A Category-Theoretic Approach to
Systems in a Fuzzy World.†

Michael A. Arbib
Computer and Information Science

Ernest G. Manes
Mathematics

COINS Technical Report 74B-3

University of Massachusetts
Amherst, MA 01002, U.S.A.

(November 20, 1974)

The last 30 years have seen the growth of a new branch of mathematics called CATEGORY THEORY which provides a general perspective on many different branches of mathematics. Many workers [1] have argued that it is category theory, rather than SET THEORY, that provides the proper setting for the study of the FOUNDATIONS OF MATHEMATICS.

The aim of this paper is to show that problems in APPLIED MATHEMATICS, too, may find their proper foundation in the language of category theory. We do this by introducing a number of concepts of SYSTEM THEORY which we unify in our theory of MACHINES IN A CATEGORY. We write as system theorists, not as philosophers. Our hope is to stimulate a dialogue with philosophers of science as to the proper role for category theory in a systematic analysis of a fuzzy world. We do not discuss applications to biology or psychology--the framework presented here is at a very high level of generality, and does not address the particularities which give these disciplines their distinctive flavor.

† The research reported in this paper was supported in part by the National Science Foundation under Grant No. GJ 35759.

This paper is divided into two parts. In Part I, we sketch how the subjects of control theory, computers and formal language have grown out of the ur-disciplines of MECHANICS and LOGIC; and then present the formal concepts of sequential machine, linear machine and tree automaton. We show how our notion of MACHINE IN A CATEGORY provides an uncluttered generalization of these three concepts.

In Part II, we introduce the "fuzzy world". Although the study of quantum mechanics provides the best known framework, we stay within system theory, showing how PROBABILITY, MECHANICS and LOGIC gave rise to the study of markov chains, structural stability and multi-valued logics. We then present the formal concepts of nondeterministic sequential machine, stochastic automaton and fuzzy-set automaton. Our notion of FUZZY MACHINE will generalize all three. Of particular interest will be the demonstration that, although fuzzy machines generalize machines in a category, we can--by a suitable enlargement of viewpoint--regard them as a special case.

The paper is self-contained both as to system theory and to category theory--but many topics must be but briefly outlined in an expository paper of this kind. The reader wishing a fuller introduction to category theory is referred to our book [2]; a text on control theory is [3]; for system theory see [4,5]; many other concepts of machine theory appear in [6]; our theory of machines in a category appeared in [4,6,7,8] while the technical details of fuzzy machines appear in [9]. The state of the art in applying category theory to systems and automata is reflected in [10].

1. MACHINES IN A CATEGORY

In Figure 1, we schematize the evolution of Machines in a Category from concepts in generalized mechanics and formal logic through the study of control theory, the impact of computers, and notions of formal linguistics. The paragraphs below are lettered with the arrows they describe:

A: Building on the work of Newton and its refinement by such workers as Legendre, Hamilton, in the middle of the 19th Century, gave the following formulation of generalized mechanics: The vector of generalized positions, $q = (q_1,\ldots,q_n)$, one for each degree of freedom of the system must be augmented by $p = (p_1,\ldots,p_n)$, the vector of generalized momenta, one for each degree of freedom of the system. There is then a function $H(p,q)$, the Hamiltonian, of these variables, in terms of which we may express the system dynamics:

$$\begin{cases} \dot{q}_j = \dfrac{\partial H}{\partial p_j} & \text{for } 1 \le j \le n \\ \dot{p}_j = -\dfrac{\partial H}{\partial q_j} & \text{for } 1 \le j \le n. \end{cases}$$

Thus, with Hamilton we see very vividly that we may study systems which are described by the evolution of state vectors over time, with this evolution governed by vector differential equations of the form

$$\dot{q} = f(q)$$

where now the state q includes position, momentum, and any other relevant variables as components.

The transition to control theory comes when we emphasize that the differential equation describing the evolution of the state of a system contains a number of parameters, representing forces, which can be manipulated from outside the system, so that we may write down the change of state as a function

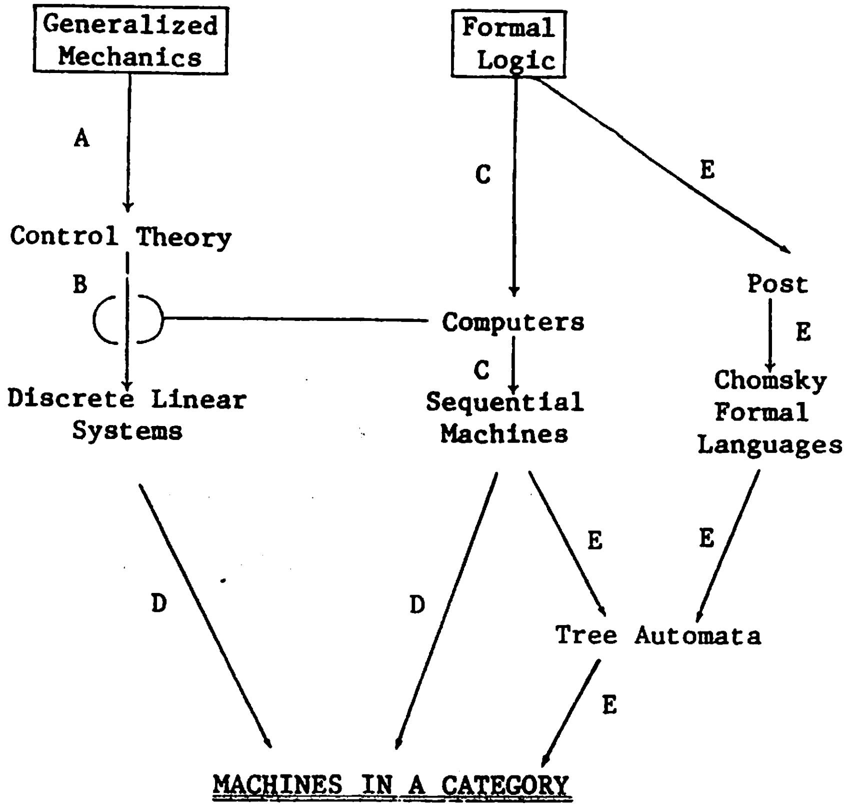

Generalized Mechanics
Formal Logic
A
C
E
Control Theory
B
Post
Computers
E
C
Discrete Linear Systems
Sequential Machines
Chomsky Formal Languages
D
D
E
E
Tree Automata
E
MACHINES IN A CATEGORY

Figure 1.

$$\dot{q} = f(q,x) \quad (1)$$

not only of the state vector itself, as in the classical formulation, but also as a function of a control vector x. We should also note that only certain aspects of the state will actually be measurable at any time, so that we may introduce an output vector y which is a function

$$y = \beta(q) \quad (2)$$

of the instantaneous state. For example, in classical mechanical systems, we can observe only the positions 'instantaneously', while the momenta--or the related velocity variables--must be built up from observation of changes in position over some period of time.

We now turn (Box 1) to three mathematical problems of control theory, which underpin the central problem of optimization.

Given a system described by a pair of equations giving (1) the rate of change of the state and (2) the observable output as a function of the state, we are to find a control signal which will drive the system from some initial state to a desired final state in the quickest possible way, or with the least use of energy--as, for example, of firing the rockets of a satellite in such a way as to bring it into a desired stable configuration. Clearly, however, before we analyze what is the most efficient way to bring it into position, we must know whether any suitable control exists at all, and this is the question of reachability. [Incidentally, it is worth noting that optimal control is closely based on the work of Hamilton, for Hamilton had observed that the trajectory of a system following given laws of motion was such as to minimize the value of a certain function. It is a natural transition, then, to apply these techniques to seek an input--or control--trajectory which will minimize some evaluation of the cost or time of system performance, and this approach

Three Problems of Control Theory

Given a system $\begin{cases} \dot{q} = f(q,x) \\ y = \beta(q) \end{cases}$

we may ask:

Is it reachable? Can we control it in such a way as to drive it from some initial state to any desired final state?

Is it observable? Given the system in an unknown state, can we conduct experiments upon it (apply controls, measure outputs) in such a way as to eventually determine the system's state.

Given a system whose equations are unknown, the realization problem is to determine a set of states, a dynamics f, and an output function β which correctly describe the observed input-output behavior of the system.

Box 1

is the basis of Pontryagin's maximum principle, one of the fundamental techniques of optimal control.]

If reachability is an important question in the design of feedback control systems--given a state, does there exist a control we can apply to move the system from that state to some other, desired, state--then no less important a question must be the one of observability. We have already commented that the instantaneous output of the system will in general tell us only some small portion of what we need to know about its state. But feedback control usually requires that we know all of the state before we can determine what is the proper input to apply. Thus, it is our concern to determine when a system is observable: namely, we wish to know how, given the system in an unknown state, we can conduct experiments upon it--namely by applying controls and measuring the consequent outputs--in such a way as to eventually determine the system's current state. Thereafter, our knowledge of the dynamics will allow us to update the state as we apply the appropriate controls to its behavior.

The above prescription is based upon our knowing the equations (1,2) which govern the system. This of course raises the very realistic problem of how we might find these equations in the first place. In general, if we come upon a system to which we can apply certain inputs, and for which we can observe certain outputs, we wish to determine a state-space which can mediate the relationship between the inputs and the outputs, and we then wish to determine the dynamics and the output function which correctly describe the observed input/output behavior of the given system. This is the realization problem, and we are frequently concerned to find a realization which is minimal in the sense of having the smallest state-space possible. One of the most pleasing general results of control theory is that if a realization is indeed minimal, then it must be both reachable and observable.

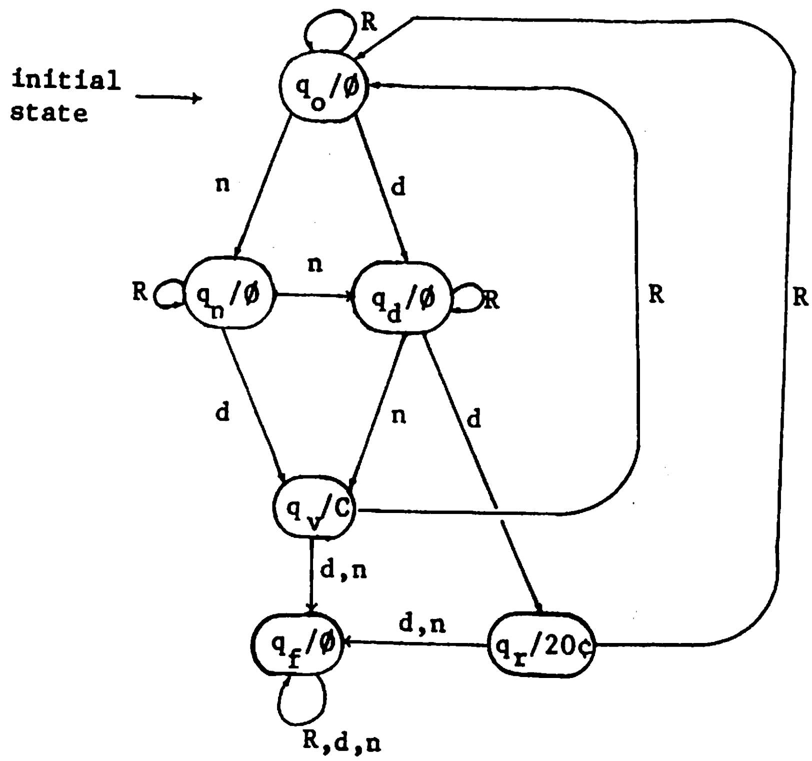
B: However, the treatment of arbitrary systems described by differential equations is too complex for efficient mathematical solution. One of the most common ways of approximating a complex system is by using linear equations. Moreover, the advent of the computer as the tool par excellence for controlling a system has led us to move from continuous time systems described by differential equations to discrete time systems in which we sample the behavior of the system, and apply inputs, at regular intervals, so that we describe the system in terms of equations which show how it changes from one sampling period to the next. In fact by using an approximation to the rate of change predicted by the derivative, and by using Taylor series, we can come up with a linear approximation to the change in state of the system over the sampling period which is linear, and we may also approximate the output by a linear function of the state:

$$
\begin{aligned}
q(t+\Delta t) &\doteq q(t) + f(q(t),x(t))\Delta t \\
&\doteq q(t) + \frac{\partial f}{\partial q}\cdot q(t)\Delta t + \frac{\partial f}{\partial x}\cdot x(t)\Delta t \\
&= F\,q(t) + G\,x(t) \qquad (3)\\
&\quad \text{where } F = \left[I + \Delta t\,\frac{\partial f}{\partial q}\right];\quad G = \Delta t\,\frac{\partial f}{\partial x}.
\end{aligned}
$$

$$
\begin{aligned}
y(t) &\doteq \frac{\partial \beta}{\partial q}\cdot q(t) \\
&= H\,q(t) \qquad (4)\\
&\quad \text{where } H = \frac{\partial \beta}{\partial q}.
\end{aligned}
$$

It is an empirical fact that many control systems can be usefully approximated by descriptions of the form (3)/(4) using constant matrices F ,G and H.

C: If computers encouraged the passage from general differential equations to discrete linear systems--or linear machines as we will call them from now on--they also gave rise to new discrete systems in their own right, which in no way



$\delta: Q \times X_o \longrightarrow Q$

$\beta: Q \longrightarrow Y$

Input Set $X_o = \{n,d,R\}$

State Set $Q = \{q_o, q_n, q_d, q_v, q_r, q_f\}$

Output Set $Y = \{\emptyset, C, 20¢\}$

Figure 2.

The 15¢ Machine.

were to be considered as approximations to continuous systems. The concepts of truth values in a two-valued logic which could be computed upon in a numerical-like but non-numerical way, due to George Boole, provided the proper framework in the 1930's and 1940's for the development of a formal theory both of relay switching networks and the McCulloch-Pitts theory of formal networks. These led to the general theory of sequential machines, which--among other things--provided the proper formal framework for talking about the various subsystems of a computer. For example (Figure 2) we can describe a vending machine which accepts nickels, dimes and rests--the set of inputs is $X_o = \{n,d,R\}$. It vends a candy bar, C, when 15¢ has been received from the initial state, puts out 20¢ if it has received either 2 dimes or 2 nickels and a dime starting from the initial state, and otherwise emits nothing, $\emptyset$--so that the output set is $Y = \{\emptyset, C, 20¢\}$. The current state and current input determine the next state via a function δ--an arrow leads from node q via arrow x to node $\delta(q,x)$. The current output is a function β of the current state--we mark the node for state q with the notation $q/\beta(q)$.

D: The point to stress here is that the various input, state, and output sets involved here are small finite sets, and are in no way the Euclidean spaces of linear system theory. In fact (Box 2) we may see that the theory of sequential machines and the theory of linear machines live in quite different domains of discourse:

First, let us examine sequential machines. It is common to assign to each machine an initial state--in this case we have represented that initial state by the map τ from the one-element set 1 to the state set Q whose image is precisely the initial state q_o. The dynamics $\delta: Q \times X_o \rightarrow Q$ is then a map which assigns to each state and each input of the sequential machine the

Formal Definitions

Sequential Machines

Initial State

$\tau: 1 \longrightarrow Q$

Dynamics

$\delta: Q \times X_o \longrightarrow Q$

Output Map

$\beta: Q \longrightarrow Y$

This lives in the category Set:

each object is a set; each morphism (arrow) is a map.

Linear Machines

Input Map

$G: I \longrightarrow Q$

Zero-Input Dynamics

$F: Q \longrightarrow Q$

Output Map

$H: Q \longrightarrow Y$

This lives in the category Vect:

each object is a vector space; each morphism (arrow) is a linear map.

Box 2.

state into which it will next settle, whereas the output map $\beta: Q \longrightarrow Y$ assigns to the current state the corresponding output. We stress that sequential machines live in the category Set--a domain of mathematical discourse comprising sets and arbitrary maps between those sets.

In describing a linear machine, we give an input map $G: I \longrightarrow Q$, a zero input dynamics which is simply the map F from the state set Q into itself, and an output map $H: Q \longrightarrow Y$. These describe the behavior of the machine via $q(t + \Delta t) = Fq(t) + Gx(t); \; y(t) = Hq(t)$. The appropriate domain of discourse here is the category Vect in which now the objects are vector spaces and each morphism--i.e., arrow going from one object to another--is a linear map. [We have lined up elements of the definitions of sequential machines and linear machines in Box 2 in a way that will seem mysterious to the reader. We hope that the reason will become clear by inference from our general definition of machines in a category in Box 4 below.]

Clearly, at this stage it is proper that we admit that the notion of a CATEGORY or mathematical domain of discourse implicit in our above comparison is in fact a formal concept of mathematics. In fact, we have as the basic notions of category theory the idea of a category and of a functor (Box 3).

A category $\mathcal{K}$ is a domain of mathematical discourse in which we have a collection of objects, such as the arbitrary sets of Set or the vector spaces of Vect, together with, for each pair A,B of objects, a collection $\mathcal{K}(A,B)$ of morphisms from the first to the second--these correspond to the arbitrary maps of one set into another of Set, or the linear maps from one vector space into another of Vect. As in both of these examples, we may compose morphisms so long as the first ends where the second begins--and the composition is associative, i.e., we may string together an arbitrary number of composable maps and know that the overall composition is uniquely defined,

Basic Notions of Category Theory

A Category $\mathcal{K}$ is a domain of mathematical discourse comprising

a collection of objects

for each pair A,B of objects a collection $\mathcal{K}(A,B)$

of morphisms

$$f: A \longrightarrow B \quad \text{or} \quad A \xrightarrow{f} B$$

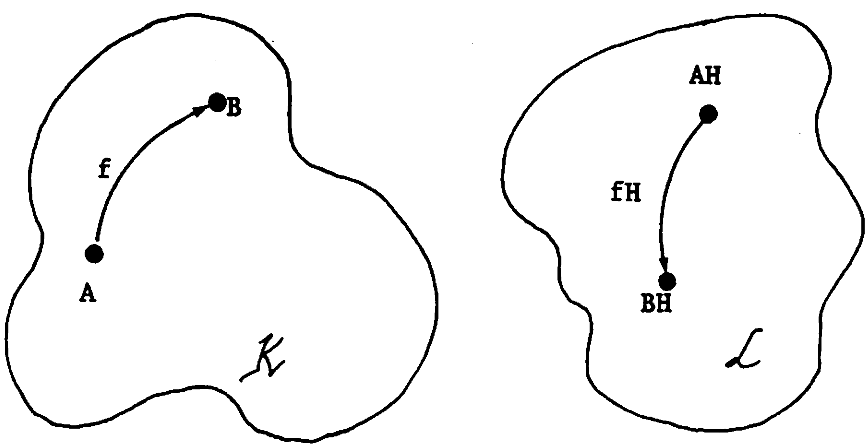
with domain A and codomain B

together with a law of composition

$$g \cdot f: A \longrightarrow C = A \xrightarrow{f} B \xrightarrow{g} C$$

which is associative and has identities $id_A: A \longrightarrow A$.

A functor H from category $\mathcal{K}$ to category $\mathcal{L}$



sends

objects A

morphisms $f: A \longrightarrow B$

in $\mathcal{K}$

to

objects AH

morphisms $fH: AH \longrightarrow BH$

in $\mathcal{L}$

in a "nice" way, namely

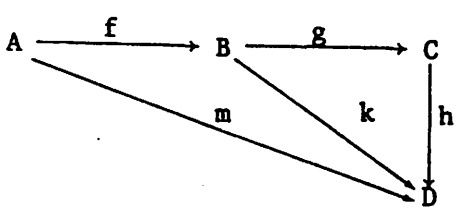
If $f = id_A: A \longrightarrow A$ then $fH = id_{AH}: AH \longrightarrow AH$

If $f = A \xrightarrow{g} B \xrightarrow{h} C$ then $fH = AH \xrightarrow{gH} BH \xrightarrow{hH} CH$.

Box 3.

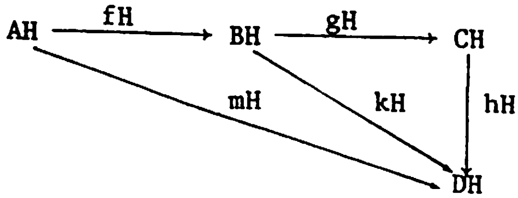
irrespective of the 'bracketing' of the constituent morphisms. Moreover, we may associate with each object an identity morphism--this corresponds to the map which sends each element to itself in Set and Vect--which has the property that if we compose it with any other morphism, the result is that other morphism. Incidentally, this equivalent definition of the identity map exemplifies the difference between the set theory (define everything in terms of elements) and the category theory (define everything in terms of morphisms) approach to the foundations of mathematics.

So far, so good. A somewhat more technical concept basic to any use of the language of category theory is that of a functor. Briefly put, a functor is simply a passage from one category to another in such a way that the identities, and the composition of morphisms, are respected. In particular, a very useful idea in category theory has been that of 'chasing commutative diagrams'--drawing graphs in which morphisms take us from one object to another over diverse paths in such a way that the overall composition is the same. E.g., to say that



(5)

commutes, is to say that $k \cdot f = m$, $h \cdot g = k$ and $h \cdot g \cdot f = k \cdot f = m$. The iterated application of the fact that a functor preserves identities and composition allows us to easily deduce that it must also preserve the commutativity of any diagram--i.e., that if we replace each object A by the object AH and if we replace each morphism f by the morphism fH, then if different paths from one initiation point to one termination have the same composition in the original diagram, then they must have equal compositions in the transformed diagram. For example, if (5) commutes in $\mathcal{K}$ then



commutes in $\mathcal{L}$--e.g., $mH = (k \cdot f)H = kH \cdot fH$.

With these concepts before us we can now present the key concept of machines in a category $\mathcal{K}$. We should not, as we were encouraged to do in the theory of sequential machines, think of the input of a machine as being a set--or, more generally, an object--of inputs. Rather, we should think of the input as being a process which transforms the state object Q into a new state object QX. In all cases, we are to think of X as being a functor from the given category $\mathcal{K}$ to itself. Then, given this object QX upon which the dynamics is to act, a dynamics is simply a $\mathcal{K}$-morphism $\delta: QX \longrightarrow Q$.

Returning to Box 2, we see that for sequential machines, the category $\mathcal{K}$ is Set, and the functor X transforms a state set Q into the cartesian product $Q \times X_o$ of all state-input pairs; while in the case of linear machines we work in the category $\mathcal{K}$ = Vect, and our functor X leaves things unchanged so that $QX = Q$. [To see that these really are functors, we must show how they act on morphisms. For $f: Q \longrightarrow Q'$ in Set and $X = -\times X_o$: Set $\longrightarrow$ Set, we define $fX: Q \times X_o \longrightarrow Q' \times X_o$ to send (q,x) to $(f(q),x)$. For $f: Q \longrightarrow Q'$ in Vect and X = identity : Vect $\longrightarrow$ Vect, we define $fX: Q \longrightarrow Q'$ to be simply f. The reader may check the functor conditions of Box 3.] Then, a sequential machine has dynamics $\delta: Q \times X_o \longrightarrow Q$, while linear machines have dynamics $F: Q \longrightarrow Q$. With this we see that both types of machine of Box 2 are subsumed in our general notion of MACHINE IN A CATEGORY, summarized in Box 4. Summarizing, we see that a machine in a category requires us to specify

MACHINES IN A CATEGORY

X-Machines

$$\left\{\begin{array}{l} \tau: I \longrightarrow Q \\ \delta: QX \longrightarrow Q \\ \beta: Q \longrightarrow Y \end{array}\right.$$

$X: \mathcal{K} \longrightarrow \mathcal{K}$ is a functor; τ, δ and β are $\mathcal{K}$-morphisms

We stress that input is a process which converts the state-object Q into a new object QX on which the dynamics can operate

Box 4.

a functor X from $\mathcal{K}$ into itself which is a process which converts the state object Q into a new object QX on which the dynamics δ can operate. We must specify a $\mathcal{K}$-morphism τ from I to Q--in the case of sequential machines this gives us the initial state, while it gives the input map of a linear machine. Finally, we give a morphism β from Q to Y--which provides an output map in both cases.

E: Instead of giving a formal treatment, let us just briefly note that tree automata do indeed fit into this general framework of machines in a category. Here, we briefly note that Post's theory of canonical systems was specialized by Chomsky to yield his formal theory of languages, and that many authors soon realized that the appropriate theory for handling the derivation trees of formal linguistics was the theory of tree automata, which could be seen as a straightforward generalization of the theory of sequential machines we have discussed above. Rather than give the general definition of tree automata, however, let us content ourselves with a simple example (Figure 3) of processing binary arithmetic trees. Here we start at the bottom--at the 'leaves' and combine pairs of numbers by addition and multiplication until finally at the 'root' of the tree we have the overall evaluation of the arithmetic expression represented by the tree. Let us see how we can think of this as a machine in a category in the sense of Box 4. Here we are to think of the state set as being the set $\underline{N}$ of all natural numbers, and we now introduce a functor X: Set ⟶ Set on the category of sets, which sends each state set Q to the union QX of two sets, one being $Q \times Q \times \{+\}$ and the other being $Q \times Q \times \{\times\}$. We then see that a map from QX to Q gives us precisely the two maps we need to evaluate nodes of

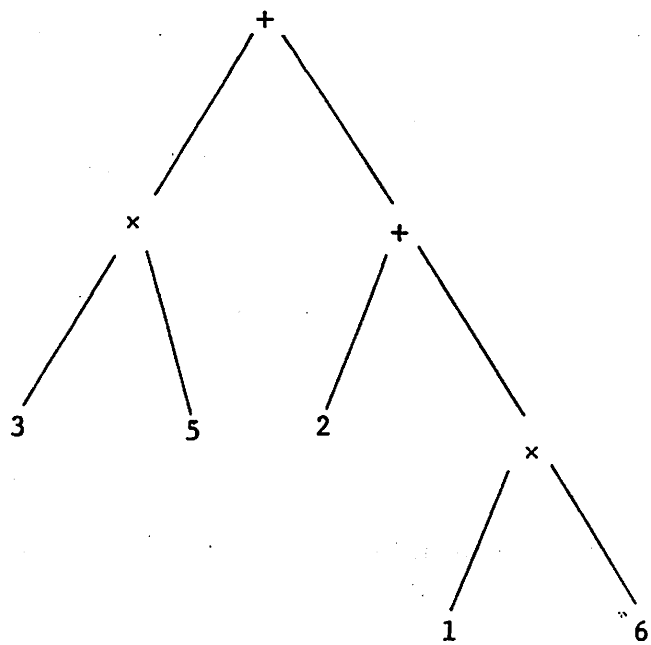


Figure 3.

Processing binary arithmetic trees.

State Set Q is $\underline{\underline{N}}$ (the natural numbers) in this example.

Introduce a functor X: $\underline{\underline{\text{Set}}} \longrightarrow \underline{\underline{\text{Set}}}$

$$QX = Q \times Q \times \{+\} \cup Q \times Q \times \{\times\}.$$

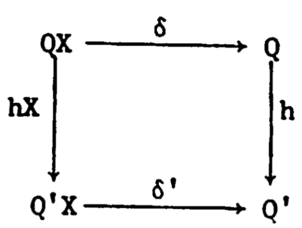
Then a map $\delta: QX \longrightarrow Q$ gives the dynamics:

$$\delta(q_1, q_2, +) = q_1 + q_2$$

$$\delta(q_1, q_2, \times) = q_1 \times q_2.$$

the tree as we pass from the leaves to the root.

With this successful subsumption of tree automata in a framework designed to embrace sequential machines and linear machines, we have almost completed the first part of the paper. But, before we look at what happens to this theory in a 'fuzzy world', it seems worthwhile to quickly summarize a number of results which have been obtained in the theory of machines in a category, even if we do not have space to spell out any of the details. In fact, given any functor X from the category $\mathcal{K}$ into itself we can define a category <u>Dyn</u>(X) of X-<u>dynamics</u>--the objects are precisely the X-dynamics, while a <u>Dyn</u>(X)-morphism--or a <u>dynamorphism</u>--is a $\mathcal{K}$-morphism of state objects which 'respects' the dynamics--we might either apply the dynamics and code the resulting state, or we may code QX and then apply the second dynamics--the result is the same, as expressed in the commutative diagram



<u>Dyn</u>(X) is a <u>category</u>
because X is a <u>functor</u>

This category is the setting for the major results of the theory of machines in a category which we have developed [We should also mention that other contributions to the theory of machines in a category--though not using exactly the same framework as that we have developed here--have been made by Goguen [11,12], Bainbridge [13], Ehrig et al. [14], Goguen et al. [15] and others. However, the nature of our survey does not make it appropriate to indicate here the ways in which these different contributions are interrelated.] The results which follow are presented far too briefly to allow comprehension--using as they do the technical category-theoretic concept of an adjoint of a functor.

However, the very point of this tantalizingly brief presentation is to stress how important adjoints are to system theory; and we hope that many readers will be tempted to turn to [2,4,6,7] for a full treatment of the following results.

We introduce a new functor $U: \underline{\underline{Dyn}}(X) \longrightarrow \mathcal{K}$ which sends an object (Q,δ) of $\underline{\underline{Dyn}}(X)$ to Q in $\mathcal{K}$, and sends a dynamorphism $h: (Q,\delta) \longrightarrow (Q',\delta')$ to the underlying $\mathcal{K}$-morphism $h: Q \longrightarrow Q'$. We call it the forgetful functor because it "forgets" the dynamics δ and just remembers the underlying state-object Q.

Category theorists give a central role to the notion of adjoint of a functor. In some circumstances we may associate to a functor $H: \mathcal{K} \longrightarrow \mathcal{L}$ another functor $F: \mathcal{L} \longrightarrow \mathcal{K}$ called the left adjoint of H. In other circumstances, there exists a functor $G: \mathcal{L} \longrightarrow \mathcal{K}$ called the right adjoint of H. The definition of adjoints is beyond the scope of this paper (see [2, Chapter 7] for the details), but we note the terminology that if H has left adjoint F and B is an object in $\mathcal{L}$, then we say that BF is the free $\mathcal{K}$-object over B; while if H has right adjoint G, we say that BG is the cofree $\mathcal{K}$-object over B. With this terminology we may summarize some of the results of [2,4,6,7]:

First, we showed that if the forgetful functor $U: \underline{\underline{Dyn}}(X) \longrightarrow \mathcal{K}$ from the category of X-dynamics to the underlying category $\mathcal{K}$ has a left adjoint $F: \mathcal{K} \longrightarrow \underline{\underline{Dyn}}(X)$ --so that we may talk of free dynamics QF in $\underline{\underline{Dyn}}(X)$-- then we can in fact construct a reachability theory and a theory of minimal realization. This theory includes sequential machines, linear machines, tree automata, and many other examples.

If on the other hand we require that the forgetful functor has a right adjoint G: $\mathcal{K} \longrightarrow$ Dyn(X) --so that we may construct a cofree dynamics QG in Dyn X--we are then able to construct an observability theory and a cominimal realization theory--which is much the same as a minimal realization theory, with differences that are too technical to detain us here. In any case, we find that tree automata do not correspond to functors X which yield forgetful functors with right adjoints, but sequential and linear machines do. Thus, both sequential and linear machines are examples of machines in a category for which the corresponding forgetful functor has both a left and a right adjoint, and we have found that in this case we get an exceptionally simple minimal realization theory using what are called image factorizations, and that we also have a framework for studying duality of systems based upon the fundamental concept of categorical duality [2]. In particular, of course, we may talk about both reachability and observability for such systems. To further tantalize the reader, we point out that, for I as in Box 4, IF is the "object of input experiments". Since IF is determined uniquely by X [2, p. 113], the nature of "input experiments" is not determined independently by intuition--a new principle in system theory. This principle has surprising consequences for affine machines [11] and group machines [8].

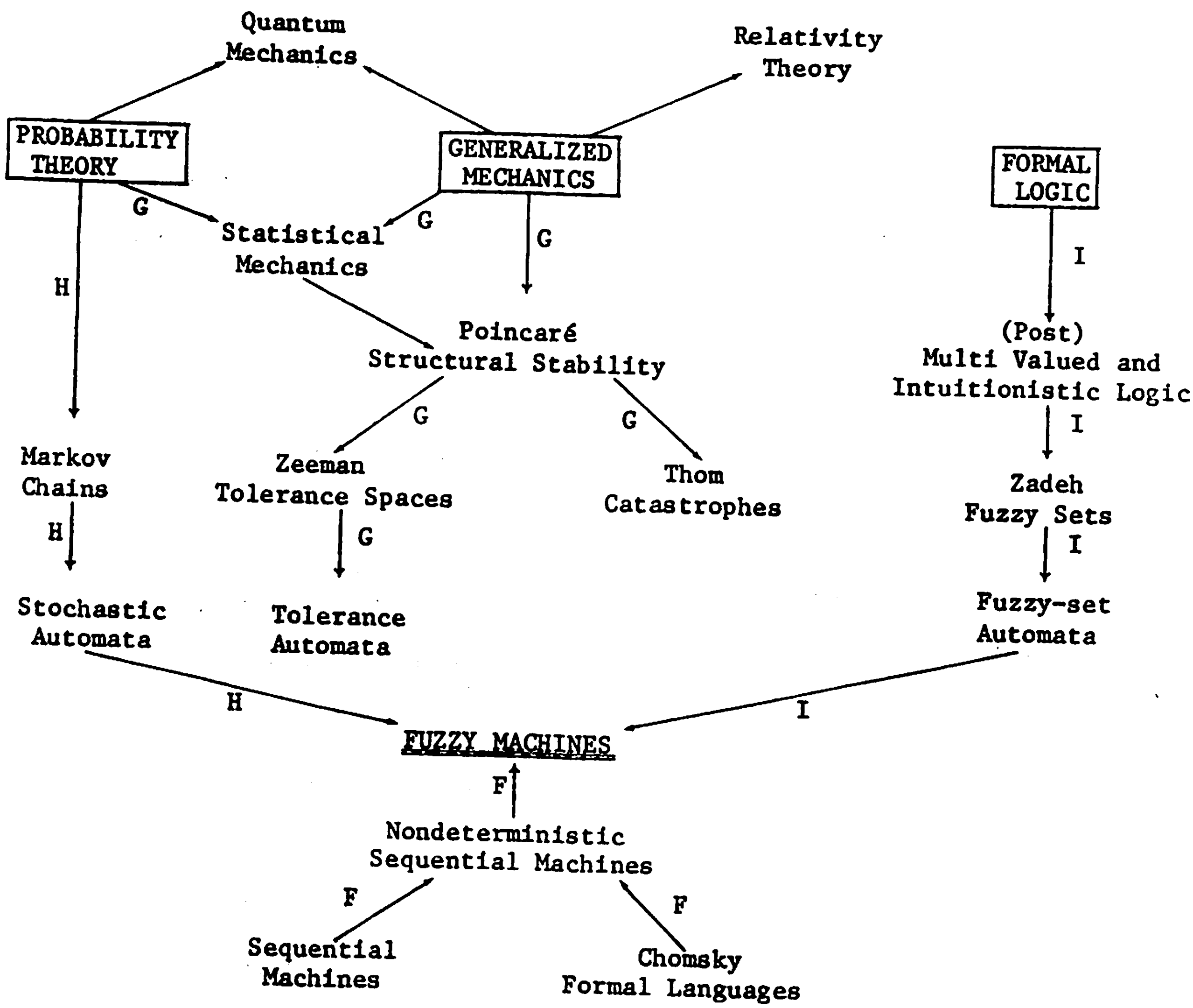
Summarizing, then, we have seen that with the idea of a functor we can embrace a far larger class of automata than we can by restricting ourselves to the situation in which the dynamics must act on something with the form of $Q \times X_0$; and--as the above flash-through of results indicates--the category theory concept of adjoints of functors is central to our approach to general system theory.

We reiterate that the above survey is far too brief, but it should be sufficient to set the stage for the new perspective that is required when we start looking at different aspects of nondeterminism in our approach to systems in a fuzzy world.

2. FUZZY MACHINES

We have now seen how to use category theory to provide a general perspective (Figure 1 to Box 4) for a number of apparently disparate classes of systems: sequential machines, linear machines, and tree automata. But the time has come to face up to the fact that we live in a 'fuzzy' world--there is no guarantee that we can be sure of the next state of a system in the real world. In the rest of this paper, we are going explore a somewhat paradoxical approach to the 'fuzziness', namely that in which one can give a precise prescription of the range of possibilities for the next state from any given starting condition. (But we emphasize at once that we will axiomatize a class of such prescriptions, frankly recognizing that there are many different kinds of fuzziness.)

The first way in which nondeterminism entered the world of automata theory was through the study of nondeterministic sequential machines (F of Figure 4). This was in part motivated by the study of formal languages--for in designing machines to parse a sentence one had to be aware of the fact that the initial portion of a sentence could be consistent with a number of possible parsings, so that there was no unique way to classify the next word, but rather a number of possible ways consistent with the information already processed. In any case, whatever the history, there has become entrenched the idea of a nondeterministic sequential machine--we suggest that perhaps a better word would be 'possibilistic'--in which the current state and the input do not determine a single unique next state, but rather determine a set of possible next states, so that the dynamics maps the set of (state, input) pairs into an element of 2^Q, the set of subsets of the state set Q.



Quantum Mechanics
Relativity Theory
PROBABILITY THEORY
GENERALIZED MECHANICS
FORMAL LOGIC
G
G
G
H
I
Statistical Mechanics
Poincaré Structural Stability
(Post) Multi Valued and Intuitionistic Logic
G
G
I
Markov Chains
Zeeman Tolerance Spaces
Thom Catastrophes
Zadeh Fuzzy Sets
H
G
I
Stochastic Automata
Tolerance Automata
Fuzzy-set Automata
H
I
FUZZY MACHINES
F
Nondeterministic Sequential Machines
F
F
Sequential Machines
Chomsky Formal Languages

Figure 4.

$$\delta: Q \times X_o \longrightarrow 2^Q$$

The idea, then, is that in any run of the machine, one and only one state will appear at any given time, but if state q appeared at time t and input x were then applied, the state at time $t + 1$ _must_ belong to the set $\delta(q,x)$ of states.

Now, we may observe that the passage from Q to 2^Q is the object map of a functor of the category _Set_ into itself,

$$2^{(-)}: \underline{Set} \longrightarrow \underline{Set} \quad \text{is a functor}$$

$$Q \longmapsto 2^Q$$

$$[f: Q \longrightarrow Q'] \longmapsto [2^f: 2^Q \longrightarrow 2^{Q'} : S \subset Q \longmapsto f(S) = \{f(s) \mid s \in S\} \subset Q'].$$

This suggests that the nondeterministic sequential machines we have just looked at may be considered to be a special case of dynamics expressed in the form

$$\delta: QX \longrightarrow QT$$

for some suitable choice of a functor T. The question before us, then, is what are suitable restrictions on functors T for the consideration of such dynamics to be in fact the proper setting for 'dynamics in a fuzzy world'?

G: Before we turn to this rather technical question, however, it is worth continuing the historical perspective of Figure 1 by considering, in Figure 4, various ways in which the idea of a 'fuzzy world' has been approached. Of course, this historical view of ours is a very sketchy one, and we can only hope that some more careful historian or philosopher of science will take this lead to more carefully chart the interconnections between these ideas. In any case, let us briefly notice that generalized mechanics in the classical sense has recently spawned two most important new theories of mechanics, namely quantum mechanics (with crucial use of probability theory) and relativity theory. Unfortunately, we have nothing further to say at this time about these important developments, but wish to draw attention briefly to the fact that classical mechanics and probability theory have also given rise to statistical mechanics--namely the description of large systems in terms of the average behavior of their myriad deterministic (or possibly quantum mechanical) components.

The theory of statistical mechanics is still in an unsatisfactory form, and we believe that its proper development is one of the great challenges of system theory. Here, however, let us briefly note that Poincaré, in pondering the various problems of celestial mechanics, came up with a very crucial notion of structural stability--a notion very much appropriate to the conduct of scientific study in a fuzzy world. Briefly, he noted that in taking any system, it is not possible to determine the parameters of that system with complete exactitude. It is thus, then, a matter of crucial import that no very delicate change in the parameters of the system should drastically alter its behavior--for then we could have confidence in the predictions that were made. This, then, is the idea of a structurally stable system: a system whose behavior is only changed slightly by a slight change in the parameters that describe the equations of motion of that system. Interestingly, these ideas of Poincaré have led to two recent developments. One is Thom's theory of catastrophes [16]--in which Thom classifies those parameters of system description which lie at the borderline between two different domains of structural stability. It is perhaps worth noting in passage our belief that Thom's mathematical contributions here are of vital importance to system theory; while at the same time expressing the greatest skepticism about the way in which Thom has suggested that his theory of catastrophes has important and immediate applications to such diverse fields of applied mathematics as theoretical embryology and linguistics.

A more direct descendent of Poincare's ideas is the theory of tolerance spaces due to Zeeman, in which he replaced the idea of a topology on a space by the more discrete notion of a tolerance: namely a reflexive and symmetric relation which tells us of any two points of the space whether or not they are in tolerance of one another. This then suggested to Arbib the idea of a tolerance automaton--namely a sequential machine in which the dynamics and output are 'continuous' with respect to tolerances on the various spaces involved. It has recently been noted by Dal Cin that we may make such tolerance automata into machines in a category in a fairly obvious way.

With this, then, let us turn to the remaining two evolutions in Figure 4--namely, that from the Markov chains developed by the probability theorists; and that which we may recognize as part of the evolution of multivalued and intuitionistic logics (the name of Post occurs here as well as in the canonical systems which led to formal language theory) from classical Boolean logic.

H: Markov chains were developed in the late 1800's as a way of modelling the dynamics of a classical system for which one could at best give probabilities as to the next state given the present state, rather than the classical systems with which we started our discussion in this paper in which the current state determined the future states for all time. The stochastic automaton, then, is related to Markov chains just as our control systems are related to classical mechanical systems. Namely, we introduce a set of inputs, such that for each input there is a corresponding Markov chain, with the probability distribution of the next state being determined by the Markov chain indexed by the current input.

More formally, a Markov chain M is given by a set $\{q_1,...,q_n\}$ of states and an $n \times n$ stochastic matrix $P = (P_{ij})$ whose interpretation is that if M is in state q_j at time t, then it will be in state q_i at time $t + 1$ with probability p_{ij}. A stochastic automaton has its dynamics given by a set $X_o = \{x_1,\ldots,x_m\}$ and a collection of m Markov chains, one P^x for each input $x \in X_o$. If M is in state q_j at time t and receives input x, then it will be in state q_i at time $t + 1$ with probability p^x_{ij}. Here the dynamics is

$$\delta: Q \times X_o \longrightarrow Q\mathbb{P} : (q_j,x) \longmapsto \begin{bmatrix} p^x_{1j} \\ \cdot \\ \cdot \\ \cdot \\ p^x_{nj} \end{bmatrix}$$

where $\mathbb{P}$: Set $\longrightarrow$ Set is a functor with

$Q\mathbb{P}$ = set of probability distributions on Q (If $p \in Q\mathbb{P}$, let $p(q)$ denote the probability of q)

$$f\mathbb{P} : Q\mathbb{P} \longrightarrow Q'\mathbb{P} : f\mathbb{P}(p) : q' \longmapsto \sum_{q \in f^{-1}(q')} p(q).$$

We see once again that the dynamics is of the form $QX \longrightarrow QT$, where now T is the functor $\mathbb{P}$: Set $\longrightarrow$ Set which sends a set Q to the set of all probability distributions on Q.

I; For our last example of a functor T for our general theory, we turn to fuzzy sets. This notion seems to have been independently established by Zadeh [17], although it is clearly a special case of ideas developed by many authors in looking at multivalued and intuitionistic logic. Briefly, Zadeh observed that there are many 'sets' in the world for which one cannot make the confident assertions of membership or nonmembership demanded by classical set theory. For example, the set of all 'tall people' is such a set. Certainly someone who is three feet tall does not belong to the set, while someone who is seven feet tall certainly does. But what of someone 5'3" tall? Perhaps they almost belong to it, say with 'weight' 0.3, while someone of height 5'8" might belong to the set with membership strength 0.8. On this basis, then, Zadeh defines a fuzzy set in the universe W to simply be a map A from W to the continuous interval [0,1] of real numbers, with $A(w)$ being the strength of membership of w in A.

Before going further, it is perhaps worth noticing that there is a certain horror in this approach to the problem of fuzziness--for if it seemed unreasonable to simply say of any element whether or not it belonged to the set of tall people, surely it seems even more unreasonable in this fuzzy world of ours to assign so precise a number as 0.7 to membership. It may perhaps be suggested that the appropriate approach to fuzzy sets is to realize that the fuzziness simply is imposed by the fact of undetermined context. If we are surrounded by short people, then we will say a person of 5'6" is tall; if we are meeting with the Watusi then such a person will be short. The idea, then, that a statement may have different truth values depending on the context suggests that there is implicit a whole series of mechanisms such as those that are being painfully developed in artificial intelligence approaches to the understanding of natural language [18]. But such an idea takes us too far afield from the particular historical domain of discourse that we have set for ourselves in this paper, and so now we return to fuzzy sets, with the observation that one can clearly define a suitable functor T associated with 'fuzzing' (indeed, $QT = [0,1]^Q$), and that with this we may then define <u>fuzzy-set automata</u> to be those with dynamics $\delta: QT \longrightarrow QT$, where T is the fuzzing functor.

With these three examples, we are ready to begin the development of our general theory. However, before we do so, it is worth making a couple of technical observations. Firstly, we may note that a continuous interval [0,1] may be replaced by any lattice, and that for technical reasons we shall usually want this to be a distributed lattice, and thus what is known as a semiring. In fact, Schützenberger [19] has constructed a rich theory of automata over semirings so that not only are fuzzy sets a particular case of models already developed in multivalued and intuitionistic logic; but the study of fuzzy automata is a special case of Schützenberger's theory. Secondly, we note that Goguen [20,21,22] has studied a category of fuzzy sets.

But all this is an aside, and it is time to return to the general study of dynamics of the form

$$\delta: QX \longrightarrow QT$$

which provide the dynamics of what we call FUZZY MACHINES. [We hope that Professor Zadeh will forgive us for appropriating his word for this general setting--we use the term <u>fuzzy-set</u> machine to refer to his special case.] Our first observation is that $QX \longrightarrow QT$ looks like a <u>generalization</u> of the case $QX \longrightarrow Q$ which is obtained by taking T to be the identity functor. It would be far more appealing, aesthetically, if in fact we could take $QX \longrightarrow QT$ to be a <u>special case</u>. But to do this we would have to consider a category $\mathcal{K}_{\underline{T}}$ whose objects are the same as those of the original category $\mathcal{K}$ but for which a morphism $A \longrightarrow B$ is actually a $\mathcal{K}$-morphism $A \longrightarrow BT$. In this case, a morphism $QX \longrightarrow Q$, and thus a dynamics, in our new category $\mathcal{K}_{\underline{T}}$ would indeed be a morphism $QX \longrightarrow QT$ in $\mathcal{K}$.

Recalling (Box 3) the need for <u>identities</u> and <u>composition</u> in defining a category, we can now develop a picture of what such a new category $\mathcal{K}_{\underline{T}}$ would look like. Our first requirement is that we can define identity morphisms for this category, and our choice for this is the morphism $Ae: A \longrightarrow AT$ which tells us how to interpret pure elements as particular examples of fuzzy elements. For example, when $T = 2^{(-)}$ we require $Ae: A \longrightarrow 2^A$ to send an element a of set A to the singleton {a} which is an element of the set 2^A of subsets of that set. Again, for $T = \mathbb{P}$, we require $Ae(a)$ to be the probability distribution on A for which a has probability 1. Given these identity morphisms, we can think of an ordinary morphism as a fuzzy morphism--namely we follow the morphism $A \longrightarrow B$ with the 'fuzzing morphism' Be. Our second requirement in making $\mathcal{K}_T$ a category is a composition of fuzzy morphisms, so that we may compose $A \longrightarrow BT$ with $B \longrightarrow CT$ to obtain a morphism $A \longrightarrow CT$--in such a way that we have the usual axioms of a category for associativity of composition, and the existence of the identities which we require to be the 'fuzzing morphisms' Ae:

$$\mathcal{K}(A,BT) \times \mathcal{K}(B,CT) \longrightarrow \mathcal{K}(A,CT) : (\alpha,\beta) \longmapsto \beta \circ \alpha$$

which satisfies

$$(\gamma \circ \beta) \circ \alpha = \gamma \circ (\beta \circ \alpha)$$

$$\alpha \circ Ae = \alpha = \alpha \circ Be.$$

(We also require that $\beta \circ (Be \cdot f) = \beta \cdot f$ for $f: A \longrightarrow B$, $\alpha: B \longrightarrow C$.) We call $\underline{T} = (T,e,comp)$, and the category $\mathcal{K}_{\underline{T}}$ it induces, a <u>fuzzy category</u> over $\mathcal{K}$. (Adepts at category theory should note [23,24] that the notion of a fuzzy category is equivalent to the notion of a Kleisli category.)

Having introduced the idea of fuzzy category we find that there is a fly in the ointment, and it must be removed: We have been looking at $\mathcal{K}$-morphisms $QX \longrightarrow QT$ and suggesting that the corresponding morphism from QX to Q in $\mathcal{K}_{\underline{T}}$ is a dynamics. But, unfortunately, so far we have only required X to be a functor on $\mathcal{K}$, not a functor on $\mathcal{K}_{\underline{T}}$. This suggests, then, that we try to 'lift' the functor X on $\mathcal{K}$ to a functor $\overline{X}$ on $\mathcal{K}_{\underline{T}}$. Clearly, X and $\overline{X}$ must act the same on objects. However, given a $\mathcal{K}$-morphism $A \longrightarrow BT$, the action of X will yield a $\mathcal{K}$-morphism $AX \longrightarrow BTX$, whereas $\overline{X}$ will yield a $\mathcal{K}_{\underline{T}}$-morphism $AX \longrightarrow BX$, i.e., a $\mathcal{K}$-morphism from AX to BXT.

We note that one way of reconciling this problem is simply to introduce for each object B a distinguished morphism

$$B\lambda: BTX \longrightarrow BXT$$

Then define, for $g: A \longrightarrow B = A \longrightarrow BT$

$$g\overline{X}: A\overline{X} \longrightarrow B\overline{X} = AX \longrightarrow BXT$$

to equal

$$AX \xrightarrow{gX} BTX \xrightarrow{B\lambda} BXT$$

the collection of BX's defined in this way, then λ must obey certain axioms which make it what a category theorist calls a distributive law. In fact, it can be verified that $\overline{X}$ is a lift of X if and only if it is obtained from X by using a distributive law λ in this way. Thus, we may always denote $\overline{X}$ by X_λ for the appropriate distributive law λ.

For example, in the case $X = -\times X_o$ and $T = 2^{(-)}$

$$Q\lambda: (2^Q) \times X_o \longrightarrow 2^{Q\times X_o} : (S,x) \longmapsto \{(s,x) | s \in S\}$$

is the only distributive law.

More generally, replacing $2^{(-)}$ with any T: Set $\longrightarrow$ Set gives rise to the distributive law

$$Q\lambda: QT \times X_o \longrightarrow (Q \times X_o)T : (P,x) \longmapsto (in_x T)(P)$$

where

$$in_x: Q \longrightarrow Q \times X_o : q \longmapsto (q,x).$$

Thus, there are many examples!

Once we have reached the stage of realizing that the proper setting for the study of nondeterministic automata is the category of some functor T using a functor X on $\mathcal{K}$ which can be lifted by a distributive law λ to a functor X_λ on $\mathcal{K}_T$ (Box 5) we can in fact show that many results holding for X are also available for X_λ. We can show that each X-dynamics 'is' an

Fuzzy Machines

$$\underline{(X,T)\text{-Machines}} \quad \begin{cases} \tau: I \longrightarrow QT \\ \delta: QX \longrightarrow QT \\ \beta: Q \longrightarrow Y \end{cases}$$

X: $\mathcal{K} \longrightarrow \mathcal{K}$ is a functor and $\underline{T}$ = (T,e,comp) is a fuzzy category for which there exists a distributive law λ: TX $\longrightarrow$ XT. Y is the carrier of a $\underline{T}$-algebra. τ, δ and β are $\mathcal{K}$-morphisms.

Box 5.

X_λ-dynamics, and we can show that each X_λ-dynamics may be 'simulated' by an X-dynamics. Moreover, if we can do reachability theory for X, we can also do it for X_λ. If we can do observability theory for X we can also do it for X_λ if certain conditions concerning "$\underline{T}$-algebras" are met. Finally--and this is a technical comment whose content is clearly beyond the scope of this exposition--we may note that the proper setting for the theory of minimal realization for these Kleisli machines is the treatment of $(X,\underline{T})$-composite algebras.

Unfortunately, there is no space here to give the necessary background on category theory to expand upon any of these results, or the earlier results of Part I. However, we can summarize our discoveries quite succinctly. The idea of a morphism

$$\delta: QX \longrightarrow Q$$

in a category $\mathcal{K}$ is the proper setting for the study of dynamics in a deterministic world. [We noted that the notion of left and right adjoint of a functor were crucial in studying reachability and observability, respectively, for such dynamics; as well as for approaching the theory of minimal realization.] What is perhaps most surprising is that dynamics in a fuzzy world is a <u>special case</u>, namely that in which the functor X is now an appropriate lifted functor X_λ, and the category in which the action takes place is a fuzzy category for some 'fuzzing functor' T. It is this 'surprise' that suggests that our general notion of a "Machine in a Category" of Part 1 is indeed a proper setting for system theory: for one of the best tests of proper generality of a theory is that it is <u>robust</u> in the sense that it can admit apparent extensions and special cases, rather than requiring a proliferation of super- and subscripts for each new variation that arises. In conclusion, we synthesize our overview in the mandala of Figure 5.

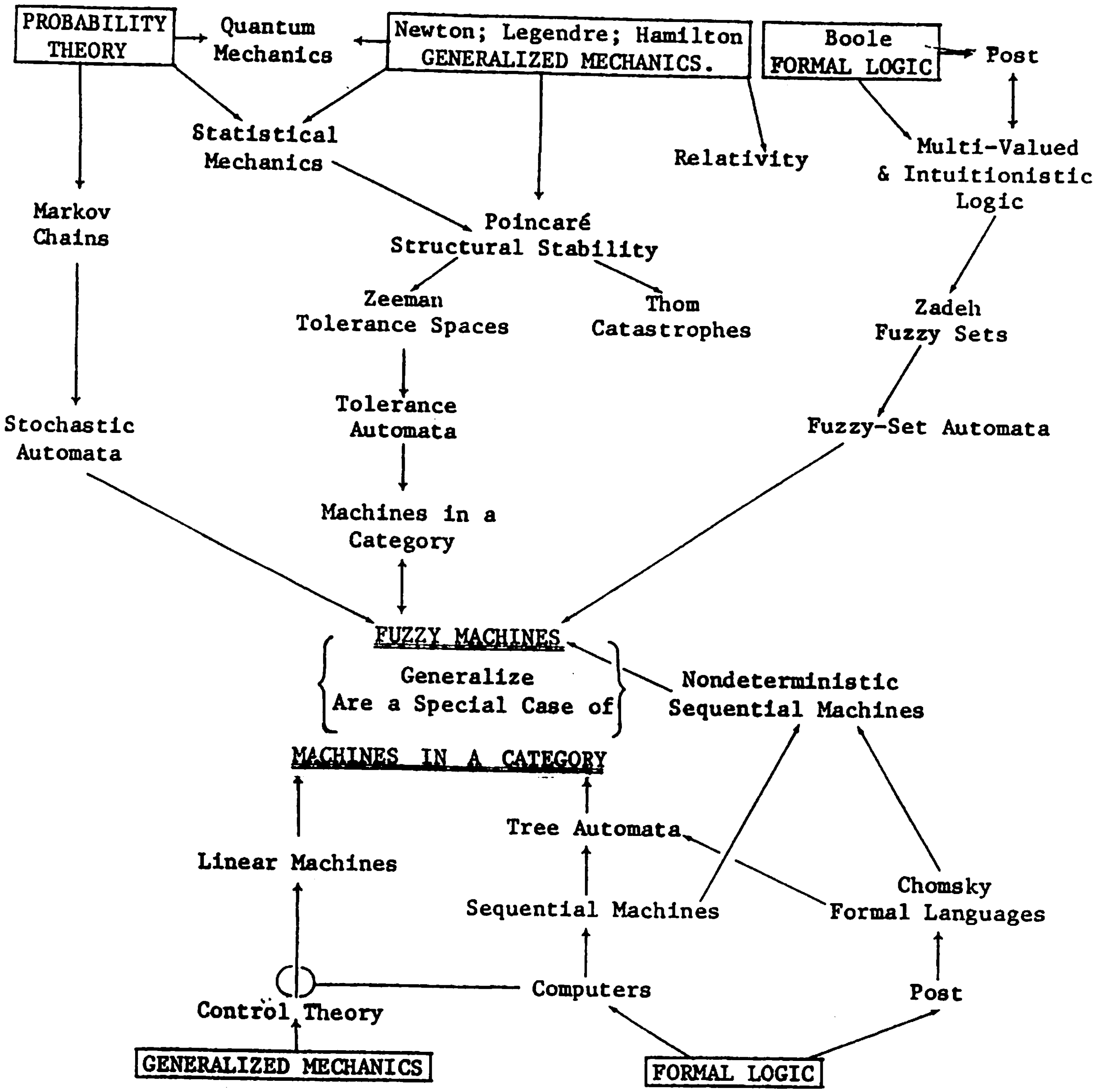

PROBABILITY THEORY
Quantum Mechanics
Newton; Legendre; Hamilton GENERALIZED MECHANICS.
Boole FORMAL LOGIC
Post
Statistical Mechanics
Relativity
Multi-Valued & Intuitionistic Logic
Markov Chains
Poincaré Structural Stability
Zeeman Tolerance Spaces
Thom Catastrophes
Zadeh Fuzzy Sets
Stochastic Automata
Tolerance Automata
Fuzzy-Set Automata
Machines in a Category
FUZZY MACHINES
Generalize
Are a Special Case of
Nondeterministic Sequential Machines
MACHINES IN A CATEGORY
Tree Automata
Linear Machines
Sequential Machines
Chomsky Formal Languages
Computers
Post
Control Theory
GENERALIZED MECHANICS
FORMAL LOGIC

Figure 5.

REFERENCES

[1] F. W. Lawvere (Ed.): Toposes, algebraic geometry and logic. Lecture Notes in Mathematics (1972) 274, Springer-Verlag.

[2] M. A. Arbib and E. G. Manes: Arrows, Structures, and Functors: The Categorical Imperative. Academic Press (1975).

[3] M. Athans and P. L. Falb: Optimal Control. McGraw-Hill (1966).

[4] L. Padulo and M. A. Arbib: System Theory. Saunders (1974).

[5] R. E. Kalman, P. L. Falb and M. A. Arbib: Topics in Mathematical System Theory. McGraw-Hill (1969).

[6] L. S. Bobrow and M. A. Arbib: Discrete Mathematics. Saunders (1974).

[7] M. A. Arbib and E. G. Manes: Machines in a category: an expository introduction. SIAM Review (1974) 16: 163-192.

[8] M. A. Arbib and E. G. Manes: Foundations of system theory: decomposable machines. Automatica (1974) 10: 285-302.

[9] M. A. Arbib and E. G. Manes: Fuzzy Machines in a Category. (To appear).

[10] E. G. Manes (Ed.): Proceedings of the First International Symposium: Category Theory Applied to Computation and Control. Lecture Notes in Computer Science, Springer-Verlag (to appear).

[11] J. A. Goguen: Minimal realization of machines in closed categories. Bull. Amer. Math. Soc. (1972) 78: 777-783.

[12] J. A. Goguen: Realization is universal. Math. Sys. Th. (1973) 6: 359-374.

[13] E. S. Bainbridge: Addressed machines and duality. In [10] above.

[14] H. Ehrig, K.-D. Kiermeier, M.-J. Kreowski and W. Kühnel: Universal Theory of Automata: a categorical approach. Teubner (1974).

[15] J. A. Goguen, J. W. Thatcher, E. G. Wagner and J. B. Wright: A junction between computer science and category theory, I: basic concepts and examples (part 1). IBM Research Report RC 4526, T. J. Watson Research Center (1973).

[16] Réné Thom: Stabilité structurelle et morphogénèse. W. A. Benjamin, Inc. (1972).

[17] L. Zadeh: Fuzzy sets. Inform. Control (1965) 8: 338-353.

[18] R. C. Schank and K. M. Colby (Eds.): Computer Models of Thought and Language. W. H. Freeman (1973).

[19] M. P. Schützenberger: On a Theorem of R. Jungen. Trans. Amer. Math. Soc. (1962) 13: 885-890.

[20] J. A. Goguen: L-fuzzy Sets. J. Math. Anal. Appl. (1967) 18: 145-174.

[21] J. A. Goguen: The logic of inexact concepts. Synthese (1969) 19: 325-373.

[22] J. A. Goguen: Axioms, extensions and applications for fuzzy sets: languages and the representation of concepts (to appear).

[23] Saunders Mac Lane: Categories for the working mathematician. Springer-Verlag (1972).

[24] E. G. Manes: Algebraic Theories. Springer-Verlag (1975).