THE UNIVERSITY OF CHICAGO

A MODEL OF DEVELOPMENTAL AND

PSYCHOLOGICAL PROCESSES

A DISSERTATION SUBMITTED TO

THE FACULTY OF THE DIVISION OF THE SOCIAL SCIENCES

IN CANDIDACY FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

COMMITTEE ON HUMAN DEVELOPMENT

BY

MARK H. BICKHARD

CHICAGO, ILLINOIS

AUGUST, 1973

ACKNOWLEDGMENTS

This dissertation records a part of a conceptual structure that has been developing for many years, making it particularly difficult to adequately acknowledge the many contributions, both substantive and supportive, that have been made to it during that time. The members of my dissertation committee, David Wiley, G. David McNeill, Carol Feldman, and Wilbur Hass, share much of the credit for making the dissertation itself possible, both with their specific comments and suggestions, and with their support for a theoretical dissertation.

The development of the general conceptual structure has benefited from the influence and support of, in addition to my committee members, Sebastian Grossman, Gene Gendlin, Joseph Goguen, Daniel Freedman, Jack Butler, and Marlene Dixon. Among others who made this process possible and enjoyable, I would like to mention Phillip Powell, Bob Marvin, Jim Murray, Karl Pribram, Robert Levine, Starkey Duncan, Joe Cobb, Debby Salisbury, Richard and Barbara Kopp, George Yates, Shag Coker, Charles Cook, and, most especially, Margaret Iwanaga.

Frances Skozen has transformed my hand printed draft into the final typed manuscript, as did Sharon Hanna with an earlier draft.

TABLE OF CONTENTS

Chapter

Systems Psychology

CHAPTER I

INTRODUCTION

Knowing, Consciousness
and Cognition

The fundamental thesis explored in this dissertation is that the Piagetian cognitive developmental stages are naturally emergent from the formal nature of knowing. The general argument is that the logical character of knowing constrains any knowing system to a specific hierarchically organized sequence of (classes of) potential objects of knowing, and that this formal hierarchical sequence can be identified in the cognitive stage sequence described by Piaget (Piaget & Inhelder, 1969).

A second theme in the dissertation is the plausibility of the evolution, that is, the natural existence, of such hierarchically struc-tured knowing systems. It is argued that such an evolution would occur in two basic steps: first, the physical evolution of a species with a particular system structure, followed by the cultural evolution within that species of higher levels of the knowing hierarchy. The biological considerations of the evolutionary argument serve to complement the logical considerations of the argument for the knowing hierarchy, and together they contribute to an understanding of the existence of thought in physi-cal systems--of the relationship between biology and knowledge (Piaget, 1971).

The presentation of the argument begins with a general discussion of the characteristics of knowing processes, and of the constraints that these characteristics impose on any model of knowing. A formal model of a general knowing system is then defined within abstract automata theory, and shown to meet these constraints. Analysis of one of the model constraints leads to an explication of biological stability and adaptability within the automata model, and it is argued that knowing in at least a trivial form is a characteristic of any living system.

This initial development of the knowing model thus introduces the two basic themes of the dissertation--the logical and the biological characteristics of knowing. The next several sections pursue the biological considerations with an analysis of a potential evolutionary path beginning with the basic knowing model. The path consists of a sequence of abstract systems or machines beginning with the automata knowing model, each one a modification of the preceding, and each one more adaptable than its predecessor. The sequence is thus presumed to identify a path of potential system evolution of monotonically increasing adaptability.

The sequence of machines ends with a two-layered knowing model, with the first layer knowing the environment and the second layer knowing the first. This will in a later section turn out to be the critical structure for the cultural evolution of higher levels in the knowing hierarchy, and the evolutionary plausibility of this machine sequence will thus constitute the plausibility of the initial physical step in the evolution of a hierarchically structured knowing system.

Processes within the two-layered knowing model are heuristically identified with reflexive consciousness, and processes in other machines

in the sequence are identified with learning and emotions. These identi-
fications require their own explications and defenses. For the purposes
of this dissertation, however, the function of the discussion is to argue
the plausible evolution of a particular system structure, independent of
the relationship of that structure to standard psychological terms. The
identifications are therefore left at the suggestive level, with their
elaborations and defenses postponed.

In the next several sections, the biological considerations of the
evolutionary argument are set aside and consideration of the formal logical
characteristics of knowing is begun. The first step is to transfer the
basic knowing model from automata theory to Turing machine theory--a more
convenient formalism for these purposes.

Within the Turing machine model, it is shown that the objects of
knowing are necessarily arranged in a certain hierarchical sequential struc-
ture of indefinite extent. Several properties of this hierarchy are
derived which are identical to properties attributed to the Piagetian cog-
nitive stage structure, and the identification of particular cognitive
stages with particular levels in the hierarchy is addressed. The identifi-
cation arrived at suggests several differences in emphasis and conceptu-
alization from standard versions of Piaget's model (e.g., Flavell, 1963).
It is emphasized that the formal hierarchy is derived from an independent
conceptualization of knowing, and thus can serve both as an explanation of
the stage structure and as an independent source of hypotheses about that
structure.

A second theme addressed in these sections is the adequacy of the
formal knowing model to human knowing--can the Turing machine model of

knowing be taken as an adequate representation of human capabilities. The argument occurs in two basic parts: (1) a discussion of the relation- ship between the formal knowing hierarchy and formal logic taken as proxy for human thought, and (2) a discussion of some of the inadequacies of formal logic to human thought, and of the likelihood that the Turing machine model escapes these limitations of formal logic. The general point receives strong support from Turing's thesis in the mathematical literature (Rogers, 1967), a thesis that the general Turing machine model is adequate to any realizable process, and, in turn, the knowing model can be viewed as an exploration of Turing's thesis relative to knowing processes.

The Turing machine model establishes the existence of an abstract knowing hierarchy; it does not establish the likelihood of a physical system which in fact exhibits interesting parts of that hierarchy.[1] The evolutionary sequence of machines began to address this issue, and it is now taken up again in conjunction with the formal knowing hierarchy developed in the intervening sections. Essentially, the discussion of the machine sequence argued the plausibility of the biological evolution of a certain system structure, and it is now argued that a species with such a structure, but not with any simpler structure, could be plausibly expected to develop further levels of the knowing hierarchy in a formal logical sense rather than in a physical or biological sense. It is argued that the critical new capability obtained with this system structure is symbolic language.

----
[1]If the identifications of the formal hierarchy with the Piaget stages are accepted, then human beings constitute prima facie proof of the existence of such systems. They do not, however, establish whether or not such systems are expectable on general grounds--they establish the fact, but not the plausibility, of such existence.

This discussion thus reunites the logical and the biological themes introduced in the early sections defining the basic knowing model: logical considerations argue for a hierarchical constraint on any knowing system, and biological considerations argue for the plausible existence of systems actually manifesting those constraints. The separate discussions of the biological and the logical in the middle sections are thus seen to comple-ment each other in the explanation of the existence of hierarchical knowing systems. A final section presents a general discussion of these results, and of the general strategy of modeling that they exemplify.

CHAPTER II

KNOWING: THE BASIC MODEL

<u>Interactive Knowing:</u>
<u>Characteristics of</u>
<u>the Model</u>

There is a sense of epiphenomenality about knowing in psychology.
It seems to be associated with mentalism in a pejorative sense of the word,
and to be accordingly avoided and considered unsuited for scientific study.
And yet, whatever we want to make of knowing from a phenomenological per-
spective,[1] it is clear that, when we refer to an organism <u>knowing</u> something,
we are referring to something going on in ordinary physical reality--some-
thing that might be worth investigating.

There is an active sense of 'knowing' in which we wish to indicate
that an organism is engaged in knowing something right now. Presumably,
in this sense "knowing" is referring to some kind of physical process or
processes in the knower and some kind of relationship between the knower
and whatever is being known. I will tend to reserve "knowing" for this
active process sense, and to use "knowledge" to refer to a capability of

----
[1]To clarify somewhat my own position: I consider the phenomene-
nological perspective to be a primary perspective on reality of equal
validity to several interpenetrating alternative perspectives--including
materialism. No single view of a tree gives us the whole tree, and,
similarly, I consider that no single perspective on reality is complete--
we have to do some walking around to fill out the picture.

knowing. There seem to be many kinds of knowing, and a conceptual analysis
is clearly in order if we are to keep the meanings straight,[1] but these two
distinctions will be sufficient for now.

Knowing as physical process must be some special kind of process--
not everything is counted as knowing--and at least broadly characterizing
that kind of process would be a reasonable next task. Such a character-
ization is always part definition and part explication; its correctness
can never be proven, only its usefulness and appropriateness defended. I
will propose a characterization that, in one way or another, underlies most
of the rest of the model. I will not at this time give much explicit atten-
tion to the defense of the explication, though its apparent usefulness in
later discussions constitutes some kind of defense, but propose it explic-
itly because it serves as one of the primary integrating concepts for, and
thus helps to make sense out of, much of what follows.

Consider a system in an environment. If the system engages in some
process which is at least in part determined by the environment, and if the
system can in some sense differentiate among the internal outcomes of that
process, then those different outcomes will correspond to differing courses
of the process, and those different courses will correspond to differing
influences from the environment, and, finally, those different influences

----

[1]For example, is all 'knowing' constituted by 'knowing how to';
how is 'knowing algebra' different from 'knowing John'; or, can an organism
know something without having knowledge of it? It is difficult to deter-
mine how many basic meanings there are in the multiple linguistic usages
and variants, yet any analysis of knowing must cope with this in the very
language of the analysis. Unless we remain on a pure language level, the
task is thus one of selection, explication, and modeling among the ordinary
language possibilities--perhaps somewhat like the development of the con-
cept of 'heat' or 'force' in physics or 'set' in logic. See, e.g., Roth
and Galis, 1970.

will correspond to differences in the environment.[1] Thus the different outcomes impose a structure on the environment, grouping together those parts and characteristics that yield the same outcome, and differentiating among those that don't. Furthermore, if the system engages in the process and arrives at a particular outcome, then it <u>knows</u> that it is in one of the environmental circumstances that yield that outcome. This I take to be the basic sense of knowing.[2]

At this level of generality, the explication risks triviality, and it is difficult to understand how it could possibly be adequate. The potential for adequacy, however, is felt through use, and will come at best slowly.

Any living system is engaged in processes that depend on the environment, and these processes eventuate in life or death for the system. Furthermore, life and death are differentiated by the differences in processes that ensue from each. Thus, in a broad sense, knowing as explicated above is an intrinsic characteristic of any living system.[3] What is known may not seem particularly interesting to some--perhaps 'existence'?--and

----

[1]Several questions arise: when does a process stop, what counts as an outcome, and what counts as a difference between outcomes. These are essentially subconcepts in need of explication. They will be dealt with in the construction of the model.

[2]This sense connects for me with primary phenomenological experience: Phenomenologically, I can act (cannot <u>not</u> act), and the 'action' is not fully determined by my initiation. The differences that ensue form the basis for differentiating my reality. Note that this constitutes a kind of active 'implicit definition', as in logic.

[3]To say that knowing is an intrinsic characteristic of living systems is no more to exhaust the subjects of knowing or life than to say that being prime or not prime is an intrinsic characteristic of integers is to exhaust the subjects of primes or number theory. There seems to be a tendency to interpret such statements as exhaustive and reductive.

is certainly not structured or differentiated, but the point is that the basic relationship is there in the simplest life forms, that 'common' examples of knowing are highly sophisticated versions of it, and that the development of these sophisticated versions is intrinsic to the evolution of sophisticated organisms.

The model to be presented is a model of knowing, but that is only one of its characteristics. It is also a model of biological adaptiveness, of consciousness, and others, and it has the ambition to have the potential for saying interesting things about many parts of psychology. Correspondingly, it could have been approached from any of several directions-- general evolution, problem solving, adaptive regulation, etc.--like any other thing, it shows different profiles and different details from different perspectives. It is being developed from the perspective of knowing because that was how it originated, and that is the perspective most related to the application of the model to be made at this time. Other perspectives, especially evolutionary adaptiveness, will remain explicit themes, but most will be unexplored.

The general characteristics of the model, then, will depend greatly on the general characteristics of knowing. I have proposed an explication of knowing, but that constitutes, if anything, only the minimal character- istics for a model to be of knowing at all, and doesn't begin to explore the possible characteristics of knowing--that is, the maximal constraints on an adequate model of knowing. I will develop a few more characteristics of knowing, and argue the constraints they impose on any adequate model of knowing.[1]

----
[1]And, by extension, on any sufficient model in psychology.

It is clear that certain passive processes can be knowing proc-
esses by the above explication--for example, a system can engage in a
recoding process whose outcome depends only on an initial input, and thus
serves to classify that input. But it is also clear that most interesting
knowing processes are interactive in the sense that parts of the process
are conditional on the environmental results, the environmental feedback,
from earlier portions of the same process. A clear example is the inter-
active dance by which the stickleback recognizes a mate--no simple input
suffices, it is rather the responses to earlier outputs that are critical
(Marler & Hamilton, 1966). The same is true, however, for almost any com-
plex activity, from eye-hand coordination to stalking prey to playing chess.

The model, therefore, must be adequate to such conditional inter-
actions. The primary constraint that this imposes on the model is that it
contain some minimal model of the environment within which the conditionals,
the environmental process from system output to subsequent input, can be
represented. Without such a minimal environment, such as in S-R models,[1]
conditional interactions cannot be formally defined or analyzed.

Many conditional interactions are not simply sequential, but are
goal directed in the sense that the system's processes tend toward a goal
condition, and tend to counter or avoid deviations and obstacles on the way.
This familiar fact has myriads of examples--from posting a letter to check-
ing for a tautology in logic--and is hardly in need of demonstration.
Furthermore, as with conditional interactions, goal directed processes are

----

[1]In this and most other cases, the use of examples is intended to
be illustrative rather than critical. In most cases, the models used as
examples make no claim to do what I claim they cannot do.

not in general reducible to or replaceable by simpler processes[1]--much of

what we know about an environment depends not so much on indicators that

the environment has presented to us, but on indicators that we have

'deliberately' (goal orientedly) induced from the environment.[2] Thus the

model must be adequate to goal directed processes.

There is an additional important reason for the necessity of goal

oriented processes in the model--only with respect to a goal condition can

a general learning process be determinate. The general logic is, given a

system that can 'learn', that is, alter its internal processes, those

alterations will be random and arbitrary unless they are with respect to

some standard or rule applied to the processes[3]--and approaching that stan-

dard or following that rule thus becomes the goal around which the learning

is oriented.[4] Furthermore, if the learning is to be functional for the

system, the learning 'goal' must in some sense be equivalent to the improve-

ment of the ability of the system processes to reach their process goals.

----

[1]All possible situations have to have been 'anticipated' for a
strictly sequential interaction to proceed, but a goal directed feedback
system is capable of generating appropriate responses to situations that
have never before been encountered.

[2]That is, we detect and test as well as merely receive.

[3]In other words, if learning is to involve any kind of trial and
error correction, however simple or sophisticated, it must involve some
standard or rule with respect to which 'error' can be defined.

[4]An exception would appear to be the fixed selection of a new
process or process alteration based in predetermined way (by a predeter-
mined rule) on the first or early inputs--i.e., a 'passive' 'learning'
process in the sense of 'passive' above. An example might be imprinting.
There is no feedback and goal comparison in such a process: the match of
the chosen process with the environment is assumed in the predetermined
way in which it was selected. Perhaps it would be useful to restrict
"learning" to goal determined processes, and to classify such exceptions
under the more general term "self-organization." Certainly it can't be a
general learning process--the limitations are severe. See the preceding
footnotes and the following paragraphs.

The issue can be illustrated with the concept of a pattern recognizer. A pattern recognizer is a device that receives inputs--often thought of as organized into a grid or retina--engages in internal processes, and emits an output. The output is taken to classify the input, and a pattern is said to be recognized if all of its instances are classified as a distinct group--similarly, an instance is recognized if it is classified in the 'appropriate' group. Relevant issues include: how do the recognition capabilities of the device vary with differences in the allowable kinds of internal processes, and, how do the recognition learning capabilities of the device vary with differences in learning processes.[1]

The earlier discussion about goal-oriented processes amounts, in this context, to an argument that some things cannot be recognized or 'perceived' without interactive goal directed processes. The issue does not seem to have been addressed in exactly that form, but related discussions and proofs can be found in Minsky and Papert (1969), Hopcroft and Ullman (1969), and MacKay (1969). It is the dependence of learning on goal directedness, however, that is made especially clear by the pattern recognition perspective. The critical question is: how can the recognizer know which of the infinity of possible recognition rules[2] is the 'correct' one--the one it is 'supposed' to learn; how is it to know when its learning is successful and when it has still more to do? Clearly, any particular recognition rule is completely arbitrary from the perspective of the

----

[1]Strictly, the device can be thought of as a responder, and the issues are what kind of response rules can be manifested and learned. Treating the outputs as classifications of the inputs is incidental to the basic process.

[2]Classification rules, response rules.

recognizer, and therefore the 'correct' rule can only be determined with respect to some external standard. Consequently, the recognizer can know if a particular classification is right or wrong, and therefore if its current classification rule is working or not working, only if some comparison with the external standard is made and the result is fed back to it.[1] Furthermore, if the learning is to be effective,[2] it must in some sense seek "correct" feedbacks to the recognition processes as a goal. In other words, only with respect to a goal condition can a general learning process 'know what to learn.'[3]

The necessity for the model to be adequate to goal directed processes imposes a seemingly obvious constraint--the model must include a sophisticated language for defining and analyzing processes internal to the system. Earlier it was argued that the conditionally interactive structure of behavior required a minimal attention to the environment in order to be adequately modeled--enough 'attention' to at least define an environmental feedback. It might conceivably be thought that, as well as

----

[1]Unless some direct comparison with the 'correct' rule itself is possible, in which case we have some kind of copying or duplication process, not a learning process.

[2]If it is to count as 'learning' at all, rather than irrelevant 'noise.'

[3]Note that for the formal recognizer, the rule to be learned is predetermined, and the learning response to "correct" and "incorrect" feedbacks is defined with respect to that rule. In the equivalent biological case, on the other hand, the learning goal condition is the prior element--for example, hunger satisfaction--and it is with respect to that goal that response rules are to be defined as 'correct' or 'incorrect.' 'Correct' is whatever is necessary or at least sufficient to get the goal 'response' from the environment. Note that, even in the strict recognition version, the rule learned need not be identical to the external standard-- it need only be functionally equivalent to it, that is, produce the same outputs for the same inputs. For example, for N an integer, $y = N^2$ is equivalent as a function to $y = \sum_{i=1}^{|N|}(2i-1)$. See Sloman, 1965.

being necessary, an environmental perspective was sufficient to model the complexities of behavior--this, of course, would ignore the processes underlying that behavior, but those processes might be considered a separable problem, perhaps interesting in itself, but not necessary to the analysis of the behavior per se. Some radical behaviorists seem to have at times made exactly this claim.

The environmental, or behaviorist, restriction encounters difficulties even for conditional interactions if the conditions start becoming complex. The difficulty is that a single condition test in the system might be met by multiple possible environmental 'conditions,' and these might not have apparent (intuitive) commonalities when viewed strictly environmentally. In such a case, the environmentalist can at best somehow discover the set of corresponding environmental conditions and simply list them.[1] The situation is identical in the case of goal conditions, though the consequences can be more striking if an 'unexpected' environmental version of the goal is enwountered. It is in modeling goal directed behavior itself, however (the behavior 'on the way' to the goal), that the internal process language becomes necessary: very simply, feedback and servo-mechanism goal seeking processes can correspond to literally infinite (and almost always 'very large') sequences of possible behaviors in approaching those goals--no simple listing strategy will suffice.[2]

----

[1]Process languages provide conceptual redundancies that generate new hypotheses that are at least different from, and generally richer than, those in 'environment' languages.

[2]If 'process' and 'behavior' languages were really equivalent, then nothing would be gained by avoiding the conceptual and notational efficiencies of process languages. If they are not equivalent, then the behaviorist must show that process languages are (1) unnecessary and (2) undesirable. The feedback example indicates their necessity, and their power in expressing and analyzing complex conditional strategies and logics indicates their desirability.

I have argued that knowing processes can be conditionally inter-
active and goal directed, and that these characteristics impose the con-
straints of including the environment and providing sophisticated process
languages on any adequate model of knowing. The characteristic of goal
orientedness imposes still another constraint, though of a different kind.
The goal definition in a goal directed process must come from somewhere.
If it is 'innate' to the structure of the system, then the problem of where
it originates gets transferred to biology and evolution, but, clearly, many
goal definitions are determined in an ongoing fashion by other processes.
Thus we can have some processes receiving goal definitions from others,
these in turn receiving goals from another 'layer,' and so on. But clearly
such a hierarchy must stop: there must be some highest level goal(s).[1]
Furthermore, that highest level goal must in some sense be 'innate' or
'wired in'--there is no higher level process to provide it dynamically.
The necessity for accounting for such a highest 'integrating' goal (all
others can be viewed as subgoals) is the additional constraint imposed on
the knowing model by goal directedness.

In addition, the integrating goal must, in the case of biological
knowing, have some close relationship with survival and adaptability. To
the extent that it is able to maintain its goal condition of 'alive,' a
living system has knowledge of its environment in the sense explicated

----

[1]It might be thought that one process can provide goals for an-
other, which do the same for others, etc. in some kind of network rather
than a hierarchy--that the goal definitions and 'needs' for goal definitions
could fit together like a jigsaw puzzle--and no integrating goal would be
needed. But, at any one time in such a system, there still must be some
'highest level' currently operative goal condition--there cannot be an
infinite regress of goal subordinates in a finite system. The disjunction
of all possible such 'highest' goals is then the integrating goal for the
network, and the problem begins again--it is either 'innate' or provided
dynamically by still another process.

earlier. The greater the scope, power, flexibility, and differentia-
tion of that ability, the greater the knowledge--and conversely. A model
of biological knowing must somehow account for the intimate relationship
between knowledge and adaptability.

The characteristics of knowing that I have developed have been
conditional interactiveness and goal orientedness. The model constraints
that I have argued have been inclusion of the environment, an internal
process language, and a biologically meaningful integrating goal. I find
these views to be similar to those expressed by a number of others--for
example, Bruner et al. (1966), Pribram (1971), and Taylor (1962). The
general viewpoint of analysis by synthesis in cognitive and linguistic
psychology (Neisser, 1967, Halle & Stevens, 1964) seems to me to be
particularly compatible with the above conceptions of knowing, but it is
developed primarily with respect to problems in perception. The most
closely related discussions have been by MacKay (e.g., 1963, 1969) and
Piaget (e.g., 1971). MacKay has concerned himself with interactive models
of knowing and meaning, and has generally used precise and powerful
languages for his models,[1] but seems to have paid relatively little atten-
tion to the biological integration of his general conception. Piaget, on
the other hand, has been greatly concerned with the biological and
evolutionary foundations and contexts of his model, and they are explicitly
interactive, but they have been stated in such imprecise and ambiguous

----
[1]However, I find his interactive analysis incomplete--he analyzes
the 'knowing' or 'meaning' of a set of inputs in terms of the system's
matching responses to those inputs--certainly interactive in spirit--but
seems to leave the concept and process of 'matching' completely unanalyzed.
Also, he, among many other modelers, has a strong tendency to use prob-
abalistic control structures, which have almost always seemed to me to be
concessions to ignorance, rather than potential models of actual processes.

informal languages as to sometimes obscure even the general points, let alone the details. Nevertheless, I find the intent and spirit of Piaget's model to be closely related to my own, and at least the epistemological part of the model to be presented can be viewed in part as a potential formalization of Piaget's model of cognitive epistemology.

Automata Theory

The process language used in the initial presentation of the model is that of automata theory. The model is not a study of automata theory, but rather makes use of concepts and vocabularies from automata theory in its own development. This section introduces these basic conceptual tools.[1]

An automaton is a type of abstract model of a machine or system. It considers that a machine receives inputs, engages in internal processes in some determinate way, and emits outputs; and it attempts to formally represent these processes. In a physical machine or system, at any particular point in time there may be many situations and conditions in the machine which, together with future inputs to the machine, will determine the future behavior of the machine. An automaton model of a machine will abstract all of these simultaneous situations and conditions into a single summary or index symbol, called a state of the machine, so that a single state of an automaton will, together with its inputs, determine the machine's behavior.

An automaton in a particular state will receive an input, engage in various processes, and arrive in a few moments at a new set of situations and conditions indexed by its own state symbol. Since the state symbol

----
[1]Automata theory results and concepts taken primarily from Minsky (1967), and Ginzburg (1968).

summarizes all relevant conditions in the machine, the current state plus the current input must completely determine what the next state of the machine will be. Such a movement from one state to another is called a state transition, and the determinancy of such transitions is expressed in transition rules of the form "if the current state is X, and the current input is Y, then the next state is Z." Thus an automaton is conceived of as moving around in the set of its possible internal states, in accordance with its transition rules and the inputs it receives--an automaton, in fact; is taken to be defined by its set of states and its transition rules.[1]

A useful way of conceptualizing an automaton is in terms of what is called a state transition diagram. A state transition diagram is a diagram of points connected by arrows in which the points represent states of the automaton and the arrows represent the transition rules. The points are commonly labeled with the appropriate state symbols, and the arrows with the input symbol that induces that particular transition.

Many interesting properties of an automaton can be studied without regard for its outputs--considering the automaton as a strictly passive system. An automaton that does have outputs, and for which those outputs are determined by the current state, is called a Moore machine. The concept of a Moore machine will be central to the model.

Many other useful concepts, such as subroutine, servomechanism, etc., can be defined in terms of an automata model, and I will sketch a few of these definitions. In the state transition diagram, for an automaton, there may be many subdiagrams of the overall diagram that are internally

----

[1]Note the discreteness of the automaton model--it jumps from one state to another in response to 'an input symbol.' It is not well adapted to model continuous processes with continuous inputs, though it can often be used to closely approximate them.

identical and differ only in the transitions into them and out of them. These subdiagrams thus engage in identical processes, but are entered from and exit to different places in the main diagram; a set of such identical subdiagrams is called a subroutine.[1]

The exact description of a goal seeking process depends on the goal being sought, and the environment within which it is to be sought, but the general structure will be a goal condition test attached to some procedures for attaining or maintaining the goal condition.[2] These procedures can involve subsidiary tests, other goal seeking processes, etc., and, for that matter, so can the original goal condition test. Note that a goal seeking process may or may not be a subroutine, that is, may or may not have multiple entry and exit paths.

If there is a set of goal seeking subroutines which differ among themselves only in particular characteristics of the goal definitions, then that set can be considered a servomechanism, that is, a goal seeking sub-routine for which the goal definition can be determined external to the subroutine itself--determined, at least, within the set of available goal definitions. If the available goal definitions in a servomechanism exhaust some space of possible goals, e.g., positions on a plane, then the servo-mechanism is of special usefulness in that it absorbs all concerns about reaching such goals (if it works), and leaves other processes only the problem of selecting the goal, not attaining it. The goal attaining proce-dures may, of course, be of varying sophistication and power.

----

[1]A set of subroutines which differ in specific ways in their internal processes can be considered as a subroutine with an argument list.

[2]A goal seeking process must, of course, have outputs.

As an automaton receives a string of input symbols, it will shift from state to state in accordance with its transition rules, with the particular path of transitions being determined by the particular string of symbols being received as input. If an automaton is started in some particular initial state, some input strings will induce various differing transition paths, but will leave the automaton in the same final state when the strings are finished. Other input strings will leave it in a different final state when the input is finished, and so on. If a particular state of an automaton is chosen as a starting state, then, the automaton will differentiate all of its possible input strings into subsets depending on which final state the strings will leave the automaton in--strings leaving it in the same final state grouped together, and strings leaving it in different final states grouped separately. The state that an automaton is in thus carries information about the string of inputs that brought it there.

An automaton in which a particular initial state and some set of final states has been selected is said to 'accept' or 'recognize' any input strings that move it from the initial state to one of the specified final states. Such an automaton with its initial and final state selections is called a Rabin-Scott automaton or a recognizer, and one of the main theorems in automata theory characterizes the kinds of input strings that can be 'recognized' in this sense by an automaton.

This sense of recognition is essentially a passive version of knowing as explicated earlier, and the first step of the model is to extend it to an interactive and potentially goal seeking mode.

An Interactive Machine

      Consider two Moore machines arranged so that the outputs of each one serve as the inputs of the other. Consider one of the Moore machines as a system and the other as its environment, and let the system have the initial and final state selections that make it a recognizer.

      The system can thus recognize input strings in the standard sense in automata theory. In this interactive configuration, however, an input string corresponds to--is generated by--a state transition sequence in the environment. The set of recognizable input strings thus corresponds to the particular set of state sequences in the environment that could generate them. The recognition, or knowing, relationship is thus extended from inputs to situations and conditions in the environment.

      Furthermore, during an interaction, the environment is receiving outputs from the system--and it is these outputs from the system that induce the environmental state transitions that generate the inputs to the system that the system either recognizes or doesn't. Thus the 'recognition' process is no longer strictly passive--the 'recognized' strings are induced from the environment by the system's own outputs. In fact, the interaction doesn't need to be viewed as a recognition process at all. It is equally as much a construction or transformation process--constructing the situations and conditions corresponding to the last state of a 'recognizable' environmental state sequence--or a test or detection process--

detecting an initial state of a 'recognizable' environmental state sequence--and so on.[1]

The system need not be thought of as a single undifferentiated recognizer. It could be, for example, a collection of recognizers connected to each other, say, with the final states of one attached to the initial state of another.[2] Such connections could induce functional relationships among the recognizers, such as one testing for the appropriate conditions for another to begin, or a servomechanism being used to create a subcondition for another process to proceed, etc.[3] In discussing such interconnected systems, an interaction or process that reaches a final state will be said to be complete or successful, otherwise incomplete or unsuccessful; and, once an interaction has begun, strings and 'environments' that will complete it are said to be expected, otherwise unexpected.

----

[1]Note that the system cannot differentiate within the set of environmental states that can begin a recognizable state sequence, and similarly for the set of states that can terminate a recognizable sequence. Of course, some 'recognizers' would differentiate more finely than others, and it is possible that a system would recognize only one single state sequence in the environment--though I am not sure that the system could ever 'know' that to be the case. Note that the final states of a set of recognizable sequences could be a subset of the initial states--so that a test progress, for example, could leave the conditions it tests for unchanged.

[2]There are many details and interesting problems and possibilities involved in the formalization of this and later ideas, but I don't need the formal precision for current purposes, and so will not be concerned with them here.

[3]Note that "test," "transformation," etc. become relative concepts among the processes, e.g., a transformation transforms the conditions tested for by one process into those detected by another, but a particular process could be a test relative to one transformation, and a transformation relative to certain other tests (note the suggestion of a hierarchy among 'tests' and 'transformations'). It is of interest not only to try to formalize such concepts, e.g., must a test be able to successfully repeat itself, or what's the difference between a test and a search, but to figure out how a system could make such distinctions for itself.

Thus, for example, a goal oriented process expects to reach its goal, and is complete or successful when it has done so. "Construct," "test," "transform" and others will be used with the general heuristic meanings suggested above, and, in accordance with standard usage, a transition to a subroutine will be a <u>call</u> to the subroutine.

The model so far constitutes a minimal logical model of knowing: it is explicitly interactive[1] and easily goal directed and it uses a powerful formal process language. It doesn't, however, propose an integrating highest level goal, and certainly doesn't relate such a goal to biological processes. These issues are addressed in the next section.

<u>Stability in an Interactive</u>
<u>Machine</u>

In this section, I will discuss biological stability and adaptability in the interactive configuration, and show that it constitutes a special goal directed case of a recognizer. In this configuration, stability in an environment will turn out to be equivalent to knowledge of the environment.

Biological stability and adaptability correspond roughly to the ability of an organism to persist and survive in an environment--to maintain itself in interaction with that environment. In the interactive automata model, it is not immediately apparent what corresponds to such an interactive stability. The system is constituted in the model by an automaton--a state transition diagram. So, as long as the diagram remains the same, the machine is the same--is stable. But interactions with the environment are presumed to occur in <u>accordance</u> with the transition

----

[1]Passive processes are modelable with transition diagrams involving 'blank' outputs.

diagram, not as operations on it, so it is not clear how the system could ever change, how it could ever be unstable. It might seem that the inter-active model is stable by definition.

Suppose, however, that the machine goes off into some part of the diagram that it can't get out of--there are no transitions back. Then, in a functional sense, the machine is no longer the same because there are processes it was once capable of that it isn't any longer--it can't reach them anymore.[1] Thus change in the formal model is constituted by absorption in some part of the transition diagram, and stability is, conversely, the maintenance of the ability to reach in the diagram whatever could originally be reached.

Whatever could 'originally' be reached, was reachable from some 'original' state--the machine is always in some state or another. Conse-quently, as long as that original state is reachable, anything once reach-able from it is still reachable. Thus, reachability of the original state implies maintenance of original diagram reachability. Conversely, if the original state is not reachable, then at least one state that was once

----

[1]The definition of a state in an automaton is only that it corre-spond to 'possible' conditions of a system, it doesn't guarantee that the system can actually enter the state--that depends on the transition diagram. Of course, when constructing such a model, we will generally ignore 'pos-sible' states that the system can't reach by its own transitions--'non-reachable' states don't seem to really exist at all. But this sense of equivalence between 'existence' and 'reachability' makes it somewhat dif-ficult to realize what it might mean for a machine to be absorbed in some part of its diagram--essentially, it means that some change has taken place that is not reversible by any processes, 'deliberate' or not, of the machine itself, though conceivably (not necessarily practically) reversible externally. Thus, a robot that damages itself might be restored by a repairman, and similarly a person by a doctor (psychotherapist?)--though such abilities can also be exceeded. Of course, the ability to 'call on' (induce) such 'external' aid becomes part of the processes available to the system, and the issue can attain a philosophical subtlety in trying to track down what is 'possible' and 'reachable' and what is not.

reachable, the original state, is no longer. Thus, maintenance of diagram

reachability implies maintenance of reachability of the original state,

and diagram reachability and original state reachability are equivalent

characteristics. In other words, machine stability is equivalent to the

maintenance of reachability of some initial state in the machine.

If we assume that the processes of the machine tend to maintain

this stability in a goal directed sense--tending to 'avoid' dangerous

(absorbing) parts of the diagram, and to 'approach' safe parts of the

diagram[1]--then machine stability becomes the integrating goal for other

goal oriented subprocesses in the system: successful completion of a sub-

process[2] will constitute, for that process at least, successful maintenance

of stability as reachability.

Successful maintenance of stability means successful maintenance

of reachability of some 'original' or 'central' state--and, perhaps, a

return to this central state from time to time. But this makes the central

state into both an initial and a final state in the 'recognizer' sense,

and thus the whole system into a kind of 'cyclic' recognizer. Maintenance

of stability thus corresponds to the receipt of input strings that are

still 'potentially' recognizable, that is, that could still lead back to

the central state, and thus corresponds to a successful interactive know-

ing of the environment in the sense originally explicated. Stability in

an environment is thus equivalent to knowledge[3] of that environment.

----

[1]The assumption here is simply that the final states of sub-
processes tend not to be in absorbing regions of the diagram.

[2]Including, in fact, non-goal oriented processes.

[3]The capability of knowing.

Furthermore, as pointed out before, such knowing is constituted as much by the induction of appropriate inputs as by the receipt of them-- the machine must control the environment as much as recognize it.[1] Knowing requires the ability to counter, or to anticipate and prevent, inputs from the environment that tend toward absorbing conditions, and to induce from the environment inputs that move the system along safe reachability paths in the diagram--knowing thus requires knowledge of what the environment is doing, might do next, and might be induced to do with appropriate outputs. All of these are inextricably present in the transition diagram of a suc- cessfully stable system.

We now have an interactive knowing model with an integrating goal of machine stability. The correspondence between machine stability and biological stability remains to be explicated. The link is essentially that the 'central' state--or some set of equivalent mutually reachable 'central' states--must be compatible with the physical and physiological well being of the organism. That is, maintenance of diagram reachability must accomplish the protection of biologically safe physiological tolerances. But this suggests an obvious identification: let the 'central' states be those states which are themselves most 'central' to the safe physiological tolerances-- the states, after all, are just abstractions for conditions in the system in the first place. Thus 'central state' means 'state of biological well being,' and machine stability corresponds to biological stability.[2]

----

[1]And thus encounters Ashby's law of requisite variety, Ashby (1960).

[2]This relates to Ashby's concept of stability as regulated homeo- stasis, Ashby (1960). Note that one way to enter a central state might be to induce 'blank' inputs from the environment, that is, to quiet or termi- nate an input string--this would correspond to a deviance countering (drive reducing, noxious avoiding) homeostasis. Note also, however, that in this model this wouldn't necessarily be the only way to get back to a central state.

At this point, we encounter the evolutionary perspective.
Biological stability is the integrating goal of all life processes--is per-
haps definitive of it. The process of evolution, then, may be seen as the
elaboration or systems that are stable in new environments, and increas-
ingly stable in all environments--that is, of systems with new knowledge,
and of systems with more knowledge. It is the systems with more knowledge
that I will be concerned with, for I will be interested in some of the
general forms it might take, and in some of the processes that might be
evolved for generating it.[1]

----

[1]Note that 'new knowledge' speciations and 'more knowledge'
speciations can occur independently or together, but that all tend to
increase the knowledge of the total biomass. The view of evolution
involved here is clearly transpecific (as differentiated from speciation
per se and evolution within a single species) and correspondingly involves
the special issues associated with macroevolution. I do not address these
issues explicitly in the text, but the general approach taken is in accord-
ance with Simpson (1949, 1953) and others (e.g., Dobzhansky, 1970, Mayr,
1970) that the general course of evolution is to fill all available
ecological niches, and that microgenetic variation and selection is suffi-
cient to explain this. With regard to the issue of progress in macro-
evolution, however, I do take a somewhat special position. Simpson (1949)
points out an apparent conflict between adaptability and adaptation
(specialization) in the sense that the more adapted a species is to a
particular environmental niche, the less adaptability it has with respect
to changes in that environment--'progress' in adaptation thus appears to
yield failure in adaptability. This divergence or contradiction is a
strongly supported inductive generalization, which seems almost intuitive
after some consideration, but there is no a priori reason why it should be
true with respect to all dimensions relevant to the environmental niches.
In other words, it is clear that with respect to most relevant dimensions
or characteristics of the environment, specialization to one involves a
corresponding lack of adaptation and adaptability to others--the dimensions
or characteristics tend to diverse, e.g., wet vs. dry, or aerobic vs.
anaerobic environments--but it should not be automatically concluded that
this is true for all such dimensions. The position implicit in the text is
in fact that there is at least one such dimension that is relevant to all
environmental niches, and is ultimately 'convergent' with all relevant
dimensions, that is, a dimension along which increasing adaptation is
identical to increasing adaptability. Such a dimension would be a good
candidate as a criterion for macroevolutionary progress. This dimension
could be called something like 'temporal' or 'cybernetic' or 'logical'
complexity, and it is with respect to this (implicit) dimension that the

Note in particular that the model to this point is either inter-
actively stable in an environment or it is not. It has processes and
'behaviors' for maintaining that stability in a dynamic sense, and it has
available to it information about whether or not it is being successful--
whether or not it reaches its central states and subgoals. But it has no
way of changing or adding to any of those processes if that should become
appropriate--it has no way of learning, of improving its own adaptability.

----

path of potential evolution is to be developed in the following pages--that
potential path is essentially a path of increasing adaptation to this dimension,
and the logical analysis which follows is essentially an
analysis of some of the internal structure of the dimension. The analysis
of the dimension and its path of evolutionary adaptation will be primarily
in terms of the cybernetic and logical structure of the two, with little
attention given at this time to the corresponding physiological, and environ-
mental structures.

CHAPTER III

CONSCIOUSNESS: THE EVOLUTION OF KNOWING SYSTEMS

<u>An Interactive Machine</u>
<u>with Interrupt</u>

Entering a central state indicates success in the maintenance of overall stability. Similarly, subgoals indicate an intermediate success, and failures to reach subgoals indicate potential disruption. The model as defined to this point, however, is unable to respond to such indications--success and failure information is available, but the machine has no way to use it.

The simplest use of such information would be to try to recover interactions that seem to be failing. The general idea would be to detect interactions that were not reaching their goals, to interrupt the machine processes that were controlling the interaction, and to transfer control to some other part of the state transition diagram which could then attempt to continue the interaction. Note that as long as the failure detections were not in error, and as long as the transfers had any positive expectation of recovering the interaction, then the interrupt and transfer process would tend to improve the general stability of the machine. A machine which can effect such interrupts and transfers on the first machine will be called an <u>interrupt machine</u>.

The interrupt mechanism is essentially a second machine operating in parallel to the original machine, monitoring the first machine for

errors and 'doing things' to it to improve its stability. The parallel
processing monitor constitutes a basic conceptual addition to the model.
As its capabilities increase, as the things it can do to the first machine
become increasingly powerful, it becomes in turn a learning machine, an
inducer of feelings and emotions, and, finally, a machine which knows and
operates on the first machine in the same sense that the first machine
knows the environment. This latter process of internal or reflexive
knowing will be identified with conscious knowing.

The succession of machines beginning with the interrupt mechanism
will be presented as a possible path of machine evolution. That is, each
machine arises as a 'simple' modification of its predecessor in the
sequence, and might reasonably be expected to have evolved from it, and,
further, it is argued that each machine constitutes an increase in the
adaptability of the overall system. The machine sequence thus constitutes
a potential path of biological evolution both in the sense of being machine
continuous--not involving unexplained qualitative jumps,[1]--and in the

----

[1]To fill in the details of such a machine continuous path would
involve a definition of the path in terms of the actual mechanisms
involved--the neurophysiology in the biological case. The path as
described above is essentially machine functional rather than structural,
and thus does involve qualitative jumps. They are, however, relatively
'small' jumps, at least to my intuition, and would seem to be potentially
as small in a structural sense. Furthermore, I do have some hypotheses
concerning the mechanisms underlying the machine functions, though I will
not pursue them in any detail at this time. For example, the interrupt
mechanism is essentially modelable by any associatively addressable control
system. Pribram (1971) has argued for a hologramic coding system in the
nervous system, which is one of the best possible examples of an asso-
ciatively addressable system, and has shown that it is a reasonable
expectation from certain classes of microstructure in neural systems.
Thus, it seems plausible that, as early 'switchboard' neural structures
became increasingly complex, and as their neural packing density increased,
the advantages of an associatively addressable microstructure would have
yielded the evolution of an interrupt mechanism out of earlier reflex
neural systems.

sense of involving a monotonically increasing gradient of biological adaptability.

<u>An Interactive Machine with</u>
<u>Adaptive Self-Organization:</u>
<u>Learning</u>

The recovery attempts of the interrupt machine will not always be successful: some interrupts and transfers will lead to processes which also fail and require still further recovery attempts. It would be advantageous if the interrupt machine could detect such recovery failures and change its interrupt and transfer rules so that the same transition under the same interrupt conditions would not be repeated--that is, it would be advantageous if the interrupt machine could avoid past mistakes. Clearly, it would also be useful if such rule changes did not unduly interfere with previously determined successful interrupt transitions.[1] Such an interrupt machine would tend to avoid unsuccessful interrupt rules and stabilize around successful ones.

Such stabilized successful interrupts would not really constitute interaction failures from the perspective of the general system, and it would be correspondingly advantageous if they ceased to count as recovery failures for earlier interrupt transitions leading to them--that is, if interrupt transitions leading to stabilized interrupts were counted as

----

[1]Detecting a recovery failure is not a complex issue--a recovery failure is simply an ensuing interaction failure--but there are many possible complexities in the feedback from an interaction failure to the interrupt rules that determined a preceding interrupt transition--e.g., the effect of such a failure on the 'responsible' interrupt rules might decrease with some function of time elapsed between the interrupt and the failure: the principle being that the 'responsibility' for failures decreases the longer it takes for such failures to occur. Another biologically likely possibility is that the feedbacks from recovery failures affect not only the particular transition rules involved, but whole classes of such rules-- e.g., in terms of subroutine structures.

successful. Note that, with this addition, such a stabilized interrupt transition functions identically to a simple state transition: in both cases, when the appropriate state input pair is encountered, a determinate state transition occurs, and in neither case is the occurrence of that transition treated as an interaction or recovery failure.[1]

In other words, a machine with these properties will tend to learn successful new state transition rules[2]--and thereby learn to know corresponding new processes in the environment.[3] There are many issues involved

----

[1]The distinction between interrupt transitions and state transitions becomes somewhat blurred if we consider a machine in which most of the transitions are stabilized interrupt transitions. Logically, I have distinguished three kinds of rules, state transition rules, interrupt transition rules, and (feedback) rules for changing interrupt transition rules, but such logical distinctions need not correspond to biologically distinct entities or processes. For example, if every state transition were a dynamic process of seeking or constructing the next state, whose conditions or principles were changed by any ensuing interaction failure, then the state transition--interrupt transition distinction disappears entirely (all transitions are constructive processes), and the feedback rules become inextricably bound up with the construction process. It seems to me that something like this occurs in organisms, with an initial 'finding the right construction rule'--learning--followed by a stabilization of that rule--consolidation--and that ensuing failures tend to interfere with the consolidation process. Note that the basic construction process could plausibly be the construction, or activation, of modes of neural activity (John, 1967) operating in the hologramic--therefore associatively addressable--coding space mentioned earlier (Pribram, 1971). Modes would then be similar to subroutines in the automata model.

[2]Learning constructions introduce anew the question of what constitutes stability: this can be an important question--it is, essentially, in what sense can a changing (e.g., developing, growing) system be considered to remain the 'same' system--but it will not be crucial for latter purposes, and therefore will not be dealt with here.

[3]Learning rules, of course, constitute a field of inquiry by themselves. Some of the interesting questions which can be asked about them include: what kinds of learning rules are definable? How efficiently can they learn? What physical processes (can) realize them? Which ones are best for which kinds of environments? What are the optimal ways to encode environments for various classes of learning rules? Are there any rules that are logically adequate for all environments?

in such learning processes,[1] but the important characteristic for following sections of the model is that the original interrupt mechanism can new construct successful new state transition rules in the first level machine.

An Interactive Machine with
Harm Avoidance: Pain

One characteristic of the learning machine is that it tends to avoid interrupts--transitions leading to interrupt conditions tend to be discarded as failures. Correspondingly, the machine tends to avoid regions of the state transition diagram that tend to yield interrupt conditions-- transitions leading to such regions tend to be discarded. One example of such a region would be one in which there were many undefined state-input transition conditions, which would simply halt--producing an immediate interrupt--if such an undefined condition occurred. Note that such a sparsely defined region would be a poor representative of the environment.[2]

Suppose there was a special class of inputs that always induce interrupts--i.e., for which no transitions are defined--then, similarly, any behaviors (regions) tending to induce such inputs will produce interrupts and will be avoided. Furthermore, if a particular input, once induced from the environment, tends to be repetitively persistent--that is, repetitive and difficult to stop--and if it in addition never induces

----

[1]Some mentioned in preceding footnotes.

[2]Strictly, the critical characteristic is the probability of entering an interrupt condition, which may or may not be related to the density of undefined conditions in the area. For example, a region in which there was only a single defined transition path, but in which the outputs successfully forced the environmental inputs needed to stay on the path, would not tend to yield interrupts, and thus not tend to be avoided. Note that such a path would constitute a successful representation of its environment.

transitions to subgoals by itself, then such an input will tend to inter-
fere with normal processing, causing the machine processes to cycle around
without being able to reach goals, and thereby tending to eventually
induce interrupts. Behaviors that produce such an input will also tend
to be avoided by the machine.[1] Such an input will be called a pain input.[2]

The essential characteristic of pain inputs is that it is impos-
sible (or difficult) to successfully interact with them--and they are
correspondingly avoided.[3] Such inputs would be especially useful to the
machine if they originated in the physical entity that realized the machine
and were correlated with damage to that entity.[4] The critical character-
istic of pain inputs for following sections of the model is that they
suggest the concept of approach--avoidance, or 'valence,' tendencies within
the system's behaviors and with respect to the system's inputs.

An Interactive Machine with
Internal Feedback: Emotion

I have shown that the learning characteristics induce approach--
avoidance tendencies among behaviors depending on their interrupt conse-
quences. In this section, I will argue that there are strictly internal

----

[1]Note that an input always inducing immediate interrupts, as in the
first case above, is just an extreme case of this.

[2]Note that the interrupt characteristics of an input are in rela-
tion to a particular machine or type of machine--they are not character-
istics of the inputs per se.

[3]Note that many 'normal' inputs might be pain inputs by this
definition if they had (or were artificially given) the characteristic of
persistence. That is, many inputs are, perhaps, successfully interacted
with by being changed to some thing else, and if the change didn't work,
they might become painful.

[4]Note that this is in no way incompatible with a view of pain
signals as having instinctive or reflex withdrawal responses--it simply
focuses on the learning aspects.

characteristics of machine processes that contain viability information--
that is, that indicate that the ongoing process should be sought again or
avoided. Therefore, it would be advantageous if feedbacks of these char-
acteristics could be input to the machine in such a way as to induce
appropriate approach--avoidance tendencies.

The basic idea is that, sometimes inputs from the environment that
the machine is not ready for--require interrupts--indicate that the machine
processes are highly--therefore dangerously--inappropriate to existing
conditions. Such interrupts and associated processes should therefore be
especially avoided. They are characterized by being 'long' or 'detour'
interrupts to some new and 'hopefully' appropriate processes.

On the other hand, sometimes inputs inducing interrupts indicate
rather that the machine was awaiting more complexity or difficulty than it
actually encountered. Such interrupts are not to new processes, but simply
to later, more advanced, or closer to the goal parts of the same process,
and they indicate an overrepresentation of the appropriate environment,
therefore an especially safe situation, and therefore should be approached.
Such interrupts are characterized by being 'short' or 'short cut' inter-
rupts in the processing.

How could such information about the 'length' of interrupts be
generated in the machine? Each change of state in the abstract machine
corresponds to a change of physical condition in the physical machine. The
physical change can be considered as a construction process--the con-
struction of the physical conditions corresponding to the next state.
Presumably, the physical conditions and processes will be related to the
logical process structures--in the sense, for example, that conditions

within a given subroutine are more similar than conditions between differing subroutines. But this means that the amount of construction involved in an interrupt corresponds to the 'length' of the interrupt--long interrupts requiring a lot of construction, and short interrupts requiring little, or even a net decrease in construction when intermediate state transitions are 'skipped over' and their detailed constructions thus avoided.

The interrupt--learning processes thus already contain information concerning the length of interrupts--the only addition (evolution) necessary is a way of encoding it and feeding it back as pain-like inputs for long interrupts, and 'anti-pain' (easy to interact with) or 'pleasurable' inputs for short interrupts, so as to induce appropriate learning tendencies.[1] Such inputs will be called <u>feeling inputs</u>, and considered <u>positive</u> or <u>negative</u> as they are 'pleasurable' or 'painful.' Feeling inputs function like any other inputs in the sense that they induce machine processes attempting to interact with them. Interactions involving feeling inputs will be called <u>emotional</u> interactions.[2]

----

[1]The micro structure of neural modes very likely involve--or consist of--patterns of slow wave potential interactions. The coding of a feeling input in this environment could be accomplished by specialized neurons computing essentially the second time differential of the patterns. The metric could, again, be with respect to a hologramic coding space. Note that learning would involve changes in the structure of the space with respect to such a metric.

[2]The structure of feeling inputs is positive or negative, and perhaps of varying degrees. Thus the multiplicity of differentiations among emotions must in this model be based on other structures of the emotional interaction. To some extent this additional structure is based on other accompanying inputs, and to some extent it is logically arbitrary: thus, for example, a particular negative feeling may be considered as arising from a block to an ongoing interaction (i.e., the original goal remains salient)--giving rise to anger--or it may be considered as the introduction of a new and possibly unsuccessful interaction--giving rise to fear. The logical indeterminancy of reactions to feelings allows a large scope to learning in determining the kinds and tendencies of emotional interactions,

As with learning, this section is intended as primarily tran-
sitional to the next--so many issues involved in the concept of emotion
will not be addressed here.[1] Essentially, to this point I have argued
that a certain kind of variability and selection added to an interrupt
process produces a process that might reasonably be called a learning
process, and that a particular kind of encoding and feedback added to that
produces something that might be called emotion.[2] Independent of the
heuristic, or suggestive, identifications with learning and emotions, I
have indicated that the machine additions in each case were relatively
small and in the direction of increased adaptiveness--and thus constitute
potential evolutionary steps. The next section will be a similar attempt
for a machine addition producing what I would like to call consciousness.

A Reflexive Interactive
Machine: Consciousness

I will argue that, with 'small' additions, the interrupt--learning-
emotion machine described in the preceding sections becomes a second

----

and it is also likely that certain basic emotions have direct neurophy-
siological foundations.

Emotions are closely related to motivations, which are not mentioned
in the model above. Essentially, the model as defined is always active and
always under the control of some set of goal definitions. Particular sub-
goal definitions are often effected by input conditions, and the interplay
between the environment and such goal directed behavior constitutes the
field for the analysis of motivation in this model. For example, a fall
in blood sugar induces signals from the hypothalamus that can only be
interacted with by eating (raising blood sugar)--thus the 'motivation' of
hunger (Grossman, 1967).

[1]This means that I will not at this time explore or defend further
the choice of "learning," "pain," and "emotion" as names for these formal
process definitions--they function for now as heuristic suggestions with
respect to the standard usages of such words.

[2]1 have not decided whether or not I think the concept of mood is
easily construable in this framework.

machine engaged in knowing interactions with the first machine in the same sense that the first machine interacts with the environment. Such reflexive knowing will be called conscious knowing, or consciousness. I will consider this second machine first in terms of its outputs, then in terms of its inputs.

The interrupt machine effects error induced state transitions in the first machine. The learning machine effects new state transitions which stabilize, that is, function like all other state transitions. In other words, stabilized learned transitions contain, or constitute, new state transition definitions (rules). If the machine could separate the transition <u>process</u> from the transition <u>definition</u>, e.g., by initiating a transition, then triggering on its own an interrupt back to itself before the transition process is completed, it could in effect output state transition <u>definitions</u> <u>per</u> <u>se</u> to the first machine.[1]

The interrupt machine is engaged in a constant monitoring of what the first machine is doing and what it is about to do. In particular, if it is about to do 'nothing'--an undefined condition--it initiates an interrupt and attempts to recover the interaction. An extension of this monitoring information, from 'on or off' to 'amount of construction involved,' was posited as a part of the development of feelings and emotions.[2] If

----

[1]It is also conceivable that the transmission of such transition rules from the second machine to the first involves an extension of the feeling inputs, though the connection is not so obvious.

[2]Strictly, the monitoring of 'amount of construction involved' was within the learning machine itself, not in the first machine, yet, as mentioned in an earlier footnote, if most state transition rules are stabilized learned transitions, this distinction becomes void.

this monitoring information is now extended to the actual state transition rules being readied or realized in the construction process, then the second machine will be able to read and receive state transition rules as inputs from the first machine.

Thus the second machine would receive and emit as its input and output signals the state transition rules--the basic functional components --of the first machine. It would interact with and operate on the underlying state transition diagram, taking the structures of and processes within that diagram as the objects of its knowing.[1] Note again that the learning machine already outputs state transition rules--it must only develop the ability to dissociate them from the state transition processes --and that the emotion process already involves the encoding of the length of state transitions--it must only encode their form as well.

Along with its inputs and outputs, the internal processes of the second machine will presumably have changed as specializations of it developed the new forms of inputs and outputs just defined, and the second level machine would function according to its own particular internal rules and heuristics.[2] It is reasonable to assume that such a second level machine would also benefit from, learning and internal feedback,[3] and I will

----

[1]There are several ways this could be formalized, but so far none of them have seemed obviously superior to me.

[2]Still presumably modelable as an automaton if desired. See the preceding footnote.

[3]It is very unlikely, in fact, that the second machine layer would ever have existed as a simple switchboard structure--it evolved out of learning and emotion systems and would have retained those characteristics and structures during its evolution.

in fact assume that both machine layers have such auxiliary machines.[1]

Thus we have two machine layers, each with learning and emotions, one

interacting with the external environment, and the second interacting

with the first.

The adaptive value of such a second machine layer is of many forms.

Perhaps the most basic form is that the second machine layer can 'look

ahead' in the logic of the first machine to check on possible consequences

of (1) current events in the environment, to see if they require anticipatory

behavior, or (2) possible behaviors of the first machine, to see if they

are likely to lead to desirable consequences before actually engaging in

them.[2] Eventually, such a second machine layer would be able to engage

in deliberate (goal directed) learning and planning, and to deal with higher

order logical considerations as discussed in later sections.

The first level machine engages in knowing interactions with the

environment. These interactions will involve perceptions, behaviors,

anticipations of environmental response, goal oriented manipulations, and

so on.[3] If we identify such a basic knowing process of the first machine

with awareness, then the knowing processes of the second machine level

----

[1]This raises a question of why the second machine layer doesn't
(hasn't) develop a third machine layer, etc. Essentially this question,
though from a different perspective, is the focus of "consciousness and
semantic ascent."

[2]Note the assumption that the first machine layer contains informa-
tion, 'knowledge,' about the environment.

[3]Other characteristics associated with awareness may also be
identified. For example, to the extent that the environment is familiar
to the machine, the state transition structures of the relevant sub-
routines and servomechanisms will implicitly represent--will constitute
an apperception of--portions of the environment that are not currently
being directly known--e.g., the 'back' of a tree.

constitute a reflexive awareness or consciousness in the sense of

G. H. Mead (1934). It is in this sense that the processes of the second

machine level will be called <u>conscious knowing</u>, or <u>consciousness</u>.[2]

----

     [1]This would also seem to be consistent with the reflective
consciousness of the phenomenologist.

    [2]Not all uses of the word "consciousness" involve this reflexive
or reflective element--often it is used synonymously with "awareness" or
even "wakefulness."

CHAPTER IV

COGNITIVE DEVELOPMENT: LOGICAL CONSTRAINTS
AND POSSIBILITIES

Logical Constraints

In the preceding sections, I have argued that living systems encounter a certain relatively smooth, increasingly adaptive path of potential evolution. Beginning with the basic automata model as representing the knowing characteristics of primitive systems, I argue first for the development of a parallel processing interrupt mechanism, or associatively addressable control structure, as improving the probability that the system processes are appropriate to the environmental situations which they encounter. As this second system becomes increasingly flexible and sophisticated in its transfers, it takes on the characteristics of a learning system--constructing useful new elements in the primary control structure; and as specializations of such a learning system develop increasing sophistication in the kinds of errors and information they can monitor, they become capable of introducing the general valences or meta-goals of feelings and emotions. Ultimately, these parallel structures develop a fully sophisticated second knowing system with the first system as its environment--thus realizing the reflexiveness of many concepts of consciousness. In a later section, I will argue that, with this attain-ment, cultural evolution can be expected to significantly supplant physical

evolution in the development of more complex knowing structures--and thus that consciousness is a qualitative limit to strictly biological evolution.

At this point, the discussion turns from primarily biological characteristics of knowing to more formal logical characteristics. The primary result of the next several sections is that any knowing system is necessarily constrained within a particular hierarchical structure for its potential objects of knowing, and it is argued that this structure underlies the Piagetian cognitive stages. Examination of the relationship between the knowing hierarchy and the cognitive stages suggests several changes in emphasis from usual conceptualizations of Piaget's model.

Much of the discussion is organized by the issue of whether or not the knowing model is adequate to human knowing capabilities. The primary sources of concern for this issue are that the general knowing model claims such capabilities, and that the attempt to explain the cognitive stages by constraints in the model assumes such capabilities. In addition, for this discussion the basic knowing model is transferred into a different mathematical formalism, Turing machine theory, and the Turing machine version of the knowing model encounters two particularly salient elements of the mathematical literature: Turing's thesis and the halting problem. Turing's thesis is a generally accepted assumption, with strong inductive support, that the general Turing machine model is adequate to any realizable process--which would, in particular, include knowing processes, and therefore supports the general claim for the knowing model. The halting problem, on the other hand, is a paradigmatic problem proving that there exist well defined problems which Turing machines are not capable of solving--such provable limitations, of course, suggest the

possibility that human beings exceed the capabilities of Turing machines, and, in particular, that they exceed the capabilities of the knowing model. The issue is explored first by the construction of the hierarchical structures of objects of knowing,[1] then by examining the adequacy of this structure to formal logic as a partial proxy for human thought, and finally by an indication that Turing machines exceed the capabilities of formal logic relative to human thought.

The Turing machine model is intrinsically ambiguous about whether its components exist as physical systems or as logical systems: the two versions are mutually translatable, and the underlying formal relationships are the same in either case. For convenience, most of the following discussion is in terms of the logical systems, but the resolution of the ambiguity is addressed in a final substantive section. Reintroducing the evolutionary concern of earlier sections, this section asks, essentially, how the various levels of the knowing hierarchy could actually be attained by systems in the physical world. It concludes that the biological evolution of consciousness introduces the potential for certain language capabilities which allow the strictly formal attainment, through cultural evolution, of higher levels in the knowing hierarchy.

Turing Machines

A Turing machine is a Moore machine in interaction with a special environment--a paper or magnetic tape (or some equivalent). The tape is presumed to carry symbols from some alphabet of symbols, and the Moore

----
[1]Thus the hierarchy first arises as an extension of the capabilities of the knowing model, and only later is it considered as a structural constraint.

machine interacts with the tape through a tape head that can read symbols
from the tape, write symbols on the tape, and move the tape in either
direction.[1] The tape is considered to be indefinitely extendable in the
sense that, if the tape head encounters an end of the tape, more tape is
spliced on as needed.

The Turing machine model with its tape environment and the knowing
model with its automaton environment can be viewed as special cases of
each other, and therefore as functionally equivalent models. The system
parts of both models are Moore machines--they differ only in their environ-
ments. Therefore, their equivalence must be shown through the equivalence
of the environments: the general logic is to reduce the tape model to the
automaton model by treating the various tape configurations as descriptions
of automaton states, and to reduce the automaton model to the tape model by
coding the automaton's transition rules on the tape.[2] Given their func-
tional equivalence, one or the other versions of the model can be chosen
depending on their convenience for particular purposes. For much of the
discussion to follow, the Turing machine version is most convenient, and
so I will begin with a discussion of some of the basic results and concepts
from the general theory of Turing machines.

----

[1]Turing machine results and concepts have been taken primarily
from Minsky (1967), Davis (1958), Rogers (1967), and Hopcroft and Ullman
(1969).

[2]Among the many details of such mutual reductions, only one seems
particularly troublesome: an automaton is generally considered to be
finite, while a tape, and thus the number of potential tape configurations,
is considered to be potentially infinite. This would correspond to a
potentially infinite automaton, and potential trouble in the reduction.
The automaton model of the environment, however, is not necessarily of some
restricted size; it's intended to represent all of the relevant environ-
ment, and is thus in principle the 'size' of the universe--a 'restriction'
that should make little difference relative to a Turing machine (whose
tape is also presumably 'limited' by the universe).

The original heuristic for the formal definition of a Turing machine was a person engaged in some determinate symbolic activity. Such a person could, presumably, receive some symbol representation of a problem, operate on those symbols according some set of rules, using as many pieces of scratch paper as necessary, and, finally, produce some symbolic result or answer. The automaton part of the Turing machine is taken to represent the rules, and the tape to hold the input symbols, the intermediate scratch results, and the final results.

The point of the Turing machine definition was to try to explicate the general concept of an 'effective procedure.' A procedure is a specified way of doing something, and an effective procedure is a procedure, or a specification of a procedure, that 'is workable'--in particular, a procedure that contains instructions that are ambiguous or impossible to carry out is hardly 'effective.' A Turing machine, then, is a way of formally specifying a procedure.

Turing's thesis is a hypothesis that the concept of a Turing machine is equivalent to the concept of an effective procedure in the sense that any effective procedure can be represented (specified) by a Turing machine. In technical language, this would mean that a Turing machine could do anything that was doable. Turing's thesis can never be proven[1] because 'effective procedure' is a basically intuitive concept while 'Turing machine' is a formally definable concept: two things can be formally proven equivalent only if both are formalized, but 'Turing machine' is precisely an attempt to formalize 'effective procedure' and proving the adequacy of the formalization is precisely the problem.

----

[1]Though it could be disproven if a more powerful formal model were presented.

Turing's thesis is not without support, however. Several decades of potential explications of 'effective procedure,' some amounting to variations on the basic Turing machine definition, and some taking completely different forms, have all been proven either equivalent to or weaker than the Turing machine. Essentially, the basic issue is considered closed, and the interest is on applications and implications.

Turing's thesis, with its great inductive support, claims that the Turing machine <u>can</u> do anything that can be done, including anything that a human being could do, and the identification with the earlier model of knowing puts special emphasis on any <u>knowing</u> that a human being could do, but the thesis says nothing at all about <u>how</u> it would do it, or what constraints would be encountered on the way. Current results, in fact, have relatively little to say about this problem, and what relevant few there are are specific to computer applications. The exploration of knowing from a Turing machine perspective that I will discuss later can in one sense be viewed as an exploration of the cognitive implications of Turing's thesis, or of the formal constraints encountered in realizing Turing's thesis.

Note that a procedure is a set of steps or rules for doing things. Generally, in interesting cases, these rules will tend to accomplish some result or goal, but there is nothing in the concept of effective procedure as used here that guarantees that the goal will in fact be reached. "Effective" isn't used to mean 'effective in reaching a goal,' but rather 'effective in defining what to do next.' A procedure that does guarantee

results--such as for multiplying two numbers--is called an algorithm, while

one that doesn't--such as for playing chess[1]--is called a heuristic.[2]

One of the most interesting and important positive results of

Turing machine theory is the existence of what are called Universal Turing

machines: essentially, Turing machines that can simulate any other Turing

machines--including themselves--as long as they are given appropriate

descriptions, or programs, for the machines to be simulated.[3] One of the

consequences of the existence of Universal Turing machines is that, if

desired, machines can be studied in terms of their programs for some

Universal machine--and this often is conceptually and notationally very

convenient.[4]

A Turing machine begins with an initial symbol string, engages in

various processes (computations), and ends with some other symbol string.

----

[1]An algorithm for winning in chess is definable--simply explore all
possible game sequences and stay on those that win--but would require
greater than the age of the universe to be computed.

[2]For examples and discussions of heuristic problem solvers, see
Feigenbaum and Feldman (1963), Newell and Simon (1972), or Slagle (1971).

[3]Let T_i represent Turing machine i, w_j symbol string j, and $T_i(w_j)$
the result of T_i operating on w_j as an initial string. Then, Turing proved
the existence of Universal Turing machines, T_u, with the property that, for
any Turing machine T_i, there exists a symbol string w_{Ti}, such that
$T_u(w_{Ti}w_j) = T_i(w_j)$ for all strings w_j. The string wT_i called the program
for T_i--note that a program is relative to a particular Universal Turing
machine. Given a machine definition, there are effective ways of writing
appropriate programs, and, given a program, there are effective ways of
defining a corresponding machine: thus we can, if we wish, consider that
when talking about programs, we are in effect talking about physical machines.

[4]This translatability between 'physical' Turing machines and 'formal'
programs introduces an ambiguity into the nature of existence of Turing
machines. This ambiguity, referred to earlier, does not affect the logical
considerations addressed in the next several sections, but it is relevant
to other considerations and its resolution is addressed in a later section.

The Turing machine computations are stopped only when the Turing machine enters a special halt state, and the result of the computation is whatever is on the tape at that time.[1] The procedure is taken to be undefined on the initial string if the computation never halts. The halt state of one Turing machine can be connected to the initial state of another, giving the possibility of interconnected machines, subroutines, etc., as discussed earlier in the knowing model, but the conceptual problems become difficult as we encounter subroutine classes of machines, classes of subroutines, and so on. In the Universal Turing machine perspective, however, it is possible to represent not only single machines, but also subroutines, etc. by single programs, and to deal with the complexities of machine interconnections with flexible means of passing control among the programs--essentially, we encounter all the power and flexibility of general programming languages.

The most interesting and important 'negative' result in Turing machine theory is the existence of precisely definable, but effectively unsolvable problems. The paradigmatic unsolvable problem is called 'the halting problem': it asks for an effective procedure for deciding if, given any particular Turing machine and any particular input string, the computation will ever halt. The assumption that such a procedure exists leads to a logical contradiction.[2] The study of unsolvable or uncomputable

----

[1]Other halt conditions have been studied and proven equivalent to this one.

[2]Many proofs of the unsolvability of the halting problem exist. I will sketch one--with a few essential, but essentially trivial, details missing--from Minsky (1967): Suppose that an effective procedure for solving the general halting problem exists. Using the notation of the penultimate footnote, call the corresponding Turing machine T_H. Then:

problems is a major part of Turing machine theory, but the major effect of
unsolvability for my purposes is to introduce the constraint that
epistemological and knowing processes be computable--a process for knowing
something is of limited use if it can never arrive at a result.[1]

A Turing machine engages in knowing interactions with its tape
symbol environment in the sense of the formal knowing model--its inter-
actions are environmentally conditional, and its results can be distin-
guished.[2] Knowing in this sense has many characteristics, and some of them
have been especially emphasized in Turing machine theory. Perhaps the most
intuitive characteristic of the Turing machine's processes is that it's
computing a function on tape symbol strings, with the initial string as the
argument and the final string as the value of the function. The results of
such a process can be taken as being 'about' the initial input to the
process: thus a machine that halts with a blank tape is said to have

----

$$T_H(w_Tw) = \begin{cases} 1 \text{ if } T_{(w)} \text{ ever halts} \\ 0 \text{ if } T(_w)\text{never halts} \end{cases}$$

Given T_H, define:

$$T_H{}'(w) = \begin{cases} \text{no halt (cycle forever) if } T_H(_{ww}) = 1 \\ 0 \text{ if } T_H(_{ww}) = 0 \end{cases}$$

Then:

$$T_H{}'(w_{TH}{}') = \begin{cases} \text{no halt if } T_H(w_{TH}{}' w_{TH}{}') = 1 \\ \\ 0 \text{ (halt) if } T_H(w_{TH}{}' w_{TH}{}') = 0 \end{cases}$$

$$= \begin{cases} \text{no halt if } T_H{}'(w_{TH}{}') \text{ halts} \\ 0 \text{ (halt) if } T_H{}'(w_{TH}{}') \text{ never halts} \end{cases}$$

Thus, $T_H{}'(w_{TH}{}')$halts if it never halts, and doesn't halt if it ever halts,
clearly a H contradiction, and we conclude that $T_H{}'$ doesn't exist. But
$T_H{}'$ is easily and effectively definable given T_H (I haven't demonstrated
the effectiveness of the definition), therefore T_H cannot exist either,
therefore the halting problem is unsolvable.

[1]This is a constraint that does not seem to have been considered
among epistemologists.

[2]Distinguished by later computations.

'recognized' the string it started with;[1] or the machine can compute the characteristic function of a set, where a result of "1" indicates that the initial string belongs to the set, and a result of "0" indicates that it doesn't; or, finally, the machine can be taken to be computing predicates on the strings, with a result of "l" (say) indicating "true" and a result of "0" indicating "false." Note that string recognition, characteristic function and predicate computation are all variants of each other[2]--and all are specializations of function computation. Actually, the reductions can be carried out in any direction--for example, consider a function as a set of ordered pairs of argument and value symbol strings, then computing that function is reducible to computing the characteristic function of that set of ordered pairs--all of them are intrinsic characteristics of the general concept of knowing.

The possibility of uncomputability introduces a number of additional distinctions into these characteristics--in particular, there are functions, predicates, and characteristic functions that are not computable.[3] Those that are are called, simply, "computable." Another equivalent adjective is "recursive"; this derives from another attempted explication of effective procedure called 'recursive function theory,' and the equivalence of 'recursive' to 'computable' is a theorem, not a definition.[4]

----

[1]Again, there are many equivalent halt conditions.

[2]For example, recognizing a string is identifying an element of a set, and computing a predicate is computing the characteristic function of its extension.

[3]For a hierarchy of string recognition capabilities, see Hopcroft and Ullman (1969).

[4]Note that recursive procedures--those guaranteed to halt--constitute algorithms.

The essential meaning of 'uncomputable' is that the computation may not necessarily ever halt, but there are various degrees of such recalcitrance. In particular, there is a class of 'almost' computable procedures called "semi-computable" or "partial recursive." Intuitively, a semi-computable or partial recursive procedure is one whose computation will halt for some kinds of results, but not necessarily for others.[1] For example, a semi-computable function is a function whose values may or may not be defined on all strings, and whose computation will halt with the correct value when it is defined, but will compute in vain forever when it isn't. Similarly, a partially recursive set is one for which the computation of the 'characteristic function' will halt and identify an element of the set, but may never halt for a non-element;[2] and the computation of a semi-computable predicate will halt if it's true, but not necessarily if it's false. Note that a value for which a semi-computable function is undefined may never be identified, it is distinguished by the computation never ending, but there's no way to determine that an ongoing computation will never end, it might halt at the next step.[3] Similarly for non-elements of a partially recursive set and false instances of semi-computable predicates. Partially recursive sets have a special property from which their standard name is derived: the elements of such a set can be generated, or enumerated, one by one by a recursive procedure[4]--thus their standard name, "a recursively enumerable set."

----

[1]Note that computable procedures are a subclass of semi-computable procedures.

[2]And is therefore no longer a true characteristic function.

[3]Remember the unsolvability of the halting problem.

[4]Such a generating or constructing possibility is another characteristic of knowing.

Semi-computable predicates also have a special characterization. In terms of formal logic, semi-computable predicates that are not computable turn out to be precisely those predicates that require exactly one existential quantifier.[1] Thus, truth functional and sentential logic are recursive, but quantificational logic may only be partially recursive. From this perspective, partial recursiveness or semi-computability has an interesting intuitive interpretation: To establish an existential quantifier, the machine need only find an instance, but to falsify it, the machine has to search an infinite set, and that search may never end.

There are worse forms of non-computability--functions, predicates, etc. whose computations are not guaranteed to halt under any circumstances--and the investigation of the properties and conceptual structures of non-computability is a major focus of the theory of computation. The critical point for current purposes, however, is that non-computability exists and must be taken into account. The identification of the general knowing model with Turing machines has thus encountered the general support of Turing's thesis and, through the existence of Universal Turing machines, the conceptual power of general programming concepts, but has at the same time encountered formal limits of computability, and thus of knowability.

----

[1]It is the case, however, that the recursive predicates do not themselves constitute a recursive set, and thus that there is no effective procedure for deciding that a predicate requires no quantifiers. (Of course, it's easy to recognize one that doesn't have any quantifiers in the first place.) Another property holds equivalently for both sets and predicates: A set (predicate) is recursive if and only if both it and its complement (negation) are recursively enumerable (partially recursive).

The Turing Machine
Hierarchy

The discussion of knowing has been primarily in terms of the process of knowing, with little attention thus far to what can be known. Turing's thesis, however, and the 'knowing' version of it, forces attention on what can be known by Turing machines, and its presumed equivalence to what can be known by human beings. It is in exploring this equivalence that we encounter a natural hierarchy of Turing machines.

In the general Turing machine model, the environment is presumed to be coded on the tape in the form of various symbol strings. This means that, in principle, rocks, chairs, automobiles, trees, etc. can all be adequately represented on the tape by appropriate strings, and the justification for such an assumption rests in a general mathematical-physical conception of reality.[1] Given such representations, a Turing machine can know rocks, trees, etc. within the limitations of the knowing model, and, in particular, can compute functions and predicates and recognize sets of them.

In particular cases, the details of such representational codings are external to the Turing machine model itself, and become, in practice, engineering questions.[2] There is one special class of objects, however,

----

[1]Note that the complexity of the tape and the tape head are directly related. In a sense, the tape is the 'umwelt' of the Turing machine determined by its tape head. Thus, in principle, the 'tape head' can consist of whatever input-output (sensory-motor) processes we choose, e.g., those of a human infant, and the 'tape' or 'environment' can become as directly intuitive as we wish. Such a process, however, risks burying exactly the issues of importance in the assumptions about the 'tape head' and the 'reality' it encounters.

[2]For example, "How do we best hook up the master computer to the automated factory?", that is, "What is the best sensor and effector arrangement?", or, finally, "What is the best 'tape head' design?"

for which the coding details are not external to, but intrinsic to, and thus constrained by, the Turing machine model itself: The class of Turing machines.

I have indicated how Turing machines can be coded on the tapes of appropriate Universal Turing machines in the sense that they can define and determine procedures and can transfer control among themselves. It is possible, however, that the fact of these programs being on the same tape as the 'environment' is simply a notational convenience, and that they do not in fact constitute a part of the 'environment' in the sense that they can have procedures executed upon them. It is possible, in other words, that Turing machines can determine knowing processes, but cannot themselves be objects of knowing due to constraints inherent in trying to represent them to themselves. The issue is critical because, clearly, human beings can know Turing machines, and, therefore, Turing machines must as well if their adequacy to human knowing is to be maintained.

The possibility is essentially that, in some sense, Turing machine representations are intrinsically 'special' symbol strings, e.g., requiring a special alphabet, that other Turing machines can't operate on. The possibility is refuted in the general case[1] by a process known as Gödel numbering. Gödel numbering is essentially an effective procedure for translating any alphabet, including any extended alphabet 'needed' to represent Turing machines, into one very special 'alphabet': the positive integers.[2] Universal Turing machines can easily be defined with respect

----

[1]It is, of course, possible to design particular cases where such constraints hold.

[2]This is only one of many methods and many alphabets that would serve the same purpose.

to this alphabet, and it follows that Turing machines (now 'merely' integers 'like anything else') can both execute procedures and be the objects of procedures: can both know and be known.

The thoroughness of the method, however, raises a subsidiary question--sort of the inverse of the first--: can Turing machines recognize other Turing machines, i.e., can they distinguish programs (integers) from other symbol strings (integers)? Note that the question is critical only if it is assumed that human beings are capable in the general case of making such distinctions. It turns out, however, that the answer is yes-- the recognition of programs is a computable problem.[1]

At this point we have a general class, or first level, of Turing machines which operate on general symbol strings, and a second class, or second level, of Turing machines which operate on first level Turing machines. Similar questions about knowability can be asked about this second level as were asked about the first, and similar answers are available--inducing a third level. The general procedure is, in fact, indefinitely expendable, and we obtain an indefinite hierarchy of reflexively applicable Turing machines.[2]

This Turing machine hierarchy, in which an element of the N+lst level operates on elements of the Nth level, is the central result of the Turing machine analysis of knowing. Its necessity emerges from an analysis of the thesis that the knowing model is adequate to human knowing--in particular, from the necessity that Turing machines (knowing systems) be potential objects of knowing.

----
[1]Davis (1958).

[2]See the definitions and theorems concerningin $K^{(N)}$ in Rogers(1967). The issue of recognizability for higher levels is not computable.

The Turing Machine Hierarchy
and Formal Logic

     The Turing machine hierarchy has emerged from an examination of the Turing machine 'environment,' and the requirement that it be potentially as inclusive as the human environment. Turing's thesis, however, would also claim that the Turing machine could know about that environment any-thing that a human being could. Such a claim encounters directly the impossibility of proving the thesis, for there is no representative of potential human knowledge available. The issue can be approximated, how-ever, by posing a formal proxy for human thought, and examining the Turing machine's relationship to that. In particular, I will consider the formal logic of predicate calculus and set theory.

     A Turing machine represents a predicate in the sense that it deter-mines a procedure for computing it--for deciding its truth or falsity for a given instance. Recall that truth functional and sentential logic are recursive. Thus all non-quantificational predicates are (algorithmically) computable. Quantifiers, however, introduce issues of non-computability, and it might be asked how Turing machines are to represent uncomputable predicates. Uncomputability, however, does not imply that a corresponding program can't be defined, only that the computations of that program will or may never halt. The program (Turing machine) itself is finite as long as the predicate is.

     Note that, considering Turing machines as representating predicates, a Turing machine which operates on Turing machines represents a predicate about predicates. Thus the Turing machine hierarchy corresponds to the order hierarchy of (nth order) predicate calculus. This gives a deep

additional meaning to the Turing machine hierarchy, as well as to the

sense in which a Turing machine represents a predicate[1]--it serves as the

'object' of higher order predicates.[2]

Turing machines represent sets in the sense of recognizing, or

determining, their elements--again, even if uncomputably so. Taken as set

representations, Turing machines which take other Turing machines as

objects correspond to sets with other sets as elements. Thus, the reflexive

relation in the Turing machine hierarchy corresponds to the elementhood

relation among sets--and the hierarchy itself corresponds to the hierarchy of

logical types.

The Turing machine hierarchy thus bears close correspondences to

natural hierarchies in both predicate calculus and set theory, and the

knowing model together with Turing's thesis suggests that these corre-

spondences adequately model human logical capabilities.[3]

----

[1]Including an uncomputable predicate.

[2]Essentially, we obtain uncomputability hierarchies at each level
of the predicate hierarchy. Little is known about the general structure--
only the first two levels have received any extensive attention.

[3]There are a number of issues in the philosophy of logic and mathe-
matics that are concerned with these correspondences. In particular, are
they injective, surjective, defined on their entire domains, homomorphic,
and--most importantly--are they ontologically reductive. For example, the
correspondence from predicates to sets is not injective, as multiple
predicates can determine identical sets; it is generally considered to be
not surjective, since there are an uncountable number of sets and only a
countable number of predicates (the Löwenheim-Skolem-Gödel-Henkin theorem,
however, would seem to toss some kind of monkey wrench into that argument);
it is not defined on its entire domain, since there are predicates for
which there are no sets (antinomies)--as in Russell's paradox; and, finally,
the homomorphism and reducibility of the correspondence is much disputed.
Most of these issues are relevant to, though not critical to, the Turing
machine model of knowing. Turing's thesis, however, would imply that the
correspondence from predicates to Turing machines was defined on the entire
domain, and I will suggest later that it is not surjective.
    Some philosophers, e.g., Quine (1969, 1970), dispute the existence
of the predicate hierarchy, and thus, of course, the homomorphism and

Turing Machines and
Human Thought

In arguing the adequacy of Turing machines to human thought, I have

used formal logic as a proxy for human thought and exploded the adequacy of

Turing machines to that. The general proposition can never be proven, since

no formal definition of human capabilities can be given. There are,

nevertheless, still two objections I would like to anticipate.

The more fundamental of the two objections is against the adequacy

of formal logic as a proxy for human thought: There are definite elements

of natural language that have not been successfully formalized in logic,

----
reducibility of any correspondence from predicates to sets. I don't wish to
comment on the morphism or reducibility of the correspondence, but I do
have a comment on his argument against the hierarchy. The structure of
the argument seems to be as follows: Predicates are legitimate as long
as they are construed only as open sentences. But if they become values
of variables, as they must if we have predicates of predicates, then we
encounter the necessity of treating predicates as names of something (i.e.,
we do not substitute physical objects, like an apple, for the variable in
"x is red" but rather names of objects, such as "the apple"), and, Quine
asks, 'names of what'? 'Attributes' is a common answer, but Quine objects
that we don't know what attributes are, and, in particular, we can't
determine if different predicates name the same attribute (Quine speaks of
the problem of 'individuation'). Apparently, it would be agreeable with
Quine if predicates could name their own equivalence classes--attributes
could then 'be' the equivalence classes--but Quine seems to object variously
that (1) the equivalence involved is not clear, not well defined, e.g.,
Quine (1970), and (2) the equivalence involved is not for certain deter-
minable (computable), e.g., Quine (1960). My response to (1) is that it
would seem to pose a problem of formal explication, rather than the dis-
missal of a concept. In this regard, the second objection seems to be the
more fundamental--it seems to claim that there is no computable explication
of the necessary equivalence relation (synonymity), and that therefore the
equivalence and any concepts based on it are void. It is at this point
that I feel a strong objection: many concepts in Turing machine theory are
uncomputable, yet none the less exact and useful. It is recursively
unsolvable, for example, whether or not two Turing machines are computa-
tionally equivalent, yet such an equivalence is a well defined concept
(expressable as a Turing machine). The Turing machine knowing model would
suggest, in fact, that the intension of an attribute be taken precisely as a
Turing machine corresponding to the predicate, and that Quine's problem
of individuation of attributes (also propositions), i.e., the equivalence
of predicates, is precisely one of the uncomputability of Turing machine
computational equivalence.

and this would seem to invalidate its adequacy as a proxy. My general

response is to accept the objection, but to argue that in fact the Turing

machine model is more powerful than formal logic, and, in particular, that

it is able to cope with the characteristics that have proven resistant to

standard logic.

I will illustrate the general approach with what are known as

propositional attitudes. These are constructions of the form ". . .

believes that . . . ,""". . . wishes . . . ," and so on. The difficulty they

pose for logic is that they violate the principle of extensionality--the

principle that constructions with equal extensions (referents) are equiv-

alent, and, in particular, substitutable for each other. For example,

Tom thinks that Tully wrote the Ars Magna

may be true, and yet become false when 'Cicero' is put for 'Tully,' even

though Cicero = Tully.[1]

I consider[2] that such constructions require a distinction between

intension and extension, and that the objects of such constructions

properly include intensions. It seems to me, for example, that the above

sentence is appropriately paraphrased as

Tom asserts that the extension of the intension 'Tully' is equal
to the extension of the intension 'wrote the Ars Magna.'

----
[1]From Quine (1970).

[2]In opposition to Quine, though clearly in agreement with those who
call propositional attitudes 'intensional predicates.'

Clearly, the substitution of Cicero for Tully would not necessarily main-
tain the truth value of the paraphrase unless Tom knows that Tully = Cicero
extensionally.[1]

Such paraphrasing requires that the concept of intension be given
some referents, and I suggest that Turing machines (procedures) serve ade-
quately. Thus we can again paraphrase:

> Tom asserts that the satisfier of the procedure 'Tully' is the
> satisfier of the procedure 'wrote the Ars Magna.'

If propositional attitudes take procedures as objects, then substituting
one procedure for another may change the truth value of the sentence even if
the satisfiers of the procedures (the extensions) are equivalent. In partic--
ular, such a substitution will hold only if the subject of the propositional
attitude, e.g., Tom, knows that the procedures have equivalent extensions--
but this is not an algorithmically solvable problem and depends entirely on
the details of Tom's past experience. Formal logic does not consider such
issues of computability internal to its names and sentences--I suggest that
the Turing machine model can.[2]

----

[1]Obviously, this suggests a general insertion of propositions,
attributes, etc. between names and their referents. Many constructions
depend only on the extensions, e.g., 'Cicero = Tully,' and the intensions may
be dropped from consideration--it seems to me to be precisely those
constructions that do depend on intensions that have resisted logical
formalization. See the following discussion above and the earlier footnote
concerning the individuation of attributes.

[2]To carry out a formalization of a system in which all construc-
tions referred to procedures would be a formidable task. The tools, how-
ever, are generally all present. For example, sticking entirely to the
Turing machine model, the sentence above could be paraphrased as: The
system (procedures) 'Tom' has generated as a theorem "'Tully' (x) → 'wrote
Ars Magna' (x)." For those constructions that don't depend on particular
procedures (intensions), there is already an operator in function theory
which yields, in effect, an element of the extension of a procedure--and it
is precisely this operator that generates semi-computable (non-algorithmic)
functions (the minimization operator). It strikes me as slightly ironic
that logic has long tended to paraphrase names as predicates with unique

The second objection I want to consider is that the creativity of human thought is antithetical to the 'sterile' determination of a Turing machine computation. This objection is more fundamental than the first in that it is directly against the Turing machine model of human thought, without reference to any intermediary proxies, and yet is less fundamental in that it rests essentially on a lack of familiarity with, or a lack of understanding of, the foundations of logic and mathematics.

Results of the last four decades have shown that the 'classical' and common conceptions of logic and mathematics as closed and fixed systems is fundamentally in error. In particular, non-algorithmic (heuristic) procedures are capable of indefinitely producing new and unanticipatable elements, structures, systems, etc., and such procedures appear to be as accessible to Turing machines as to human beings.[1]

----

satisfiers--essentially a definition in terms of procedures--and yet doesn't seem to have considered what computability (or lack of it) might do to the substitutability of such predicates.

[1]There are some relatively sophisticated, though, in my judgment, none the less in error, versions of the 'sterility' arguments against Turing machines as sufficient models of human thought. For a review, see Turner, 1971. Such arguments proceed from Gödel's theorem: Gödel proved that in any consistent formal logic of sufficient complexity, there will exist an undecidable theorem--that is, a theorem which cannot be proven either true or false within the formal logic--but which is nevertheless seen to be true from the perspective of a metalanguage for the system--a perspective which 'looks at,' 'refers to,' or 'operates on' the given system as a semantic object. The arguments against Turing machines then generally equate the Turing machine to a formal logic and claim that the human being can find and prove the theorem (from Gödel) that is undecidable for the Turing machine--the human presumably has this advantage because he is able to consider the Turing machine system from a 'metaperspective,' as a semantic object.
    This line of argument clearly assumes that a Turing machine is incapable of semantic ascent, and that assumption is clearly false. It may still be that a Turing machine cannot ascend the semantic hierarchy as powerfully or as completely as a human being--again we encounter the indeterminate and indeterminable edges of Turing's thesis--but it is clear that a Turing machine can ascend to higher levels and can develop new systems of programs (logics) at those levels.

Other objections to the model of knowing could be considered, but there seems to be little reason to believe that they would be any more successful than the many attempts to disprove Turing's thesis in its general procedural form. For current purposes, then, I will conclude that the knowing model is adequate to human knowing, and that a particular reflexive Turing machine hierarchy is a necessary structure in those know-ing capabilities.

The Semantic Ascent Operator

I would like to explore one more question concerning the Turing machine model of knowing: Given the abstract existence of the Turing machine hierarchy developed in the preceding pages, could a particular Turing machine access or realize--at least in principle--that hierarchy in its sub-Turing-machines?

The last line of defense against the sufficiency of Turing machines appears to be one of claiming that a Turing machine that has ascended to a new semantic level is no longer the same Turing machine that it was before. The reasoning appears to be: a Turing machine is a formal logic, so if it changes the logic, it's a different Turing machine. In the first place, this is internally contradictory: if a Turing machine were a formal logic, it couldn't ever change the logic--to be computationally equivalent to one or more logics is not to be those logics. Equating Turing machines to the provable theorems in a logic seems to involve the assumption that Turing machines are at best capable of partial recursive computations-- again clearly false. Turing machines are as capable of heuristic computa-tions as anything else. In the second place, it's semantically arbitrary: There is no more reason to deny continuity to a Turing machine that has just proven a theorem at or from a new semantic level than it is to deny continuity to a Turing machine that has just proven a theorem within an old semantic level, and, furthermore, if an arbitrary linguistic convention regarding continuity of identity is to be adopted with respect to some such kind of change, then consistency requires that the same convention be applied to human beings--and we will have discontinuous new human beings whenever, e.g., someone understands Gödel's proof (see also the section concerning semantic ascent and stages of knowing). Linguistic conventions can, of course, be inconsistent, but they thereby loose philosophical and methodological interest.

A particular Turing machine must be finite, and therefore must contain only a finite number of sub-Turing machines. Thus, it must realize only a finite number of levels of the Turing machine hierarchy. This gives rise to two subquestions: (1) Can a Turing machine construct (that is, can it have a procedure for constructing) new component Turing machines at higher levels in the hierarchy than any of its current components, and (2) given that it has a procedure for constructing Turing machines at higher levels, can that procedure 'in principle' generate the entire class of Turing machines at these higher levels? The "in principle" is used as "without regard for issues of computability": It is clear that such a procedure will have to be heuristic--the levels will be neither recursive nor recursively enumerable.

Any procedure satisfying these two conditions will be called a semantic ascent operator. The rationale for this is that if the strings operated on by a procedure are considered to be the 'referents' or 'semantic objects' of the procedure, then a procedure with the above properties will 'ascend' the semantic levels in the Turing machine hierarchy.[1]

Two further questions can now be asked: (1) Do semantic ascent operators exist, and (2) Can one (or more) of them be identified as operative in man?

It is easy to construct operators that move to a higher level. For example, given, a procedure, construct a new one--at the next level--that 'checks' procedures to see if they are identical to the given procedure.

----

[1]The term was originally motivated by Quine's notion of semantic ascent (e.g., Quine, 1970), though I'm not sure how much Quine would agree with the implied connection.

In other words, construct a procedure to compute the predicate "x=A" where "x" varies over procedures, "A" is a particular procedure, and "=" is a notational equality.[1]

The condition that a semantic ascent operator access 'all' of the higher levels, however, encounters directly some of the more difficult and controversial issues in the philosophy of logic and mathematics. The difficulty is not so much that there are no candidates for a semantic ascent operator[2] (though not generally considered as procedures), but that there is no agreement on exactly what properties such an operator should have-- for example, there is no agreement on exactly what mathematical entities are to be considered as existing, and therefore no agreement on what such entities should be accessible to such an operator.[3] I will not pursue these issues--though the Turing machine perspective should have interesting things to say about them--but will stop with having followed them to the same frontiers as exist for man.

The question of whether or not some semantic ascent operator is identifiable as operative in man is clearly not answerable at this time. I ask it in order to point out some things: If the general hypothesis concerning the equivalence of the Turing machine hierarchy with human cognitive capabilities is true, then an answer to this quest for a psychologically real semantic ascent operator would constitute an explication of

----

[1]That is, construct the name predicate as mentioned in an earlier footnote.

[2]For example, finitistic constructability, Gödel's constructability, definability.

[3]Recall the footnote concerning Quine's rejection of the predicate hierarchy.

the procedure(s) of human learning. This conclusion, in turn, would imply

that human learning involves much deeper issues than are normally con-

sidered by learning theorists--e.g., issues of the computability of

learning procedures, and issues of the range of access of learning proce-

dures. Only recently has the probability begun to be taken seriously that

there are different learning processes (procedures, heuristics) for dif-

ferent areas of learning (Seligman and Hager, 1972). The epistemological

consequences and constraints of learning are still left to philosophers.

Semantic Ascent and Stages
of Knowing

If the initial knowing model is valid, then the Turing machine

analysis shows that any knowing system will exhibit an intrinsic hier-

archical structure in what it can know--in its knowledge. The simplest

case, of course, and undoubtedly the most common, is a system whose

knowledge is restricted to some portion of the first Turing machine level.

Furthermore, the finiteness of any particular knowing system restricts

its possible knowledge to a finite number of the levels of that hierarchy,

and to a finite portion of each of those levels. Development of new

knowledge thus consists of accessing more extensive portions of each level,

and of ascending to new levels in the hierarchy. The possibility of

ascending to new semantic levels of knowing introduces a stagelike struc-

ture into the development of knowledge, and it will be the intent of this

section to examine some of the characteristics of that structure.

Most clearly, a stage structure constituted by ascent through

the semantic Turing machine levels would exhibit a relationship of hier-

archical inclusion between successive stages. In particular, the proce-

dural elements of each stage are taken as objects by the procedures of

the next stage.

One consequence of this structure is that the elements of the
highest level attained by a particular knowing system cannot themselves
be known by that system--they cannot be taken as objects of knowing until
the system attains the next higher stage, until procedures at the next
semantic level have been developed. Such highest level procedures can be
manifested in the system's behavior, they can know lower level elements,
but they cannot themselves be part of the system's known world. From the
perspective of a particular system, such procedures do not exist. Corre-
spondingly, they will be called virtual procedures, and those which can
be taken as objects of knowing will be called real.[1] Development through
the stage structure thus consists of making virtual procedures real and
adding a new level of virtual procedures in their place.

Another perspective on this same relationship is to consider, not
the highest semantic level procedure attained by a particular system, but
rather the highest level procedure operative at a particular time. Such
a highest level procedure will be taking other elements as objects, and
will be operating on, transforming, exploring the structure or implica-
tions of--in general, knowing them. If these objects happen to be other
procedures, then the higher level procedure will in effect be taking these
procedures as representations or images of the elements and characteris-
tics that they know. Correspondingly, procedures being taken as objects
of knowing will be said to be functioning as images, and currently

----

[1]This terminology is motivated by Quine's distinction between
virtual and real sets (e.g., Quine, 1969).

operative procedures will be said to be functioning as plans.[1] Thus, all procedures can function as plans, but only real procedures can function as images.

Another major consequence of the general hierarchical structure of the states is that development through the stages would necessarily realize an invariant order of sequence of the stages. Very simply, no procedure can be operative without the lower order elements for it to operate on. Furthermore, with a constructive semantic ascent operator, no higher order procedure can exist in any sense without lower order procedures out of which it can be generated. Thus the semantic knowing levels generate an ordinal scale of development for any knowing system.

Recall, in particular, that the knowing model was shown to be intrinsic to any biological system, and that it has been argued as adequate to human knowing. Thus we should, on the basis of these analyses alone, expect to find an invariant hierarchical state sequence in the development of human knowing.[2]

These characteristics are, of course, clearly and deliberately suggestive of the cognitive development stages empirically described and studied in the development of the child. The obvious next step is to

----

[1]Clearly motivated by the Plan and Image of Miller, Galanter, and Pribram (1960). This definition differs from theirs, however, in that they would seem to accept, for example, a subroutine return containing information about the current status of the environment as part of the image of that environment, while the above definition tends to accept only general time independent knowledge as part of an image. The distinction becomes a bit fuzzy, however, when we consider the possibility of a currently operative procedure, engaged in interaction with the environment, being simultaneously taken as an object of knowing by a higher level procedure.

[2]Note that the general argument applies both ontogenetically and phylogenetically.

further explore this relationship between stages as emergent from the nature of knowing and stages as empirically manifested in child development. In particular, I would argue that the formal model provides an adequate explication of the developmental model described and analyzed by Jean Piaget. Like any such task of explication, there is no determinate point of completion to such an argument, and it must be halted by extrinsic considerations of time, space, and purpose. My current purpose is primarily to illustrate the plausibility of such an explication, and the discussion will correspondingly be limited to an exploration of the correspondence between the semantic level stage model and the major developmental stages of Piaget's model.

The general task is to determine the assignments of machine levels to cognitive stages, and this depends on the discovery and identification of critical characteristics of the levels in the empirical and theoretical definitions of the stages. There are essentially two characteristics of the machine levels that can be looked for: (1) A machine level is characterized semantically by the elements it can take as objects, and (2) A machine level is characterized computationally by having the potential capabilities of a Universal Turing machine. The two characteristics are not independent: clearly it would be possible to design a machine that contained procedures at a given semantic level, but did not have full computational potentialities at that level. The possibility of such a 'partial' semantic level is thus an empirical issue for any particular system.

Universal Turing machine capability is thus the more general machine level characteristic. If it is accepted as an identifying characteristic, then it seems most likely that the preoperational stage

appropriately corresponds to the first machine level. On the one hand,

a preoperational child clearly has Universal Turing machine capabilities--

some Universal Turing machines can be relatively simple systems (see, e.g.,

Minsky, 1967), and would not seem to provide undue training difficulties

for a conditioning psychologist and a preoperational child. On the other

hand, a sensory motor child would not seem to be capable of Universal

Turing machine capabilities--first, it is questionable if the sensory motor

child has the necessary general learning capabilities,[1] and, even if the

learning is granted, it is further questionable if the general perceptual

and motor systems, the basic contacts with the environment, would be ade-

quate.[2] This is, of course, in principle an empirically testable issue.

If preoperational processes are identified with first machine level

procedures, then the sensory motor stage presents some subsidiary special

questions. It has already been suggested that sensory motor procedures

are not equivalent to a Universal Turing machine, and this leaves two

possibilities: sensory motor procedures are merely 'partial' semantic

level precursors of the 'full' semantic level preoperational procedures--

this implies that both 'stages' are at the same semantic level and differ

only in computational capabilities--or, preoperations are at a different

semantic level than sensory motor procedures and can thus take sensory

----

[1]Much of the learning is of specific kinds, e.g., eye hand coor-
dination, and might possibly be limited to specific neural systems that
lack sufficient generality for a Universal Turing machine.

[2]That is, they may be too labile and/or involve too small an
alphabet. Strictly, the learning and perceptual-motor questions are not
indendent: in the formal Turing machine model, a smaller alphabet gen-
erally corresponds to a larger Moore machine, and vice versa. The 'size'
of a Universal Turing machine is generally taken to be the number of
symbols in the alphabet multiplied by the number of states in the Moore
machine.

motor procedures as objects. The critical question, then, is whether or not sensory motor procedures can be objects for preoperational procedures.

There are many senses in which the sensory motor stage can be seen as the development of the basic input and output subroutines and servo-mechanisms for the latter stages. Eye-hand coordination, locomotion, object indexing, etc., constitute the basic foundational procedures through which[1] preoperations interact with the environment. Such subroutine relationships do not constitute a difference in semantic levels. If this is the extent of the relationship between sensory motor and preoperational procedures, then both are at the same semantic level, and sensory-motor developments constitute a 'partial' level precursor to preoperations-- analogous to the development of a complex interactive tape head in the formal model.

The primary indication that sensory motor procedures might be taken as objects by preoperational procedures is the existence of images in pre-operations.[2] Images might seem to necessarily involve some procedures being taken as objects of knowing by others--some procedures functioning 'as image' with respect to others. There is a critical sense in which even virtual procedures can constitute implicit images, however, and it is still possible that preoperational images are of this kind.

Any procedure represents the environment with which it interacts in the sense that it successfully 'guides' or determines those interactions.

----

[1]Not "upon which."

[2]Note that the relationship of 'taking as object' from one semantic level to another does not preclude 'transferring control to' among procedures across the differing levels. Thus, it is clear that sensory motor procedures are subroutines for preoperations, the critical question is if they <u>can</u> <u>also</u> be objects.

Images would seem to be a special kind of representation that carry structural and contextual information about that environment, presumably derived from past experience. Yet this is precisely the information contained in <u>any</u> 'integrated' procedure, e.g., a servomechanism, whether virtual or real. Such 'contextual' information simply means that the procedure is able to handle many, or all, possible interaction sequences in its environment.

The critical question is functional: virtual procedures can only represent contextual information 'for the sake of' or 'in the guidance of' ongoing interactions--they cannot be used to deduce or anticipate the consequences of behaviors except in the execution of those behaviors. Such a characteristic of a virtual procedure will be called the characteristic of being a <u>virtual image</u>. Furthermore, a virtual image is a representation of its environment, and a set of rules for interaction with that environment, it is <u>not</u> a representation of interactions with that environment.

However, a real procedure, that is a procedure that can be taken as image by a higher level procedure, can correspondingly constitute a <u>real image</u> to that higher level procedure. In particular, the plan procedure can simulate the interactions of the image in order to anticipate new consequences, or it can represent particular interaction sequences of the image, thus representing behaviors within or transformations upon the corresponding environment.

If we now return to the issue of images in preoperations, we find that they are precisely the static non-anticipatory kind that would be expected of virtual images (Piaget and Inhelder, 1971). Furthermore, anticipatory and transformational images, characteristics of real images,

begin with concrete operations--which presumably corresponds to the development of a second machine level. Thus, preoperational images would not seem to entail a difference in semantic level from sensory-motor procedures to preoperations.[1]

The sensory motor stage can therefore be identified with a 'zero' (or fractional, partial) semantic or machine level, and preoperations with the first semantic level. The objects of these procedures would be the perceptual complexes identified by the sensory motor procedures, and the first level procedures would compute various functions and predicates on these complexes. In particular, certain servomechanism procedures, involving rotational and displacement transformations, would represent physical objects--the permanency of the object is implicit in the ability of the servomechanism to produce as a goal any of the perceptual 'versions' of the object from any of its other versions, including hidden ones.[2] Note, however, that this is a virtual representation during preoperations, the real representations are of the perceptual arrays.

----

[1]In spite of the above argument against the logical necessity of operations upon sensory motor systems, my guess is that preoperational images do in fact physiologically involve a limited 'read only' form of operating on sensory motor systems--at least in the occipital lobe for visual systems. The general logic of such a system is described in Neisser (1967). It seems to be fairly common for the organism to use more powerful methods than are necessary for particular tasks as long as the full power of those methods is used elsewhere or later--another example would be the use of visual scanning detection methods for recognition problems that could logically be solved by passive recoding. Note that this possibility does no damage to the above argument--the basic point would be that such a 'read only' operation would be logically equivalent to no 'operation upon' at all, but that it would be more easily extendable to a later 'read and write' operation than other logically equivalent systems.

[2]A formal characterization of object representation is possible, but it is beyond the scope of the current discussion.

With concrete operations, and the second machine level, these pre-operational representations can become real.[1] Thus, for example, physical objects can be taken as explicit entities and various characteristics of the objects, as differentiated from their perceptual presentations, can in principle now be computed.[2] An illustrative example of the effect of the second machine level involves the transitivity of an order relation. A preoperational child can learn to compute an order relation--e.g., match two sticks and pick the longer.[3] The corresponding procedure thus repre- sents that order relation, but only virtually--thus the relation can be computed in any given case, but no characteristics of the relation itself can be deduced: that would involve taking the procedure as an object and computing predicates of it. With the higher machine level, however, the order procedure can be a real representation, and various predicates, e.g., transitivity, computed on it--perhaps using anticipatory images.[4]

----

[1]The Universal Turing machine equivalence of concrete and formal operations seems clear, thus the primary topic of relevance is the reflexive relation.

[2]For example, amount of substance, weight, or volume.

[3]Strictly, this is the computation of the meet in the lattice generated by the order relation, though for most purposes the difference is not crucial. Also, note that the relation for the preoperational child is defined on the object presentations, not the objects per se.

[4]Which involve taking the object representations ('sticks') as objects.

Thus concrete operational procedures can compute functions and predicates on objects.[1] A predicate applied to an object constitutes a sentence, or proposition, about that object, and it is in the form of propositions that concrete operations are usually considered when taken as objects for third level, or formal operational, procedures. In particular, formal operational procedures can generate new, and perhaps hypothetical, propositions, can represent structures of propositions, and can compute deductive chains of propositions.[2]

Thus we find that the first, second, and third semantic levels can be identified with preoperational, concrete operational, and formal operational thought respectively, and that these identifications seem at

----

[1]On object representations, or, by extension, on objects themselves. Marvin (1972) has evidence that suggests that the first real object, in the sense, for example, of being able to internally simulate or rehearse its behavior, is the mother--and that this occurs around age four. Iwanaga (1973) has data which suggest a similar role simulation ability in children's play behavior at age four. These results, with the above model, suggests that concrete operations begins around age four, and that Piaget's more strictly cognitive tasks have picked up a later portion of a horizontal decalage. It is interesting to consider that there might be a similar early social version of formal operations.

[2]Note that this is just a semantic extension of earlier capabilities: a preoperational child can generate new perceptual presentations, can represent (virtually) structures of them--e.g., physical objects, and can compute chains of them--e.g., in tracing the displacements of a hidden object. Similarly, a concrete operational child can generate new (perhaps fictional) object representations, can represent (virtually) structures of them--e.g., propositions, and can compute chains of them--e.g., ordering relationships. It is suggestive that structures of objects or sets of objects are only virtually represented at the concrete operational level-- this implies that the representation of such structures is limited to control transfers between 'adjacent' set or object representations, and this may explain the curious contiguity constraints in groupings (Beth and Piaget, 1966).

least consonant with the major features of those stages.[1] The sensory
motor stage has a different status in this identification, being essen-
tially a partial first semantic level. These identifications constitute
the basic proposed correspondence between the formal stages and the Piaget
stages.

It was suggested earlier that the formal model can account for
the hierarchical relations, the invariant sequence, and the biological
emergence of the stages. There is one characteristic, however, for which
the formal model suggests a difference from the Piaget model, or at least
a difference in emphasis. This difference concerns the qualitative nature
of the stages.

The Piaget stages are often identified with the mathematical
structures that are manifested within them--especially, concrete operations
is the stage of groupings, and formal operations is the stage of groups.
It is not always clear just what is being emphasized here--the mathematical
structures per se, or the objects out of which they are constructed--but
the general emphasis seems to be on the structures. As if, for example,
the abstract structure of a grouping were definitive of concrete opera-
tions. The formal model suggests exactly the opposite emphasis--the stages
are defined by what can be taken as objects of the structures, not by the
structures themselves.

In order for this to be a non-vacuous distinction, it must be
claimed that the objects are specific to particular stages--that much has

----

[1]Note that there are an indefinite number of formal semantic
levels, and that only three have been accounted for. This suggests the
possibility of post formal operational stages, and Powell (1971) has
evidence that just such a stage may be found in a small percentage of the
population.

been argued in the development of the formal-to-Piaget correspondence--
and that the abstract structures are not similarly specific. This latter
point is actually quite easy to make: for example, we find a group of
displacements on arm position in the sensory motor stage, a group of
locomotor displacements on body position in preoperations, a group of
addition on the integers in concrete operations, and a group of logical
operations on propositions in formal operations. In each case, the abstract
structure is a group, but the elements involved are qualitatively very
different. Piaget has posited differing structures for the same stage--
e.g., groupings and groups in concrete operations, of groups and lattices
in formal operations--and the same structure in differing stages--e.g.,
groups in all stages--but it does not seem to have always been explicitly
realized that this implies that the structures per se cannot be the
definitive characteristics[1]--it must be the elements. The formal model
suggests that it is the semantic level of the elements.[2]

The discussion has focused on the basic stage structure and char-
acteristics of the Piagetian model--clearly the major task of applying
the formal model to cognitive development remains. The apparent success
of this partial application suggests that such a general application might
be both feasible and desirable. Note especially that the formal model has
been derived independently of the general cognitive and developmental data,

----

[1]For example, Flavell's (1963) discussion of the "integrated char-
acter of stages." What does the 'integrating' are the structures, not
the elements. Strictly, particular structures with particular elements
are usually specified, e.g., the IRNC group on propositions, but the
emphasis is still on the structures.

[2]This would suggest that all other structures could in principle
also be manifested at any stage, e.g., lattices or groupings. Finding
elements at the appropriate semantic levels that exhibited appropriate
structural properties, however, would not necessarily be an easy task.

and thus offers an independent source of hypotheses and conceptual distinctions. In particular, if the knowing model is valid, then the formal stage model <u>must</u> hold in some form or another, even if not identically to already described stages, and, furthermore, it serves as an explanation of the characteristics and relationships that can be derived from it. Conversely, success of the model in explicating and suggesting empirical results constitutes inductive support for the validity of the knowing model.

CHAPTER V

HIERARCHICAL KNOWING: POSSIBILITY

AND PLAUSIBILITY

Consciousness and Semantic
Ascent

There is an apparent contradiction in the fact that the semantic

stage model contains an indefinite number of levels, while the conscious

knowing model contains only two levels. To be sure, the semantic levels

are permissive rather than prescriptive--there is nothing in the model to

prevent a system from realizing a small number of levels--yet I have al-

ready identified three levels as operative in man in correspondence with

the cognitive stages. How is the existence of more than two levels to be

explained?

A preliminary response to the question consists in pointing out

that in the Turing machine model the indefinite number of semantic levels

were developed in the context of only one machine--a Universal Turing

machine. That is, the semantic levels are logical in character, not physi-

cal, and even a single physical machine can suffice to the semantic hier-

archy as long as it has Universal Turing machine capabilities.[1]

----
[1]Note that from the perspective of a physical machine layer, the
many Turing machines of a corresponding semantic level are sub-machines,
i.e., subroutines, to be enacted and transferred among according to the
control structures. A physical Universal Turing machine can thus be
viewed as a potential substrate for the enactment of an indefinite number
of logical Turing machine definitions. In particular, a physical level
in the consciousness model is presumed to correspond to a large number of
machines of the corresponding semantic level.

But this only reconstructs the anomaly in the opposite direction: if one physical machine level is sufficient, why posit two of them? Or, if it is accepted that two do in fact exist in man, why should the second, and presumably superfluous, machine have evolved?

The answer to this is more involved. Consider a Universal Turing machine operating at only one semantic level. Such a machine can in principle have programs of other Turing machines on its tape, and can ascend semantic levels by constructing further programs that operate on the original ones. The first step, however, is the existence of Turing machine codes on the tape--that is, in the environment--and preliminary to that is the development of a language within which such codes can be written. Turing proved that machines are definable for which such programming languages are possible--they are now called Universal Turing machines--but this says nothing at all about how such a language might develop or evolve in a natural setting. I will argue essentially that such a language is unlikely to evolve except in machines (systems) that already have two physically realized semantic levels. Once evolved, however, the language would be available for ascents to higher semantic levels without corre-sponding physical levels.

In other words, the ascent to the second semantic level of a Universal Turing machine is logically possible either through a second physical machine level or by means of an external language. In the development of the consciousness model, I argued the plausibility of the evolution of a second machine level. I will now argue (1) the implausibility of the evolution of an appropriate language in (a species of) a one layer machine, (2) the plausibility of the evolution of such a language in a two

layer machine, and (3) the likelihood that further semantic ascents would be in terms of the language rather than the physical machine, once the language was available.[1] The critical issue, then, is what bearing the number of machine levels has on the evolution of a language.

A program on a tape--in the environment--can, as a program,[2] be taken in only two possible senses: either as an initiator and controller of ongoing interactions, or as a static or 'timeless' definition of a procedure. I have shown that this second sense is available only with the existence of higher level procedures. But this means that the programs for a single layer machine must always be enacted whenever the machine encounters them.

The Universal Turing machine manages to avoid the potential dif-ficulties of this restriction by carefully controlling when it is 'ready' to respond to a program, and by carefully controlling where the tape head is on the tape when it enters such a condition of readiness.[3] Thus a Universal Turing machine does always respond to its programs whenever it encounters them, but, through the restriction of the tape head to a single tape mark at a time, and with sometimes elaborate 'bookkeeping' about where it's at, the Universal Turing machine manages to encounter programs as programs only when appropriate.

Thus, in a natural or ecological setting, the evolution of an adequate programming language among single layer machines involves the

----

[1]Thus tending to limit the physical evolution to two levels.

[2]They can, of course, be taken as 'meaningless' elements.

[3]A Turing machine interaction generally involves a lot of spacing to appropriate positions on the tape.

evolution of a language that is always 'responded to' (enacted), whenever
encountered and either the simultaneous or the delayed evolution of an
appropriate apparatus to restrict those encounters. Such an apparatus is
not particularly trivial, and it seems clear that it would have to be
designed, not evolved. It is perhaps conceivable that it could be
designed with the aid of the language,[1] once it had evolved, but I will
argue that no such language is likely to evolve.

The primary restriction of such a language is that it must always
be responded to--the programs it names or defines must always be enacted.
This means that such a language is only appropriate to situations and con-
ditions in the immediate environment. Anything outside the immediate
environment is not available to be interacted with, and thus any language
usage relevant to such non-immediate situations would induce inappropriate
behavior in any systems (organisms) receiving the language. It is plausible
that a few signals of fear, anger, pleasure, and perhaps a few social inter-
action signals might evolve under such constraints, but the evolution of
much more than that, and, in particular, the evolution of a language ade-
quate to general programming needs, seems extremely unlikely.

Thus the evolution of an adequate programming language is not to
be expected among single machine level systems, and any evolution of a
second semantic level would be expected to occur in terms of physical
machine levels--as in the consciousness model.

----
[1]Though not to me.

With two semantic levels, however, a language element can be taken either as invoking an environmental interaction, or as naming or defining a general procedure.[1] In the later sense, the language element can be construed as invoking an internal interaction of the knowing (perhaps the locating or constructing) of the procedure defined. Such language elements can be used to identify and define procedures that are relevant to situations and conditions external to the immediate environment--without inducing inappropriate immediate behaviors. Furthermore, to the extent that such language constructions can be stored, in memory or cultural transmission or in writing, they can serve to record and store general knowledge, and for the indirect transmission of knowledge from one system to another.

The usefulness of such a language seems clear, and the plausibility of its evolution, once possible--i.e., once two machine levels exist, seems equally so. The probable setting for its evolution would be social interaction, and a potential path of evolution would involve the social coordination of activities relevant to increasingly removed situations--the need

----

[1]I take this functional distinction to be the most useful distinction between signs and symbols. Thus any element invoking only a virtual procedure could only be a sign, while a symbol could invoke either a real procedure or a higher level--perhaps virtual--procedure that would 'know' the corresponding real procedure. Symbols, then, could only exist for a system with at least one real level of procedures, that is, with at least two semantic levels, that is, for a concrete operational but not a preoperational child. The above argument thus claims that a language with strictly sign functions is not likely to evolve to any complexity, while a language with symbolic functions as well could evolve to a general programming level. In this perspective, the intension of a symbol could be construed as the real procedure to which it corresponds, and its extension to the extension or domain of that procedures. The same distinction could be made for a sign, but only for a symbol can the system itself make such a distinction. Such a usage of "sign" and "symbol" is different from Piaget's usage in two major senses, but I do not have the space to analyze and argue those differences at this time.

for such communication among social predators has often been noted.[1] As

such a language became adequate to the coordination of arbitrary activ-

ities, it would thereby become adequate, at least in principle, to general

programming functions.

Furthermore, the existence of such a language would allow cultural

evolution to supplement biological evolution in the ascent to new semantic

levels, and the speed of cultural relative to biological evolution would

result in, in effect, a supplanting of biological evolution by cultural

evolution with respect to semantic ascent above the second semantic level.[2]

----

[1]Note that this does not claim that the dimension of 'externality'
of referent would necessarily be the only relevant dimension in the
evolution of such a linguistic semantics. It does not even necessarily
claim that it would be a primary dimension--though that claim might be
argued--but only that it would be a salient dimension. The discussion
above does not consider the syntax of such a language, though it is clear
that it would have to have one if it was to be more than just a list of
program names, and it might prove valuable to examine syntax from such a
programming language perspective. This would not be equivalent to looking
at computer programming theory for at least two reasons: (1) computer
programming languages are restricted so as not to unduly burden their
interpreters or compilers, and (2) much more importantly, computer program-
ming languages are intended for the function of defining programs, while
most language usage is involved in a far greater range of perlocutionary
and illocutionary acts (Austin, 1962)--and undoubtedly evolved for such
functions rather than for definition. The above analysis suggests, how-
ever, that not until the more purely locutionary use of language of
'definition' was differentiated was it possible to use language to move to
a higher semantic level--i.e., formal operations. Examining illocutionary
and perlocutionary interactions between Turing machines, however, might
well give insights to natural language.

[2]Undoubtedly general cultural evolution in terms of an extended
language community would be the dominant form of this, but there is nothing
in the analysis to prevent particular individuals from developing and using
for possible semantic ascent their own specific externalizing representa-
tions--not strictly languages in the social sense, but serving a similar
function for semantic ascent. Such idiosyncratic processes would in general
be expected to be heuristic and intuitive, and not limited to knowledge at
new semantic levels. It seems likely that most knowledge is obtained more
than once in such an intuitive 'virtual' sense before someone is able to
artiuclate it (encode, externalize) in a form suitable for transmission to
the general language community.

This analysis would therefore lead to the expectation that cognitive operations (second semantic level) were biological in origin and perhaps in part dependent on neural maturation for their timing, and that formal operations (third semantic level) were historical in origin and dependent primarily on cultural experience and training for their timing.

The resolution of the 'contradiction' between an indefinite number of semantic levels in the formal cognitive model and only two machine levels in the consciousness model thus rests upon (1) a distinction between logical semantic levels and physical machine levels, (2) a second look at the function of programming languages in semantic ascent, and (3) an analysis of the effects of the number of machine levels on the probability of language evolution. Together with the preceding section, in fact, this discussion has shown conscious knowing to be an integral part of cognitive development: the first operations of a second machine level is taken to be definitive of concrete operations, the general reflexive relationship of consciousness is taken to be identical to the relationship between adjacent stages, and reflexive consciousness is found to be essential to language processes allowing ascent to higher stages. Thus, two separate developments of the basic knowing model are shown to have an essential integration.

CHAPTER VI

CONCLUSIONS

Systems Psychology

Beginning with a basic model of knowing, I have demonstrated that
any knowing system is constrained by an indefinitely extending hierarchical
structure of potential objects of knowing, and have argued that this struc-
ture is sufficient to account for the cognitive developmental stages. Thus,
I have argued that the cognitive stages are emergent from the essential
logical character of knowing.

Knowing was shown to be a characteristic in at least a trivial form
of any living system, and a path of potential evolution beginning with a
basic knowing system was outlined. This path was later found to constitute
the evolution of the second level of the logical knowing hierarchy, and
it was argued that higher levels of the hierarchy could be attained in a
strictly formal sense once this basic two level system structure had
evolved. That is, the path of general evolution contains the knowing hier-
archy as one of its dimensions, and it can be expected that the first two
levels of the hierarchy will be attained in a physical sense, and higher
levels in a formal sense.

Thus, I have demonstrated the logical existence of a hierarchical
constraint on knowing, the plausibility of the physical existence of systems
actually manifesting those constraints, and the potentiality for identifying
those systems and constraints in the human cognitive developmental stages.

It is conceivable that the model of knowing that I have proposed is weaker than human knowing, that it is subject to limitations that do not hold for humans, and therefore, perhaps, that the hierarchical constraints on the knowing model do not apply to human beings. This would, of course, invalidate the constraints as an explanation of the cognitive stages. Demonstrating this, however, would involve disproving Turing's thesis in its general form, as well as the more specific arguments I have given for the adequacy of the knowing model to human knowing. This does not seem likely.

In general, the internal validity of the argument seems to me to be difficult to question, including its applicability to human beings. Details of the particular correspondence proposed between semantic levels and Piagetian cognitive stages may be in error--these details are not derived from the formal knowing model, but constitute independent hypotheses in their own right[1]--but the general existence of such a correspondence seems assured.[2] This correspondence with the Piagetian model would seem to constitute a confirmation of the primary conclusion of the argument--that is, the cognitive stages would seem to instantiate the formal knowing hierarchy.[3]

----
[1]For example, the conclusion that the sensory motor period does not constitute a full and independent semantic level could be invalidated by additional data.

[2]Strictly, the applicability of the semantic hierarchy to human knowing seems assured. It is conceivable that this constraint is somehow unrelated to the cognitive stages, but this seems unlikely in the extreme.

[3]Conversely, the correspondence of the results of the logical and biological considerations of this discussion with the Piaget stages would seem to provide support for the Piagetian model.

Clearly, this application to cognitive development does not exhaust the potential scope of the model. First of all, even the cognitive application is given only preliminary attention--the correspondence to the cognitive stages would seem to be a nontrivial application, but, nevertheless, there are clearly vast areas of Piaget's model in particular and cognitive development in general that remain to be considered from the perspective of the knowing model--and, in addition, the identifications with learning, emotions, and consciousness remain to be explored and defended.[1] Relative to the potential scope of the model, the application to the Piaget stages is a narrow base of confirmation: confidence in the applicability of the general model to psychology will be enhanced by the development of further applications, by the broadening of that base.[2]

The structure of the model as it has been presented begins with knowing as a primary model, considers biological and logical characteristics of that model, then integrates these two characteristics and applies the results to the Piaget stages. From a broader perspective, however, the basic model is constituted by the learning, emotion, and consciousness models--a model of general psychological processes. Knowing then becomes a characteristic of this basic model through which it can be applied to cognitive development, and biological evolution becomes a dimension of coherence internal to the model. From this perspective, the presentation of the model is very incomplete: the missing explorations of learning, emotions, and consciousness become, not simply supportive, but critical to

----

[1]However unexplored and undefended, these identifications do not seem to me to be entirely unmotivated in the discussion.

[2]For example, the applications to psycholinguistics suggested in the footnotes of the section 'consciousness and semantic ascent.'

the foundations of the model. In spite of its incomplete presentation,
however, this would seem to be a more fundamental way of viewing the
model--an explication of primary psychological processes which can then be
used as a foundation for the analysis of higher order processes. From
this perspective, the potential scope of the model is greatly enlarged:
the characteristics of motivation and cognition become directly relevant
to the basic model, as well as the issues of physiological realizability
and phenomenological plausibility, and the range of potential applications
of that model should cover much of psychology.[1]

The model as presented, in fact, is intended as much as an illus-
tration of a method as a presentation of specific results. Cybernetic
systems theory contains languages and results that are powerful enough in
principle to explicate any process whatever, including psychological ones,
without suffering the limitations of any particular data language,
methodology, or analogy, and thus offers the opportunity of the construction
of psychological explications that can mediate among all such languages,
methodologies, and analogies--in particular, among the basic physiological,
phenomenological, and behavioral epistemological perspectives in psychology.
That is, cybernetic languages can in principle explicate any physical
process, including neurophysiological ones, can manifest any possible
input-output relationship, including behavioral ones, and offer their own
epistemological principles as explications of phenomenological ones: a
single model, then, could integrate all three fundamental approaches to
psychology. Furthermore, formal considerations can introduce constrains
that are independent of any particular empirical observations and that can

----

[1]Including ultimately to personality and social psychological
processes and their pathologies.

yield their own structurings, explanations, and predictions of empirical results,[1] as when a formal analysis of knowing yields a hierarchical constraint on what can be known.

Mathematics is more than a language of measurement, it is a language of structure and process as well, and it seems advisable to make maximal use of its potential. Formal systems languages offer a rigorous conceptual framework for the analysis of control and information processes, and, to the extent that these encompass psychological processes, offer the possibility of an integrated systems psychology. That possibility remains largely unexplored.

----
[1]Philosophical issues are relevant to all methodological perspectives. They are particularly relevant to the above discussion because it rests on several positions and assumptions that are different from those normally submerged in the literature of psychology. I will not address these issues directly at this time, but discussions of some of them--not necessarily discussions that I agree with--can be found, for example, in Fodor, 1968, Kuhn, 1970, and Turner, 1967, 1971.

BIBLIOGRAPHY

Adams, P., ed. Language in Thinking. Baltimore, Md.: Penguin, 1972.

Anderson, A. R., ed. Minds and Machines. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1964.

Arbib, M. A. Brains, Machines and Mathematics. New York: McGraw-Hill, 1964.

Arbib, M. A. Theories of Abstract Automata. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1969.

Arbib, M. A. The Metaphorical Brain. New York: John Wiley & Son, 1972.

Arnold, M. B., ed. The Nature of Emotion. Baltimore, Md.: Penguin, 1968.

Ashby, W. R. Design for a Brain. London: Chapman and Hall, 1960.

Ashby, W. R. An Introduction to Cybernetics. New York: John Wiley & Son, 1963.

Austin, J. L. How To Do Things with Words. New York: Oxford University Press, 1962.

Ausubel, D. P., and Sullivan, E. V. Theory and Problems of Child Development. New York: Grune & Stratton, 1957, 1970.

Ayer, A. J. Language Truth and Logic. New York: Dover, 1936.

Bach, E., and Harms, R. T., eds. Universals in Linguistic Theory. New York: Holt, Rinehart and Winston, 1968.

Baldwin, A. L. Theories of Child Development. New York: John Wiley & Son, 1967.

Bartlett, F. Thinking. New York: Basic, 1958.

Bartlett, F. C. Remembering. London:Cambridge UniversityPress, 1967.

Benecerraf, P., and Putnam, H., eds. Philosophy of Mathematics. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1964

Beth, E. W. The Foundations of Mathematics. Amsterdam: North-Holland, 1959, 1968.

Beth, E. W., and Piaget, J. Mathematical Epistemology and Psychology, Dordrecht, Holland: Reidel, 1966.

Bindra, D., and Stewart, J., eds. Motivation. Baltimore, Md.: Penguin, 1966.

Bochenski, I. M. A History of Formal Logic. Notre Dame, Ind.: University of Notre Dame Press, 1956, 1961, 1970.

Bowlby, J. Attachment. New York: Basic, 1969.

Bongard, N. Pattern Recognition. New York: Spartan, 1970.

Braffort, P., and Hirschberg, D., eds. Computer Programming and Formal Systems. Amsterdam: North-Holland, 1970.

Bruner, J. S.; Goodnow, J. J.; and Austin, G. A. A Study of Thinking. New York: John Wiley & Son, 1956.

Bruner, J. S.; Olver, R. R.; and Greenfield, P. M.; et al. Studies in Cognitive Growth. New York: John Wiley & Son, 1966.

Buckley, W., ed. Modem Systems Research for the Behavioral Scientist. Chicago: Aldine Press,1968.

Bunge, M. Causality. New York: Meridean, 1959, 1963.

Chisholm, R. M. Theory of Knowledge. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1966.

Chomsky, N. Aspects of the Theory of Syntax. Cambridge, Mass.: M.I.T., 1965.

Cofer, C. N., and Appley, M. H. Motivation. New York: John Wiley & Son, 1964.

Corning, W. C., and Balaban, M. W., eds. The Mind. New York: John Wiley & Son, 1968.

Craik, K. The Nature of Explanation. London: Cambridge University Press, 1943.

Crossley, J. N., ed. Sets, Models and Recursion Theory. Amsterdam: North-Holland, 1967.

Crossley, J. N., and Dummett, M. A. E., eds. Formal Systems and Recursive Functions. Amsterdam: North-Holland, 1965.

Cunningham, M. Intelligence: Its Organization and Development. New York: Academic Press, Inc., Publishers, 1972.

Curry, H. B. Outlines of a Formalist Philosophy of Mathematics. Amsterdam: North-Holland, 1951, 1970.

Darwin, Charles. The Origin of Species. New York: Collier, 1859, 1962.

Davis, M. Computability and Unsolvability. New York: McGraw-Hill, 1958.

Deutsch, J. A. The Structural Basis of Behavior. Chicago: University of Chicago Press, 1960.

Deutsch, S. Models of the Nervous System. New York: John Wiley & Son, 1967.

Dobzhansky, T. Genetics of the Evolutionary Process. New York: Columbia University Press, 1970.

Eilenberg, S., and Elgot, C. C. Recursiveness. New York: Academic Press, Inc., Publishers, 1970.

Elkind, D., and Flavell, J. H., eds. Studies in Cognitive Development. New York: Oxford University Press, 1969.

Erikson, E. H. Childhood and Society. New York: Norton, 1950, 1963.

Feigenbaum, E. A., and Feldman, J., eds. Computers and Thought. New York: McGraw-Hill, 1963.

Feigl, H. The "Mental" and the "Physical." Minneapolis: University of Minnesota Press, 1958, 1967.

Feigl, H., and Brodbeck, M., eds. Readings in the Philosophy of Science. New York: Appleton-Century-Crofts, 1953.

Feigl, H., and Maxwell, G., eds. Current Issues in the Philosophy of Science. New York: Holt, Rinehart and Winston, 1961.

Fillmore, C. J., and Langendoen, D. T., eds. Studies in Linguistic Semantics. New York: Holt,Rinehart and Winston, 1971.

Flavell, J. H. The Developmental Psychology of Jean Piaget. Princeton, N.J.: Van Nostrand, 1963.

Fodor, J. A. Psychological Explanation. New York: Random House, 1968.

Fodor, J. A., and Katz, J. J., eds. The Structure of Language. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1964.

Furth, H. G. Piaget and Knowledge. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1969.

Gamer, W. R. Uncertainty and Structure as Psychological Concepts.
     New York: John Wiley & Son, 1962.

Ginzburg, A. Algebraic Theory of Automata. New York: Academic Press,
     Inc., Publishers, 1968.

Glushkov, V. M. Introduction to Cybernetics. New York: Academic Press,
     Inc., Publishers, 1966.

Grossman, S. P. A Textbook of Physiological Psychology. New York:
     John Wiley & Son, 1967.

Halle, M., and Stevens, K. N. "Speech Recognition: A Model and a Program
     for Research." The Structure of Language. Edited by J. A. Fodor
     and J. J. Katz. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1964.

Harper, R. J. C., et al., eds. The Cognitive Processes. Englewood
     Cliffs, N.J.: Prentice-Hall, Inc., 1964.

Harrah, D. Communication: A Logical Model. Cambridge, Mass.: M.I.T.,
     1963.

Hartmanis, J., and Steams, R. E. Algebraic Structure Theory of Sequential
     Machines. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1966.

Hayes, J. R., ed. Cognition and the Development of Language. New York:
     John Wiley & Son, 1970.

Hebb, D. 0. Organization of Behavior. New York: John Wiley & Son, 1949.

Hempel, C. G. Philosophy of Natural Science. Englewood Cliffs, N.J.:
     Prentice-Hall, Inc., 1966.

Herstein, I. N. Topics in Algebra. New York: Blaisdell, 1964.

Hilgard, E. R., and Bower, G. H. Theories of Learning. New York:
     Appleton-Century-Crofts, 1966.

Hook, S., ed. Dimensions of Mind. New York: Collier, 1960.

Hopcroft, J. E., and Ullman, J. D. Formal Languages and Their Relation to
     Automata. Reading, Mass.: Addison-Wesley, 1969.

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

Iwanaga, M. "Development of Interpersonal Play Structure in Three, Four,
     and Five Year-Old Children." Journal of Research and Development
     in Education, VI, No. 3 (1973), 71-82.

Jakobovits, L. A., and Miron, M. S., eds. Readings in the Psychology of
     Language. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1967.

John, E. R. Mechanisms of Memory. New York: Academic Press, Inc.,
     Publishers, 1967.

Kessen, W., and Kuhlman, C., eds. Thought in the Young Child. Chicago:
     University of Chicago Press, 1962.

Khinchin, A. I. Mathematical Foundations of Information Theory. New York:
     Dover, 1957.

Klir, G. J. An Approach to General Systems Theory. New York:
     Van Nostrand, 1969.

Kuhn, T. S. The Structure of Scientific Revolutions. Chicago: University
     of Chicago Press, 1962, 1970.

Lauer, Q. Phenomenology. New York: Harper and Row, 1958.

Lenneberg, E. H. Biological Foundations of Language. New York: John Wiley
     & Son, 1967.

MacKay, D. "Mindlike Behavior in Artefacts." The Modeling of Mind.
     Edited by K. M. Sayre and F. J. Crosson. New York: Simon and
     Schuster, 1963.

MacKay, D. M. Information, Mechanism and Meaning. Cambridge, Mass.:
     M.I.T., 1969.

MacLane, S., and Birkhoff, G. Algebra. New York: MacMillan, 1967.

Maier, H. W. Three Theories of Child Development. New York: Harper and
     Row, 1965.

Marler, P., and Hamilton, W. J. Mechanisms of Animal Behavior. New York:
     John Wiley & Son, 1966.

Marvin, R. "Attachment, Exploratory and Communicative Behavior of Two,
     Three and Four Year Old Children." Unpublished Ph.D. dissertation,
     University of Chicago, 1972.

Mayr, E. Populations, Species, and Evolution. Cambridge, Mass.: Harvard
     University Press, 1970.

McCleary, R. A., and Moore, R. Y. Subcortical Mechanisms of Behavior.
     New York: Basic, 1965.

Mead, G. H. On Social Psychology. Chicago: University of Chicago Press,
     1934, 1964.

Melton, A. W., ed. Categories of Human Learning. New York: Academic
     Press, Inc., Publishers, 1964.

Merleau-Ponty, M. The Structure of Behavior. Boston, Mass.: Beacon Press, 1963

Mesarovic, M.D., ed. Views on General Systems Theory. New York:
      John Wiley & Son, 1964.

Miller, G. A.; Galanter, E.; and Pribram, K. H. Plans and the Structure
      of Behavior. New York: Holt, Rinehart, and Winston, 1960.

Minsky, M. Computation. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1967.

Minsky, M., and Papert, S. Perceptions. Cambridge, Mass.: M.I.T., 1969.

Mischel, T., ed. Cognitive Development and Epistemology. New York:
      Academic Press, Inc., Publisher, 1971.

Mussen, P. H., ed. Carmichaels Manual of Child Psychology. Vol. I.
      New York: John Wiley & Son, 1970.

Nagel, E. The Structure of Science. New York: Harcourt, Brace and World,
      1961.

Nagel, E., and Newman, J. R. Gödel's Proof. New York: New York University
      Press, 1958.

Neisser, V. Cognitive Psychology. New York: Appleton-Century-Crofts, 1967.

Newell, A., and Simon, H. A. Human Problem Solving. Englewood Cliffs, N.J.:
      Prentice-Hall, Inc., 1972.

Nilsson, N. J. Learning Machines. New York: McGraw-Hill, 1965.

Olson, D. R. Cognitive Development. New York: Academic Press, Inc.,
      Publisher, 1970.

Piaget, J. Play, Dreams and Imitation in Childhood. New York: Norton, 1962.

Piaget, J. Six Psychological Studies. New York: Random House, 1967.

Piaget, J. On the Development of Memory and Identity. Barre, Mass.: Barre
      Publishers, 1968.

Piaget,J. Genetic Epistemology. New York: Columbia University Press, 1970.

Piaget,J. Structuralism. New York: Basic, 1970.

Piaget,J. Psychology and Epistemology. New York: Grossman Publishers, 1971.

Piaget,J. Biology and Knowledge. Chicago: University of Chicago Press, 1971.

Piaget, J. Insights and Illusions of Philosophy. New York: World, 1965, 1971.

Piaget, J., and Inhelder, B. The Psychology of the Child. New York: Basic, 1969.

Piaget, J., and Inhelder, B. Mental Imagery in the Child. New York: Basic, 1971.

Pitcher, G. The Philosophy of Wittgenstein. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1964.

Powell, P. "Cognitive and Role-Taking Development in Mental Retardates and Normals." Unpublished Masters Thesis, University of Chicago, 1970.

Powell, P. "Higher-Level Cognitive and Social Competence Stages: An Extension and Integration of Piaget's and Mead-Turner's Theories." Unpublished Ph.D. dissertation, University of Chicago, 1971.

Pribram, K. H. Languages of the Brain. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1971.

Quine, W. V. O. Mathematical Logic. New York: Harper and Row, 1940, 1951.

Quine, W. V. O. Word and Object. Cambridge, Mass.: M.I.T., 1960.

Quine, W. V. O. Set Theory and Its Logic. Cambridge, Mass.: Harvard University Press, 1963.

Quine, W. V. O. Philosophy of Logic. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1970.

Reitman, W. R. Cognition and Thought. New York: John Wiley & Son, 1965.

Rogers, H., Jr. Theory of Recursive Functions and Effective Computability. New York: McGraw-Hill, 1967.

Rosenberg, J. F., and Travis, C., eds. Readings in the Philosophy of Language. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1971.

Rosenfeld, A. Algebraic Structures. San Francisco: Holden Day, 1968.

Roth, M. D., and Galis, L., eds. Knowing. New York: Random House, 1970.

Royce, J. R., and Rozeboom, W. W., eds. The Psychology of Knowing. New York: Gordon and Breach, 1972.

Rudner, R. S. Philosophy of Social Science. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1966.

Russel, Bertrand. Human Knowledge. New York: Simon and Schuster, 1948.

Russel, R. W., ed. Frontiers in Physiological Psychology. New York:
     Academic Press, Inc., Publisher, 1966.

Sayre, K. M., and Crosson, F. J., eds. The Modeling of Mind. New York:
     Simon and Schuster, 1963.

Scher, J., ed. Theories of the Mind. New York: Free Press of Glencoe, 1962.

Seligman, M., and Hager, J. L., eds. Biological Boundaries of Learning.
     New York: Appleton-Century-Crofts, 1972.

Shannon, C. E., and Weaver, W. The Mathematical Theory of Communication.
     Urbana, Ill.: University of Illinois Press, 1964.

Simon, H. A. The Sciences of the Artificial. Cambridge, Mass.: M.I.T.,
     1969.

Simon, H. A., and Siklossy, L. Representation and Meaning.
     Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1972.

Simpson, G. G. The Meaning of Evolution. New Haven, Conn.: Yale
     University Press, 1949, 1967.

Simpson, G. G. The Major Features of Evolution. New York: Columbia
     University Press, 1953.

Singh, J. Information Theory, Language and Cybernetics. New York:
     Dover, 1966.

Slagle, J. R. Artificial Intelligence. New York: McGraw-Hill, 1971.

Sloman, A. "Functions and Rogators." Formal Systems and Recursive Functions.
     Edited by J. N. Crossley and M. A. E. Dummett. Amsterdam:
     North-Holland, 1965.

Smith, F., and Miller, G. A., eds. The Genesis of Language. Cambridge,
     Mass.: M.I.T., 1966.

Steinberg, D. D., and Jakobovits, L. A., eds. Semantics. London:
     Cambridge University Press, 1971.

Stellar, E., and Sprague, J. M., eds. Progress in Physiological Psychology.
     New York: Academic Press, Inc., Publisher, 1966.

Taylor, J. G. The Behavioral Basis of Perception. New Haven, Conn.:
     Yale University Press, 1962.

Terwilliger, R. F. Meaning and Mind. New York: OxfordUniversity Press, 1968.

Turner, M. B. Philosophy and the Science of Behavior. New York:
    Appleton-Century-Crofts, 1967.

Turner, M. B. Realism and the Explanation of Behavior. New York:
    Appleton-Century-Crofts, 1971.

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

Voss, J. F., ed. Approaches to Thought. Columbus, Ohio: Merrill, 1969.

Vygotsky, L. S. Thought and Language. Cambridge, Mass.: M.I.T., 1962.

Warren, J. M., and Akert, K., eds. The Frontal Granular Cortex and Behavior.
    New York: McGraw-Hill, 1964.

Wann, T. W., ed. Behaviorism and Phenomenology. Chicago: University of
    Chicago Press, 1964.

Werner, H., and Kaplan, B. Symbol Formation. New York: John Wiley & Son,
    1963.

Wiener, N. Cybernetics. Cambridge, Mass.: M.I.T., 1948, 1961.

Winograd, T. Understanding Natural language. New York: Academic Press,
    Inc., Publisher, 1972.

Wittgenstein, Ludwig. Tractatus Logico-Philosophicus. New York: Humanities,
    1922, 1961, 1969.

Woodburne, W. S. The Neural Basis of Behavior. Columbus, Ohio: Merrill,
    1967.