     (rightmost) digit in m     */
   (1) otherwise:               /* evaluation continues here after*/
                                /*     start postconditions       */
       digit := count(p->m);
                                /* digit is a temporary variable  */
                                /*     which contains the next    */
                                /*     count after the original   */
                                /*     digit in the p-th position */
                                /*     (from right to left)       */
       (2) digit = (0,1) :      /* is there a carry in this place */
           p->m := 0;           /* replace digit by 0             */
           p := forward(p->m);
                                /* when there's a carry, advance  */
                                /*     to next digit              */
           (3) p = nil :        /* no more digits left?           */
               next := m // 1;
                                /* concatenate carry to the end   */
                                /*   of the number, making it one */
                                /*   digit longer than originally */
                                /* the next value is returned and */
                                /*   computing terminated here    */
       (2) otherwise :          /* there is no carry needed       */
           p->m := digit;       /* the advanced value is stored   */
           next := m;           /* the completed value is returned*/
                                /*   and the concept computation  */
                                /*   is automatically terminated  */
```

Relation between CML and parameterizations. Every CML procedure is directly translatable into a parameterized concept in P. These parameterizations are the lists and sets upon which P actually operates. CML descriptions are simply convenient human-oriented descriptions of conceptual operations. For our purposes, it is sufficient to say that each precondition and each postcondition is converted to a single parameterized component list. The entire set of these are then composed into a single concept. This process of forming one concept from many transformations is called systematization, about which we will have more to say in the next section.

Events and memory. P is equipped with an extensive memory for recording sets. In particular, P may record concepts and events. Events are the encodings of experiences of interactions with the trainer. Concepts are the componenets of explanations of experiences.

Suppose, temporarily, that P's memory is unlimited. Let every interaction with the trainer be recorded as a distinct five-tuple including the (1) input I, (2) prediction N, (3) response R, (4) date-time T, and (5) elapsed time ΔT. Thus, an event is a set $(I,N,R,T,\Delta T)$, which is added to the existing list on happening.

Concepts are stored in memory as sets of parameterized transformations as already discussed. When new concepts are generated by P, these are added to the list of concepts just as events are added to the list of events.

Every concept and event is assigned a unique identifying name which can be generated in any arbitrary way, for example, by using the time T at which it is recorded.

Concept: Attribute and Attribute Value. When we say r is an attribute of Y, we mean that thereis some set X such that r((X,Y)) is valid. Thus r must be some parameterization or computational procedure. Whenever r((X,Y)) is a valid relation, we say X is the attribute value of Y under the attribute r.

For example, let X =df (x_1,x_2) and Y =df (x_1,x_2,x_3). Since X C Y, X is an attribute value of Y under the subset operation C. This is true because (C,X,Y) is a wff. There are many other attributes of Y for which X is an attribute value, including C* since X C* Y and $\sim=$ since $\sim(X = Y)$ is a wff.

Concept: Explanation and Prediction. When P receives input I from the trainer, it immediately attempts to produce an explanation of I. An explanation of a sequence of symbols is any concept E for which the sequence I is a sufficient precondition for the evaluation of E(I). The predicted symbol set N is assumed to be E(I). That assumption is p-true.

These are important concepts. It should be noted that there may be many E_1 for which I is a sufficient precondition for the computation of $E_1(I)$. The extension H of this relation includes all feasible explanations of I. Suppose this extension H is partitioned into as many subsets H_1, H_2, H_3,... as there are unique predictions $E_1(I)$. Each class H_1 contains a set of feasible and consistent explanations of I. When the trainer's response R is made available to P, some of the H_1 will be invalidated and others will not.

Let (H_o, H') be a partition of the H_1 such that all E_1 in H_o are explanations which lead to predictions not invalidated by R and all E_1 in H' are explanations directly invalidated by R. Any parameterization $\overset{r}{\wedge}$ such that $r(h_o)$ is true and $r(h')$ is false, for any $h_o \in H_o$ and $h' \in H'$, may be considered an explanation for the validity of the explanations. This notion, that there are properties of concepts which explain the validity of predictions, can be of great utility to our machine. It is but one example of the practical concepts of ideogenesis, to which we now turn.

3.4 The Conceptual Endowment (Ideogenetical)

It is now possible to describe procedures for induction and

ideogenesis. First, it will be helpful to describe the nature of inductions from our newly constructed viewpoint. Once we have described induction somewhat informally, we will set forth several concepts which will achieve programmed induction for P.

In section 2.5, induction was first described purely and simply. In that definition, we said an organism O had induced a property Q when the occurrence of Q became a sufficient condition for the prediction of a definite behavior by O. That is still true and remains far too abstract. We are now interested in refining this concept sufficiently to make such change-oriented behavior controllable by P.

For my current purposes, it will suffice to introduce four methods of induction. Of course, these classes are arbitrarily defined and overlapping, but they reflect significant differences. Let us call these four types of induction types one through four or, alternatively, the methods of (1) pure abstraction, (2) systematization, (3) extension, and (4) composition. Each of these will now be discussed in turn.

Figure 3.1 here

Type 1: Pure Abstraction. Consider a set of events $P =df (E_1,E_2,\ldots,E_n)$ where each $E_i =df (I_i,N_i,R_i,T_i,\Delta T_i)$. Suppose there exists a subset Q of these events in which all $R_i =^* R_Q$ and no $E_j \in P$ which is not in Q has $R_j =^* R_Q$. That is, there may be a subset Q all of which lead to the same response R_Q by the trainer.

If all of the events in Q share some common attribute values of their inputs, for example if all I_1 =* (0,1), it is feasible that this input I =* (0,1) is sufficient for the prediction N =* R_Q.

In other words, if it is found that those events in a particular extension of a relation on I, say r(I), are also members of the extension of events for which R =* R_Q, the induction that r(I) => R =* R_Q is feasible (p-true). Such an induction is called the pure abstraction type and is the basis for classical category rule concepts.

Type 2: Systematization. Suppose we have a set of events P =df (E_1,E_2,...,E_n) as above. Further, suppose we have induced by pure abstraction the several concepts h_1, h_2, h_3 such that h_1: $r_1(I) \Rightarrow R =^* R_{Q_1}$, h_2: $r_2(I) \Rightarrow R =^* R_{Q_2}$, and h_3: $r_3(I) \Rightarrow R =^* R_{Q_3}$. Suppose, finally, that the r_i are logically exclusive in the sense that only one of r_1, r_2, or r_3 could simultaneously obtain for any given argument I. That is, the extensions of the r_i partition P. In such a case, the induction of a systematized concept H including the individual transformations as components is feasible. The formation of one concept from may diverse transformations is called systematization. Systematization is classically considered the formation of a taxonomic hierarchy.

As an example, consider the three concepts h_1: I=*(0,0) $\Rightarrow$ R=1, h_2: I=*(0,1) $\Rightarrow$ R=2, h_3: I=(0,2) $\Rightarrow$ R=3. These are, in fact, quite similar to the first three component transformations required by the concept count discussed in section 3.3. If each h_i is written as a parameterization (I,q_1), the induction of the concept H =df (I, (q_1,q_2,q_3)) is a feasible assumption.

Type 3: Extension. Consider a parameterization P of one parameter I and its extension $\mathbb{E}(P)$. Suppose there are some events in memory, $E_1, E_2, \ldots, E_n$, for which $P(I_i)$ is valid for $I_i \in E_i$. From these events I_i and R_i are taken to form the set $K =_{df} ((I_1,R_1),(I_2,R_2),\ldots,(I_n,R_n))$. If there exists some parameterization $r(x,y)$ such that $r(I_i,R_i)$ is a wff for all (I_i,R_i) in K, the assumption that this relation r will hold for all events satisfying P is called an induction by extension. Specifically, $\mathbb{E}(P) \subset \mathbb{E}(r)$.

Type 4: Composition. Consider a parameterization P of one parameter I and its extension $\mathbb{E}(P)$. Suppose, as above, events E_i which satisfy P and are elements of $\mathbb{E}(P)$. Suppose the existence of a set of concepts $(C_1,\ldots,C_m)$ with m greater than 1 such that $R_i =^* C_m(C_{m-1}(\ldots(C_1(I_i))\ldots))$, for all I_i and R_i in these E_i. The assumption that this relation will hold for all events satisfying P is called an induction by composition. The concept H which is induced, such that $R_i =^* H(I_i)$, is called the composition of concepts $C_1,\ldots,C_m$. It may be written $H =_{df} C_m{*}C_{m-1}{*}\cdots{*}C_1$.

It should be noted that inductions of type 4 subsume the other three types. As an example of composition, we may suppose that P originally possesses two concepts A and B where $A(x) =^* (x,x)$ and where $B(x) =^* (x,x,x)$. If the I_i-R_i pairs in events in $\mathbb{E}(P)$ include $(a,((a,a),(a,a),(a,a)))$ and $(1,((1,1),(1,1),(1,1)))$, the induction that $R_i =^* H(I_i)$ where $H =_{df} B{*}A$ is feasible.

Procedures for Induction. It is, of course, necessary for an

intelligent machine that all of the induction methods just described be computable and programmable. P must therefore possess apposite concepts for each of these induction mechanisms. The descriptions above, though operational, are insufficiently detailed to utilize as procedural descriptions.

Although it would be possible to describe induction concepts as CML procedures, I will not do this here. Instead, I will briefly discuss how such procedures might be programmed. In the following section, we willaddress questions of efficiency and motivation without which the actual programming of a desirable machine would not be possible.

The basic structure of the concepts of induction is simple. First, we have provided definitions of feasible inductions of all four types. A procedure for computing inductions to explain a particular input would necessarily search memory for past events and existing concepts satisfying the induction preconditions. If an event is found during this search which exactly matches a new input I, the explanation used in making a prediction can be "this is like a previous event and the prediction should match the trainer's previous response." When no such event is found, P must instead compute feasible inductions as explanations of I.

For type 1 inductions, as an example, a concept C must be found which is both computable for I and for whichthe extension G of all events x such that C(x)=*C(I) is non-empty. Further, all events in G must have R=*R_G for some constant R_G. At this point, a new concept r is created and added to memory. r is

defined by the CML procedure:

(0) r (I):

(1) C(I) =* C(x) : r := R_G ;

where x is any event in G. In processing subsequent experiences, this new concept r is available as a ready p-true explanation for all events satisfying its precondition.

It should be reiterated that there are possibly an infinite number of feasible inductions of this sort. There are many strategies, each with different strengths, for coping with this problem. One alternative is to randomize the search for feasible explanations and place a ceiling on computation time during search. Another is to develop hypotheses according to the most successful conceptual components of past events. Both are possible strategies and each is desirable under particular circumstances. In the next section, the second strategy is advanced in more detail, not because it is preferable, but because it is the more provocative for our theoretical development.

3.5 Drives for Efficiency and Predictability

P is a purposeful machine. It strives to achieve the maximum degree of predictability about the trainer's responses. Further, it attempts whenever possible to minimize the length of time which it spends computing predictions. These two goals or purposes constitute P's drives for predictability and efficiency.

The mechanisms whereby predictions can be formulated have been explained in the last section. The additional mechanisms required to assure that the predictions made by P continue to

improve probabilistically can now be introduced. I will propose here one simple but arbitrary mechanism to achieve increasing likelihoods of accurate predictions. This particular heuristic may or may not be the best, but it seems intuitively clear that it will be relatively efficient for training sequences which do no contain a great number of "surprises."

At the outset let me remind the reader again that the relative value of one mechanism versus an alternative is calculable only when no uncertainty exists about the environment in which it operates. One must be careful to realize that even the supposition of probabilistic distributions and the introduction of random variables into a mechanism does not meet this complaint. It only shifts the question of uncertainty to the issue of why use one assumed distribution and not another. There is no escape from this problem. Simply stated, an induction machine designed with any drive for efficiency, however it is stated, will be at a disadvantage in some environment with respect to a machine designed with some alternative drive.

Theorem 3.5.1. Consider two intelligent machines P and Q which differ only in the mechanisms of their drives for efficiency. Suppose, for example, P prefers to reduce computing time and Q prefers to reduce computing storage required for their predictions. Unless these two purposes are operationally identical for all possible measures--which would violate the assumption of their being different--there must exist two separate training sequences T and U with a common last problem I such that the relative efficiency of P and Q on I depends on which training sequence was followed. That

is, P achieves greater efficiency on the last trial of T than on the last trial of U while Q achieves greater efficiency on the last trial of U than on the last trial of T.

To implement P's drives, I propose these mechanisms. Let P maintain for each concept in its memory an overall count of the number of successful applications of that concept minus an increasingly weighted count of failures resulting from that concept. A successful application is defined as any use of a concept in an induction or explanation which leads to a correct prediction. A failure is defined correspondingly. Call this difference the utility u of a concept.

Whenever P must choose among several possible concepts in the computation of an explanation (for a sequential processor, such choices are made at every branch of a decision network) let it order the alternatives in accordance with the value of each concept's utility. The higher the utility, the more preferred is an application of that concept.

When several concepts are composed to generate a new concept, let P most prefer the induction i which has the greatest value for the term

$$v_i =_{df} \frac{\sum_{j=1}^{n_i} u_{ij}}{n_i^2}$$

where u_{ij} is the utility of the j^{th} component concept of the i^{th} alternative induction comprised of n_i total components. This particular heuristic would favor "shorter" explanations than longer ones and thereby accords well with the "law of parsimony."

For this reason, v_i is called the parsimony value of the induction i.[1]

Whenever a concept is used successfully in the development of a prediction its utility increases by a fixed amount, say one. Whenever a concept is used in the development of a prediction which is rejected by the trainer, its utility is decreased by the total number of failures-to-date. Thus, failures continue to be weighed at a linearly increasing rate of significance while the significance of another success remains constant.

Whenever a concept is induced and added to the memory of P, I would have it assigned an original utility equal to the parsimony value v_i of its explanation. And whenever choices are to be made among diverse concepts for the purposes of prefering one explanation to another, P should accept only those for which v_i is no less than v_T, the parsimony threshold value.

When several explanations arise in P for a single problem I which are all in excess of the parsimony threshold value, P should prefer these in accordance with its drive for efficiency. This drive is mechanized by having P make its prediction N from that explanation with the smallest elapsed time ΔT. Thus, P favors the more efficient concept of two which are both parsimonious with

---

1. Notice that the "law of parsimony" is really not a law at all. First, it cannot be operationalized except within a particular knowledge K. Once that knowledge is fully described, measures of parsimony are arbitrary orderings of some attributes of K. I have suggested one measure which favors explanations requiring fewer components to those with more, ceteris paribus. I do not suggest that this is an ideal measure. By Theorem 3.5.1 it is obvious that such a measure can be ideal for an uncertain environment only accidentally.

respect to a specific prediction problem.

Any rule designed in consonance with this objective would be as a priori equally acceptable as any other. It seems more desirable, in fact, that P actually prefer concepts which have the greatest time-utility value w_i, computed as follows. Suppose P must choose among m alternative explanations $H_1,\ldots,H_m$. Each H_i is presumed to have parsimony values v_i which are no less than the parsimony threshold value v_T. Each H_i requires t_i seconds of computing time. Let w_i =df v_i/t_i. That H_j for which $w_j = \max(w_i, i=1,2,\ldots, m)$ is P's preferred explanation.

In short, P may have to choose between alternative behaviors in many points in its execution. I propose that P be guided by drives which accomplish its purposes. In choosing one particular concept over another in the competitive search for a feasible induction, it can be guided by the utility u. The feasibility of explanations is determined relative to the parsimony threshold value v_T. In choosing among equally feasible parsimonious explanations, P prefers explanations inversely with respect to their computing time. The most preferred feasible explanation is used to produce P's prediction N. When the trainer responds with the correct answer, the utility of concepts leading to successful predictions is strengthened while that of concepts utilized in making erroneous predictions is diminished.

Thus, P will exhibit many types of learning. Correct predictions by P will be self-reinforcing as will the mere development of feasible alternatives. P can be said to be motivated for com-

plete predictability and for efficiency of time. Whether or not P achieves any particular goal depends on the environment (the training sequence), P's computational and memory capabilities, and the initial values assigned to utilities of endowed concepts. In the next section, the relation of these properties to P's development is pursued.

3.6 Training for Intelligence

The most significant fact reflected by the design of P is that, although endowed with a sufficient set of concepts to induce all possible relations, P requires a well planned training sequence to learn important ideas and to become efficient. P possesses at its inception multifarious computational capabilities but does not know, a priori, which concepts to employ to solve which problems. Rather, P will necessarily begin learning by trial-and-error --trying one method of induction here, another there--until it has induced sufficiently complex and precise concepts to reduce the most complicated problems to simple transformations. The original discovery of a complicated compositional induction which leads to a successful problem solution will appear "insightful" to a naive observer of P. Subsequently, however, P will begin to stereotype that problem-solving behavior so that what seemed originally novel and insightful becomes the commonplace and predicted.

Our problem then is to provide a training sequence for P which facilitates its continued learning in a particular environment. For example, if we are interested in teach P a certain systematized concept, it will be advantageous to teach each component transformation

first. Subsequently, P could learn the systematization by straightforward induction of type 2. In a similar way, inductions of compositions of concepts can best be trained by the piecewise development of partial compositions developed one-at-a-time until the total composition is achieved. The correspondence between this procedure and the method of "chaining" in operant conditioning is obvious.

I propose that we begin P's training by teaching external symbolic names for each of the simple concepts which P possesses internally in its endowment. That is, we should teach P the logical and mathematical symbols that we use informally to describe P's own behavior. We could then utilize these symbols to instruct P in the performance of more difficult tasks or to actually teach such concepts as next(x) = x+1. By developing an instructional language, we can acquire the power to teach P anything which we can operationalize. That is, it seems to me, precisely how we teach most adults most concepts.

In the subsequent section I will discuss the significance of teaching P verbal concepts for its internal operations. The significant fact to bear in mind is that inductions of explanatory concepts from a set of natural events are tantamount to the adoption of rules for behavior supplied by verbal or symbolic concepts. In some sense, P is the slave of its trainer. P must induce. It must induce behaviors that lead to reward whether or not the events in the training sequence are symbolically meaningful to the trainer. At some point, events which the trainer supplies P--for example, measurements taken of an exceedingly complex process--may in fact be

incomprehensible to the trainer. P is totally insensitive to this event and will continue inducing relations until it has achieved predictability or has been turned off. Perhaps, at that point, when P has reached an understanding of the problem, the trainer will examine P's concepts and acquire new knowledge about this previously incomprehensible activity.

To begin, then, we choose some very simple concept to be acquired, supply P with examples of its use, and each time ask P to predict the value of the next term in the sequence (the conclusion). We continue this process until P has demonstrated that it has learned the desired concept.

As the first, I suggest we teach the concept of value, where value(x)=x when x is simply a symbol or value(x)=true when x is a wff. Without loss of generality, I will presume that P automatically constructs from the input I =df "value (x)" the set (value,x). The method it uses can be presumed to be a concept in its endowment; that is not important.

Suppose the trainer supplies the input (value,1). P recognizes two attributes of 1 and none of "value". It recognizes $1 \in D$ (the set of digits) and 1 = 1. Suppose P prefers--because of initial utility values--the relation 1 = 1 as an explanation. If P predicts 1, the trainer responds that 1 is the correct answer. But what changes, ideogenetically speaking, have occurred?

One possible induction that P could have made is: when the string "value" occurs and "1" follows it, "1" is the correct prediction. This is, of course, equivalent to the recording of

the event where I=*(value,1), R=*(1). Another concept that could have been used is the concept of "=" that says x=x, or in this case, 1=1. Whichever of these or others occurred is reinforced by the successful prediction.

Over time, however, a set of inputs is supplied which vary the arguments of the function and complicate the possible consistent explanations. At some point, P may actually have achieved a concept which "works" fairly well. Suppose P has actually constructed a concept which says: "if the letters u and e are members of the first symbol string, the value is equal to the second symbol string." It is not until the trainer begins providing counter examples like (valeu,a)=false and (eulav,a)=false and (balue,c)=false that P discovers one induction that fully explains the experiences the trainer has provided. At this point, P will have discriminated those events which it predicted accurately (those spelled correctly) from those it made errors on(those not spelled "value").[1]

In all of the correct predictions based on the "=" concept, the relation (value) C* I was found. At some point, P induces the concept <u>value</u> by type 3 induction:

---

1. The skeptical reader should be advised that this behavior is genuinely programmable. The concepts in P's endowment are sufficient to guarantee the discovery of this induction in finite time if appropriate initial utilities are assigned. For a detailed discussion of a more specialized sub-intelligent induction program, I recommend Winston's thesis, <u>Learning Structural Descriptions from Examples</u>, Cambridge, Massachusetts: MIT Project MAC, TR-76, September 1970. In that landmark project, Winston used specific model building heuristics to develop structural models of graphic designs. The extension to sets of symbols and generalized models as advocated in this paper seems, to me, quite natural.

```
(0) value (x) :
      (1) (value,x) =* I     :  value :=  x;
```

The next concept taught to P is the concept of truth of wffs expressed in the trainer's language. The trainer supplies examples of wffs and responds "true" or supplies non-wffs and responds "false." Of course, this concept will continue to acquire new component transformations as long as new axioms are introduced until P acquires the more complex notion that "any axiom or axiomatic deduction is true." Nevertheless, a partial concept can be constructed given only one well defined operation, that of value. To accomplish this P is taught the symbol "=". Whenever the terms on either side of the = are in fact equivalent, the trainer responds "true". Otherwise, he responds "false".

The following are exemplary input-response interactions.

Trial	Input	Response
1	true	true
2	false	false
3	x=x	true
4	x=y	false
5	y=y	true
6	1=1	true
7	value(x)=x	true
8	value(3)=3	true
9	value(value(x))=x	true
10	value(value(x))=y	false

When P attains the correct concept, it will be:

```
(0) truth (I):
     (1) (true) =* I       :       truth :=(true);
     (1) (false)=* I       :       truth :=(false);
     (1) (x,=,y)=* I :
                    (2) value(x) = value (y) : truth := (true);
                    (2) otherwise            : truth := (false);
```

Once this concept is learned, P is taught that the concept of

value is extendible to cover true and false values also. The following training sequence would be appropriate:

Trial	Input	Response
1	value(true)	true
2	value(false)	false
3	value(x=x)	true
4	value(value(x)=x)	true

It is now desirable to discuss how concepts which are once induced can be modified subsequently in the face of counterexamples. In P, this behavior is easily explained. Consider P's concept of value, for example. Suppose value is not completely correct and for some non-wffs P's concept value response is true and for some valid wffs its response is false. The component concepts which P originally utilized in development of the induced concept value are still available for further inductions of alternatives to the erroneous value concept. As value continues to err, its utility decreases at an increasing rate. At some point, the time-utility value of an alternative explanation will surpass that of the old concept value. At that point, the new concept will usurp the old one. That is an explanation of how old behaviors vanish. They simply become less desirable to the machine than alternative ones. The trainer actually elicits an incompatible response from P which effectively surpresses its erring behavior.

In like manner, I propose that P be taught the remainder of its internal concepts. This includes all of the logico-mathematical concepts in section 3.2 and additional concepts of pointer operations. At that point, P is equipped to learn more complicated instructional sequences and can be used as an effective problem-solver.

3.7 Teaching with Language

It almost seems that our original problem has vanished. In the beginning, our goal was to construct a machine that was intelligent. I suppose the motivation was to study a machine which sensed the environment, grappled with unknowns, and reached a meaningful interpretation of events. To do this, we endowed P with many concepts and proceeded to teach it verbal tokens which signified each of the endowed concepts. At that point, we had acquired a language which we could use in teaching P new concepts. P could then either acquire new concepts directly via instructions from the trainer or by continuing to abstract relations from some other (non-verbal)environmental events. When doing the latter, P discovers that previously unexplainable events are reducible to a subset of an extension of a particular composition of preexisting concepts.

This process can shed much light on the nature of learning in general and on the mechanisms of language in particular. Concepts, originally described as the rules by which diverse events could be classified as a single perception in an organism's behavior, have now been fully explicated. The meaning of a verbal concept is quite clear, as a result. Every time we teach P a token for some internal set of operations--for example, truth(x), which is simply the label of the operation which determines the well formedness of P's internal operations--we have given P a verbal concept. Quite precisely, a verbal concept is an element of a language whose wffs can be put in isomorphism with the wffs of

the internal operations of some computing machine. A token signifies a conceptual parameterization in P which is invoked to predict events in the environment from which that token was learned.

Thus, a language is a set of tokens and the procedures for prediction which sets of tokens are wffs. The acquisition of language in P through induction seems quite feasible. P must learn for each token which ordered sets of tokens are valid and which are not. Learning that a set of tokens is valid means possessing procedures that transform the set of tokens into a valid prediction of the environment. P learns this precisely by discovering a system of concept compositions which produce internal wffs in parallel to wffs in the environmental language.

The language of P continues to expand indefinitely as the new events and predictions are introduced. As the language of P grows, so does its capacity to recognize distinct explainable events. Every externally supplied input event I must be comprehended in P as the composition of concepts which take I into a correct (true) or an assumed (p-true) prediction. Whether the input is verbal (composed of tokens) or non-verbal is irrelevant to the operation of P. If P can predict the next symbol from a set of symbols, P must have a concept (by definition) for that transformational behavior.

Once P has learned most of its basic verbal concepts, teaching P the concept next(x)=x+1 is quite easy. Simply, P is taught separate components of the CML procedure for next, one-at-a-time.

Systematically, these are composed until P acquires the full concept next which transcends each of the individual components. At that point, P will have learned to count.

Learning to count veritably opens Pandora's box to P. Once numbers are available so are many other ideas. If the numbers are associated with the operations of listing each item of a set --that is, setting a pointer to the first element, then the next, then the next, and so on-- P will be able to induce the real meaning of number as a measure of quantity. P can then be taught arithmetic as operations on numbers which represent quantities. For example, addition can be learned as a counting process, like counting on one's fingers. Such a procedure might be very much less efficient for computational purposes than an exact procedure for addition which is based on adding digits one-at-a-time and using a carry for overflow. However, it does not follow from such a premise that learning addition as a counting process is less desirable than learning it as a routinized procedure utilizing column by column addition. To the contrary, the question of best method is empirical. Only by knowing what the machine must do during its entire existence and knowing exactly what training it will subsequently receive can we determine what is the best way to teach it a particular operation. It is clear, nonetheless, that alternative mechanisms which take identical inputs into identical outputs are different concepts. The value of different concepts in the development of subsequent concepts based, in part, on the contents and structure of the earlier ones is not determinable in uncertain environments.

This is especially true when the organism is taught one or the other and not both of the alternatives.

At this point, I concur in part with the developmental psychologist Jean Piaget. There may be many things that organisms learn to do at one particular age which they could alternatively learn to do at some point earlier in their development. We do not infer that they ought therefore to learn everything as early as they are physically capable. In fact, it may be taken as a corollary to Theorem 3.5.1 that for some task W which it is taken as P's purpose to accomplish at a point later in its life, two different training sequences T and U could be imagined such that P would actually be unable to accomplish W if preceded by U but not if preceded by T.

Consider again the question of the better way to teach P addition. The first method is to teach it to count, as on its fingers, from the first number by ones until it has counted as many ones as the numerical value of the second number. The alternative method involves teaching the use of a table of sums for two one-digit numbers plus a carry digit that is either 0 or 1. Addition in this method is performed by invocation of the table while working from the first (rightmost) digit to the last, a column at a time. The question, "Which of these methods is better?" is clearly undecidable without much greater context. If counting on one's fingers enables one to learn rhythm as an analogue of the physical process, is it not good for that reason? It may not teach one the value of systematic composition of concepts as

learning to add in a formal way might. In some unpredictable way, either of these concepts might be extraordinarily valuable for particular purposes or extraordinarily inhibiting for others.

In short, our machine P is capable of learning what we wish to teach it. We teach by designing training sequences of verbal or non-verbal events which effectuate appropriate inductions by P. The burden of determining optimal of desirable training sequences rests squarely with the trainer. Our machine is ready to learn whatever we will.

4.0 DOES THE INDUCTION MACHINE THINK?

It is now clear that emphasis on the question, "Can a machine think?" is inappropriate. Any organism that converts problem environments into predictions of successful responses thinks. The more interesting question, "How does this machine think?" will be considered in this and the following sections. In this section, we will discuss problems which P can solve and those which it cannot. In doing this we will be concerned with characterizing the class of solvable problems. In the next section we will pursue a more general functional relationship between characteristics of problems and characteristics of thinking machines.

4.1 Problems which P can Solve

Since P reads sequences of symbols, compares them to previously encountered sequences, and computes many conceptual transformations, its performance is necessarily dependent on several exogenously determined parameters. Basically, the consideration of what capacities to provide P such as volume of memory, computation speed, and capability for parallel-processing can greatly affect the problems whcih P can effectively solve.

P should be able to learn verbal tokens for all of its endowed concepts. Thereafter, any concept which can be expressed as a CML procedure should be attainable by P. However, not all of these procedures will necessarily be acquired. Of those which P actually develops, some will be internalizations of verbal descriptions and others will be the conceptual induction of abstracted relations.

Which concepts P is taught should depend on which tasks it is to be assigned later in its training sequence. The initially endowed concepts will enable P to accomplish many ideogeneses of simple natures. For example, P can learn the equivalence of different sets for a common purpose, common orderings of attributes of events, the significance of some aspects of a problem with irrelevant or noisy attributes, and so forth.

P can be taught value systems as concepts of preferential orderings. Systems of orderings can be developed as correspondences between possible preconditions and desired postconditions. P can be taught to invoke values in some circumstances and not in others.

In particular P can develop concepts of "good" and "bad" methods of solving problems. In this way, P can be trained to use the scientific inductive method while another machine Q is trained to pursue an analytical, unempirical, method. P can be taught to compute utility distributions for the trainer's preferences and to employ these in selection of optimal Bayesian decision strategies.

In short, P can be taught many different behaviors. All of these will be tantamount to transformations on sets of symbols. The capacity of P to solve novel problems is thus dependent on P's capacity to recognize salient attributes of such sets as preconditions for problem transformations. The saliency of an attribute is to be determined only in full view of its context. P must be trained systematically to recognize those sets of attributes

which in composition constitute a single salient aspect of a problem.

4.2 Ambiguity and Error

Every concept which P possesses is absolutely concrete. That is, a conceptual precondition is evident or it is not. Abstraction occurs not from real sensations to imagined sensations. Rather, abstraction is the process of finding just those concrete sensations on which to base an action. For example, the concept dog is not abstract. It is just that condition of stimuli--eyes, feet, ears, whiskers, body--which define dog. Number is neither more nor less abstract than dog. It is just that condition of stimuli--countable--which define numbers.

There is one sense in which concepts are not absolutely concrete. That is, concepts are representations of events which are approximate and inexact. Concepts learned by induction are always only p-true. This is true because there are always numerous feasible inductions not all of which lead to the same predictions or support the same inferences. Because of this intrinsic flexibility of conceptual explanations, concepts include domains of error over which their conclusions are actually false of p-false.

The approximate validity of learned concepts, owing to the p-true status of the induction, is both a source of value and cost for any machine. It is precisely because concepts can be used sloppily (with ambiguity of meaning) that people can accomplish so much with so few words. The machine P is also capable of such ambiguity in its own thinking. The induction of a concept which

is only partially complete can provide the stimulus for many subsequently erroneous actions. The accidental but correct prediction of some event will reinforce P's mistaken concepts. Since mistaken concepts of this sort are never deleted, they can reassert themselves as means of explanation throughout P's life. In some cases, understanding of environmental events depends exactly on such erroneous concepts. In other cases, erroneous explanations will be punished and inhibited by the trainer.

The ability of P to exhibit all of the salient stages of concept formation[1] is clear. P can exhibit mistaken associations in the induction of concepts, either due to overspecification of general properties or to over generalization of specific concepts. Furthermore, P may choose inappropriate bases for concepts which accidentally correlate with the valid bases. In such a case, P will exhibit correct conceptual behavior most of the time even though employing incorrect transformations. Correcting this particular sort of conceptual error could be quite difficult for the trainer and, perhaps, impossible.

Thus P is a machine which is capable of handling noisy or inexact inputs in an intelligent way. In fact, P can develop specific concepts of noise and thereby recognize certain events as

---

1. For an excellent introduction to studies of concept formation, I recommend L.S. Vygotsky, Thought and Language, Cambridge, Massachusetts: The MIT Press, 1962. See also J.S. Bruner, J.J. Goodnow, and G.A. Austin, A Study of Thinking, New York: Wiley, 1956.

noisy occurrences of more regular lawful relationships.

4.3 Behaviors which are Unattainable by P

There is much that P simply cannot do in its current design. It cannot run, hear, or see: in that sense it is really quite limited. Furthermore, P is generally unlikely to induce relations about events which are not of direct relevance to its primary purpose, achieving predictability.

In training P, we must remember that it desires to predict whatever we think is correct behavior. If we wish it to predict what we see, we may have to introduce new forms of input and new concepts in its endowment. Specifically, we should introduce two or three dimensional networks and relevant primary concepts about these networks. To have it predict sequences of visual images, we would need to train P on sequences of the three dimensional sets. I am not suggesting that this is a good method for achieving a seeing machine. Rather, I am interested in demonstrating that P's knowledge is bounded by the primary attributes which it is able to perceive.

Specifically, P is capable only of inducing relations which represent compositions of the operations which it already possesses. If a stimulus environment is encoded in terms of elementary attributes, such as lines and contrasts, these attributes must be perceptible by P if P is to understand the environment. The remarkable specialization of human sensory apparatus lends support to this observation.

4.4 Hierarchical Knowledge

It is now possible to understand the nature of hierarchical

knowledge. Basically, P is endowed with a set of computational procedures and concepts which determine how ideogenesis is to occur. Every new concept is built hierarchically upon some old ones and attains integrity as an alternative mechanism for comprehending the environment. To the extent that concepts are utilized in the formation of an induction are they attribute values of that new concept. This attribute is loosely called "hierarchy" of concepts.

As I have shown elsewhere[1], concepts can be related hierarchically in many ways. The development of concepts which recognize such hierarchies is a straightforward example of inductions of type 2. For example, the discovery that for some purposes dog, cat, guinea pig, and parakeet are equivalent is the basis for the induction of a category, in this case of house pet. Subsequently, many different systematizations utilizing these same concepts can be effected by training.

Alternatively, it is possible to connect every concept with every other by some chain of conceptual associations. These associations are simply relations among concepts rather than among purely external events. Nothing is surprising about this. When P uses concepts to describe events it is using procedures which are themselves analyzable sets. The discovery of novel properties of those parameterizations which represent concepts which, in turn, facilitates improved predictions by P is possible. The concept noun, for instance

---

1. F. Hayes-Roth, The Structure of Concepts, Cambridge, Massachusetts: Sloan School of Management Working Paper, April 1971.

can be taught to P as soon as examples and counter-examples of noun exist for P to study.

4.5 Analogical Reasoning

Many people cite the ability of people to reason by analogy as significant and as distinguishing human intelligence from other varieties. It is now possible to reconsider the process of analogical reasoning and to relate it to the reasoning of P.

When we say "A is to B as C is to D" we may mean several different things. In general, these diverse meanings can be subsumed under the statement, "If C and D are in relation r, such that r(C,D) is true, then r(A,B) is true, and vice versa." That is, both sets (A,B) and (C,D) are members of the extension of r(x,y). For example, apple is to fruit as cabbage is to vegetable, because apple is a kind of fruit and cabbage is a kind of vegetable. It is quite easy to see that an analogy A:B as C:D is valid if there exists some concept r(x,y) such that r(A,B) ≡ r(C,D).

Analogical reasoning is similar to induction by composition. In induction by composition, two events (A,B) and (C,D) are discriminated and a relation r is composed such that r(A,B) and r(C,D) are both true. The purpose of the induction is to fabricate an explanation for the equivalence of the two sets. That is precisely the purpose of an argument by analogy, to argue that the similarity of different sets of events is so significant as to outweigh the importance of their obvious differences.

We have already shown how P can be programmed to induce that I ⇒ R for an I which is like the I's in other events. For P to

produce the prediction R and the explanation R is to I as R_E is to I_E for all events E is analogical reasoning. Training P to analogize is unnecessary. P is a born generalizer.

4.6 Systematic Behavior Traits

We have come to the end of the descriptive road. P has been introduced and its capabilities extensively discussed. All that remains is to give perspective to P as an organism: In what way does P resemble other natural organisms? A propos to our discussions in section 2, it is desirable to focus on the questions of perception, prediction, and learning.

P perceives the environment through its reader R. The input to P is produced in accordance with natural laws, as far as P is concerned. These laws are the rules that govern the trainer's behavior. P is capable originally of perceiving a variety of symbolic patterns in its input. Those patterns which are immediately recognizable as satisfying the preconditions for conceptualization might be called primary perceptions. These include the set I, the prediction N, the trainer's response R, the timers T and ΔT, and the set of symbols. Each of these primary perceptions is, in turn, perceptible by P as satisfying preconditions for secondary concepts. For example, the elements of I can be recognized as subsets of I or subsets of I can be considered equal to previously encountered sets. In general, many possible sets of composed concepts may be satisfied by the occurrence of an input.

Those input events which result in equivalent outcomes in P constitute as a set one perception. It may now be seen that it is

possible to distinguish many different levels of perception according to the particular measurement of "equivalent" we use. At the lowest level, we may talk of sensation defined as being the state of P immediately upon reading I. Or we may describe perception, at the highest level, as that conceptual composition and the requisite stimulus preconditions for P to produce its prediction. In between, it is possible to assess a myriad of measurements of the perceptual (read cognitive) processes.

P may be defined as equivalent to some knowledge K which is adequate to predict completely P's actions. That may sound amusing since P itself is a machine for predicting the trainer's responses. Nevertheless, it is always useful to remember that knowledge is not immanent in the organism but is in some sense independent of it. The machine P is considered in terms of a knowledge K just as the trainer's behavior is expressed in terms of a knowledge L.

Neither the scope nor the range of the two knowledges, K and L, need be totally intersecting. The reason for this lies in the ability of P to abstract extensive relations which incorrectly exceed the domain of validity of such a concept. Thus, each concept induced carries the potential for introducing error. Predictions made by K on the basis of such concepts need have only the most superficial dependence on the true explanatory concepts as seen in L.

As a learning machine P is quite compelling. All learning is adaptation, and P is therefore an adaptive machine. But P is also

teleological, seeking to replicate those behaviors which lead to rewards and to avoid others. Other machines have been constructed which exhibit definite tendencies to stereotype rewarded behaviors.[1] All such previous learning machines with which I am familiar, however, were not equipped with ideogenetical capacities. They could definitely perceive and learn to selectively discriminate salient pre-specified attributes of problems.

Nevertheless, they were unable to generate new concepts or attributes which they could utilize in subsequent conceptual compositions.

In sum, P can be shown to exhibit all of the attributes of intelligence which we prescribed in section 2. Many diverse measures of P's systematic behavior tendencies would therefore be possible. In the remainder of the paper, we will examine a few of these problems in a broadened psychological context.

---

1. See for example the description of stat-rat, a simulated rat in discrimination learning, in E. Lovejoy, Attention in Discrimination Learning, San Francisco: Holden-Day, 1968.

Figure 2.1

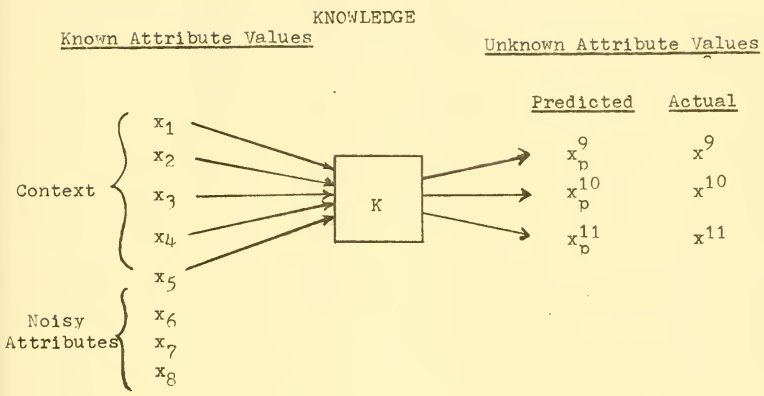
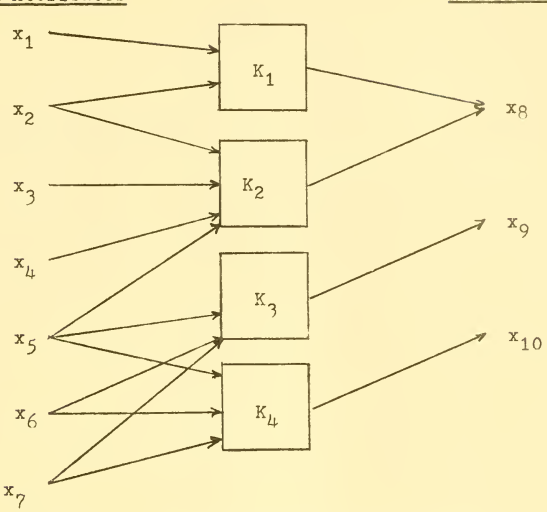

KNOWLEDGE
Known Attribute Values
Unknown Attribute Values
Predicted
Actual
Context
Noisy Attributes
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
K
x_p^9
x_p^{10}
x_p^{11}
x^9
x^{10}
x^{11}

Figure 2.2

TYPES OF KNOWLEDGE



Type	Salient Attribute	Members
1	Predicts x_8	K_1, K_2
2	Known x_5	K_2, K_3, K_4
3	Known x_5, x_6, x_7	K_3, K_4

Figure 2.3

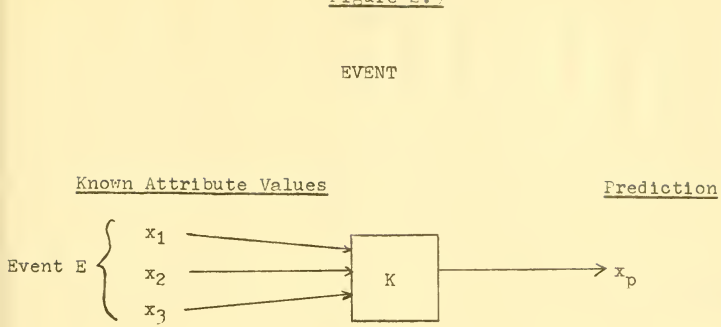

EVENT
Known Attribute Values
Prediction
Event E
x_1
x_2
x_3
K
x_p

Figure 2.4

PERCEPTION

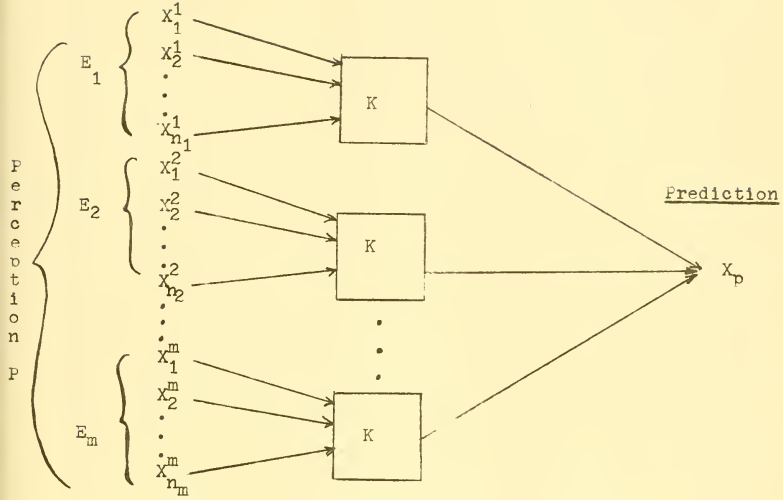


Figure 2.5

SCOPE AND RANGE

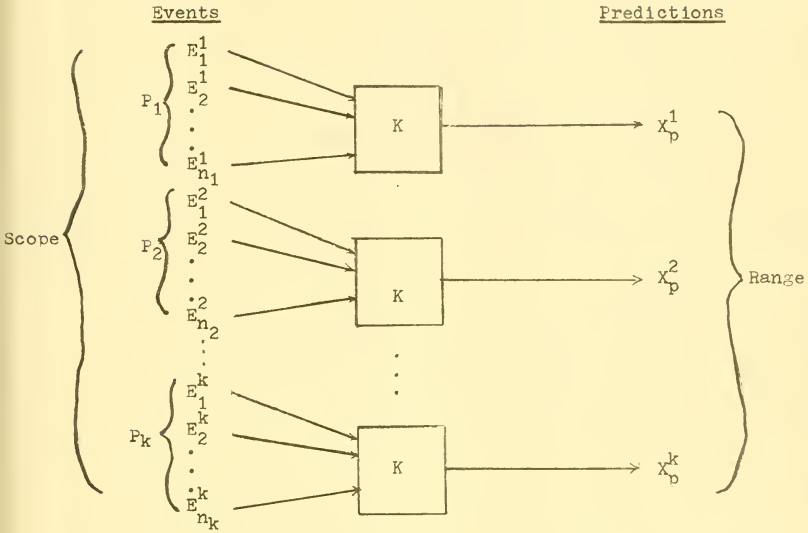


Figure 2.6

DOMAIN OF VALIDITY OF K

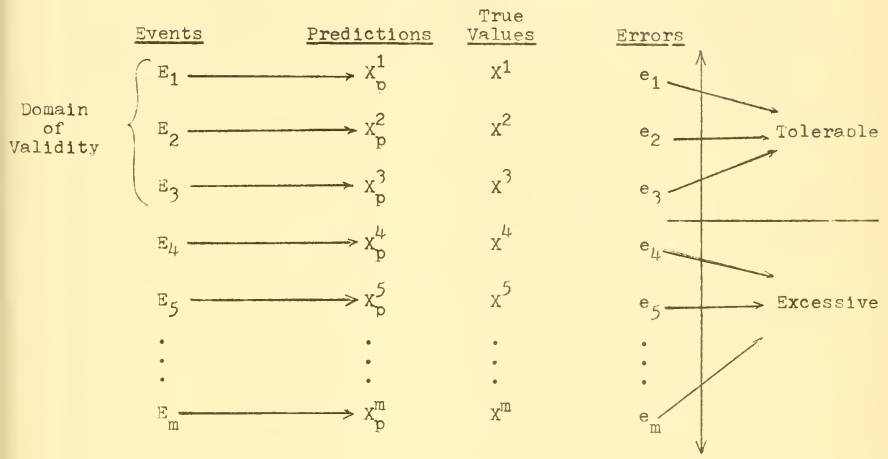


Figure 2.7

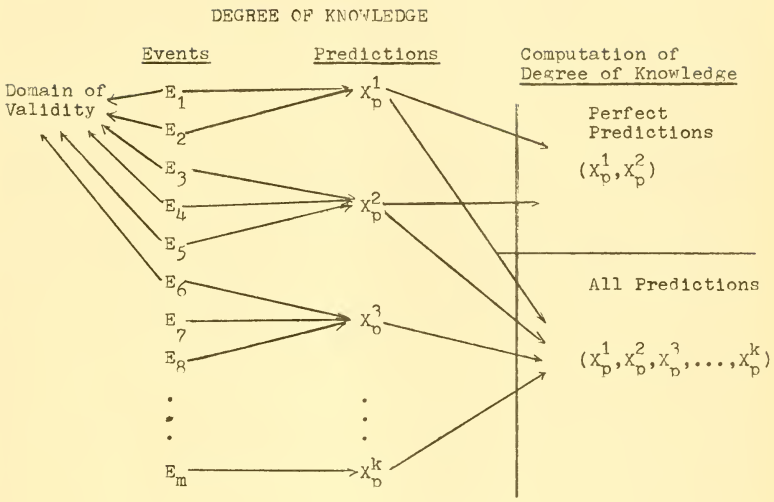
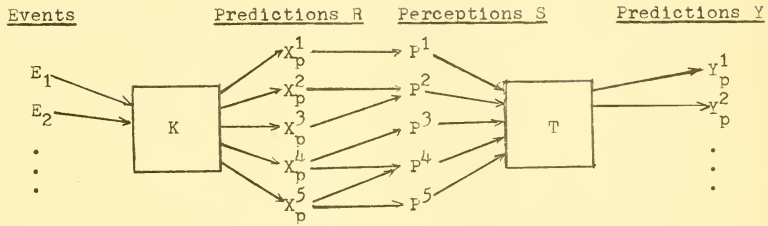

DEGREE OF KNOWLEDGE
Events
Predictions
Computation of Degree of Knowledge
Domain of Validity
E_1
E_2
E_3
E_4
E_5
E_6
E_7
E_8
E_m
X_p^1
X_p^2
X_p^3
X_p^k
Perfect Predictions
(X_p^1, X_p^2)
All Predictions
$(X_p^1, X_p^2, X_p^3, \ldots, X_p^k)$

Figure 2.8

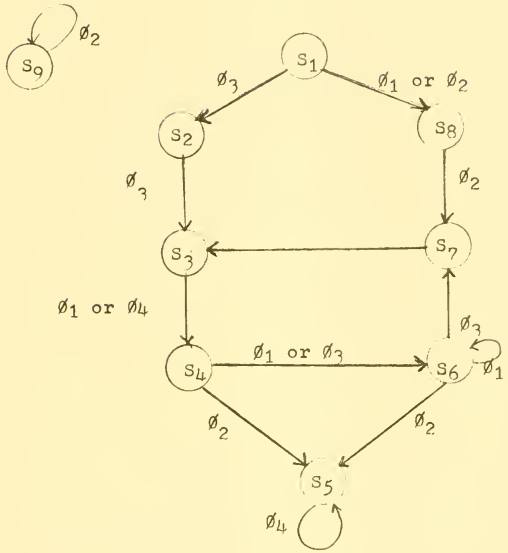
PRECISION OF KNOWLEDGE K FOR PURPOSE T



$$\text{Precision } (K,T) \propto \frac{(P^1,P^2)}{S}$$

Figure 2.9

LOGIC



The S_i are realizable attribute value states within the logic. The $\emptyset_j$ are the operators which are applicable to these S_i and which result in transformations from S_i to S_k.

Figure 2.10

THE LOGIC OF BOOLEAN ALGEBRA

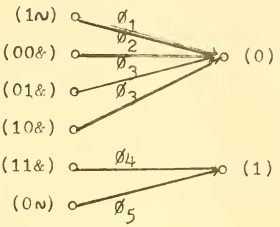


Figure 2.11

THE ENVIRONMENT Θ OF O PARTITIONED
BY THE LOGIC OF TIME

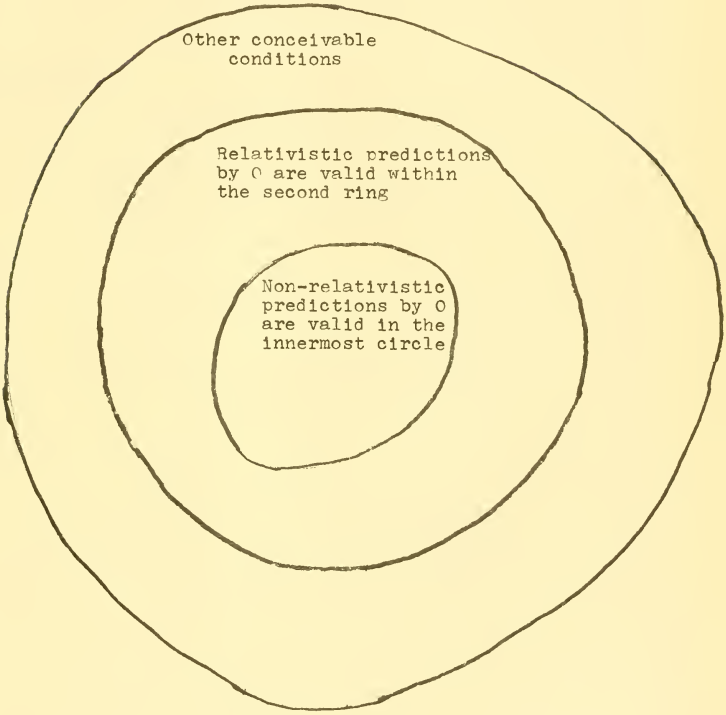
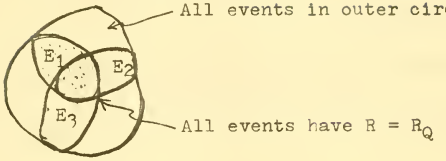


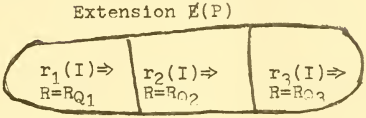
Figure 3.1

TYPES OF INDUCTION

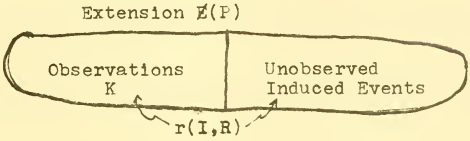
Abstraction



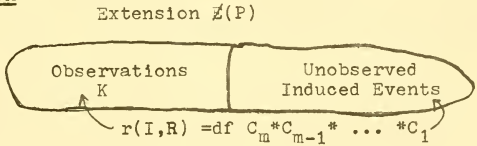
Systematization



Extension



Composition



BIBLIOGRAPHY

1. Y. Bar-Hillel. Language and information. Selected essays on their theory and application. Reading, Mass.: Addison-Wesley, 1964.

2. J.S. Bruner, J.J. Goodnow, G.A. Austin. A study of thinking. New York: Wiley, 1956.

3. J. Deese. The structure of associations in language and thought. Baltimore: Johns Hopkins Press, 1965.

4. E. Feigenbaum and J. Feldman (eds.). Computers and thought. New York: McGraw-Hill, 1963.

5. A. Gurwitsch. The field of consciousness. Pittsburgh: Duquesne University Press, 1964.

6. F. Hayes-Roth. The structure of concepts. Sloan Working Paper 531-71. Cambridge, Mass.: MIT, 1971.

7. E.B. Hunt. Concept learning. An information processing problem. New York: Wiley, 1962.

8. J. McV. Hunt. Intelligence and experience. New York: Ronald Press, 1961.

9. M. Minsky (ed.). Semantic information processing. Cambridge, Mass.: MIT Press, 1968.

10. E. Nagel and R.B. Brandt (eds.). Meaning and knowledge. Systematic readings in epistemology. New York: Harcourt, Brace, World, 1965.

11. A. Newell and H.A. Simon. Simulation of human processing of information. American mathematical monthly, February 1965,111-118.

12. J. Piaget. The language and thought of the child. Cleveland: World Publishing, 1955, 1969.

13. J. Piaget. The Child's Conception of Number. New York: Norton, 1941, 1965.

14. A. Pikas. Abstraction and concept formation. Cambridge, Mass.: Harvard University Press, 1966.

15. W.V.O. Quine. Elementary logic. New York: Harper and Row, 1941, 1965.

16. H.A. Simon and K. Kotovsky. Human acquisition of concepts for sequential patterns. Psychological review, 1963, 70, 534-546.

17. W.E. Vinacke. The investigation of concept formation. Psychological bulletin, 1951, 48, 1-31.

18. L.S. Vygotsky. Thought and language. Cambridge, Mass.: MIT Press, 1962.

19. J. Weizenbaum. Eliza. Communications of the ACM, 1966, 9, 36-45.

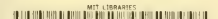
20. N. Wiener. Cybernetics. Cambridge, Mass.: MIT Press, 1948, 1961.

21. T. Winograd. Procedures as a representation for data in a computer program for understanding natural language. MAC TR-84. Cambridge, Mass.: MIT Project MAC, 1971.

22. P.H. Winston. Learning structural descriptions from examples. MAC TR-76. Cambridge, Mass.: MIT Project MAC, 1970.

23. R.S. Woodworth and H. Schlosberg. Experimental psychology. New York: Holt, Rinehart, Winston, 1938,1954.

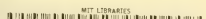
561-71



562-71

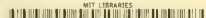
563-71

564-71



564-71

566-71



567-71

568-71

569-71