SIAM Review
Vol. 16, No. 2, April 1974

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

MACHINES IN A CATEGORY: AN EXPOSITORY INTRODUCTION*

MICHAEL A. ARBIB† AND ERNEST G. MANES‡

Abstract. We present for the first time a general definition of machines in an arbitrary category which unifies the theories of sequential machines, linear control systems, tree automata and stochastic automata. To make the paper relatively accessible, we provide a careful exposition of the notions of automata theory we seek to unify, of some basic results of automata theory we seek to generalize, and of elementary concepts we require from category theory. We then state and prove our new fundamental realization theorem which shows how to construct a broad variety of machines with prescribed behavior.

Introduction. To motivate our approach, we outline two convergent themes, the first unifying automata theory and system theory and the second applying category theory and universal algebras to the tree-processing problems which have proved central to the study of programming languages. We then show that we have discovered a formalism which unites these two themes and provides a solid basis for future research.

A central problem in both automata theory and system theory has been the *realization problem*, namely, given a description of how a system processes inputs to yield outputs, to construct a *realization* of it, i.e., a state-variable description of a system which processes inputs in the specified way. Automata theorists concentrated on those cases in which the realization was a *finite-state* sequential machine, while control theorists emphasized *finite-dimensional linear* realizations. While many authors urged, and made partial contributions to, a rapprochement between automata theory and control theory ([2], [27], [48]), it remained for Arbib and Zeiger [9] to provide a presentation of the automata-theory of realization which yielded the results of the linear theory in a straightforward way which motivated and improved the results of the control theorists (e.g., [25]). It seemed that a category theory approach was best suited to polish and extend the contributions of Arbib and Zeiger, but despite an unpublished draft by Arbib in 1969, nothing came of this until 1971, when Goguen gave a categorical realization theory. However, as we shall see below, the work of Goguen [24] is not sufficiently general to handle important classes of sequential machines, let alone the tree automata to which we now turn.

Along a different line, computer scientists had been building on the theoretical linguistics of Chomsky [17] to provide a formalism for programming languages which made use of derivation trees of the symbol strings which constituted

* Received by the editors July 11, 1972, and in revised form June 20, 1973. The research reported in this paper was supported in part by the U.S. Army Research Office, Durham, under Contract DAHC 04-70-C-0043 and in part by the National Science Foundation under Grant GJ35759.

† Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01002.

‡ Department of Mathematics, University of Massachusetts, Amherst, Massachusetts 01002.

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

programs. For a long time, the theory concentrated on the manipulation of the strings rather than the manipulation of the underlying trees. However, this situation changed with the paper of Thatcher and Wright [46] who inter alia extended the observation of Büchi and Wright [15] that ordinary automata theory could be generalized to yield a theory of tree automata. Subsequent papers, e.g., Rounds [42] and Thatcher [44], [45], convinced many workers that programming language theory should be couched in terms of tree manipulation. Meanwhile, Eilenberg and Wright [22] showed the appropriateness of categorical notions due to Lawvere [29] to tree theory. Their work was extended and related to data structures by Give'on and Arbib [23], while others [20], [34] provided important extensions of Lawvere's ideas in the study of algebras over a triple.

We have now discovered an approach to automata theory which unifies these two themes. While the development of these ideas is still very much under way, we shall set forth our basic ideas in the present paper, which culminates in a new theorem providing an exceptionally general solution to the realization problem.

1. What are we trying to unify? In this section we briefly present four sources of intuition for our general theory: sequential machines, linear machines, stochastic automata and tree automata.

1.1. Sequential machines. Imagine a system such that each time we apply to it some input, say x from the set X_0 (the reason for the subscript will be explained in § 4), we receive from it an output, say y from the set Y. In general, y will be determined not simply by x, but also by the internal state of the system, say q from the set Q. In this subsection, we consider the case in which the set Q is sufficiently rich that q and x between them completely determine not only y, but also the internal state q' in which the system will receive its next input. Clearly, this situation will be as described by the following definition: a *sequential machine* (extended sense) is a quintuple $(X_0, Q, \delta, Y, \lambda)$, where X_0, Q and Y are sets (the input, state, and output sets, respectively), while

$$\delta : Q \times X_0 \to Q \quad \text{is a function, the } \textit{dynamics}$$

and

$$\lambda : Q \times X_0 \to Y \quad \text{is the } \textit{output} \text{ function.}$$

To simplify matters, note that if we take $Q' = Q \times Y$, then we may replace δ by $\delta' : Q' \times X_0 \to Q' : ([q, y'], x) \mapsto [\delta(q, x), \lambda(q, x)]$, and may replace λ by $\beta : Q' \to Y : [q, y'] \mapsto y'$. It should then be clear that if the original machine $(X_0, Q, \delta, Y, \lambda)$ is started in state q_0 and receives an input sequence, then it will emit the same sequence of outputs as will $(X_0, Q', \delta', Y, \beta)$ if it is started in state $[q_0, y']$, for any y' in X_0, and receives the same input sequence – except that the latter machine will emit an output one step later than the corresponding output will be emitted by the first machine. For this reason, we shall only consider sequential machines in the narrow sense (also known as state-output machines [3] and Moore machines [37]).

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

DEFINITION. A *sequential machine* is a quintuple $(X_0, Q, \delta, Y, \beta)$, where X_0, Q, and Y are sets (of *inputs*, *states* and *outputs*, respectively), while we call $\delta: Q \times X_0 \to Q$ the *dynamics* (others use *transition function*) and $\beta: Q \to Y$ is called the *output function.*

A special concern of sequential machine theory has been the following: given our *initial state* q_0 in Q, how does the machine $M = (X_0, Q, \delta, Y, \beta)$ process input strings? Let us be more precise about this.

Given the set X_0, we let X_0^* (the free monoid[1] generated by X_0) be the set of all strings of $n \geqq 0$ elements of X_0 in order

$$\langle x_1, x_2, \cdots, x_n \rangle$$

(we write Λ for the string $\langle\ \rangle$ of length 0), together with the binary operation of *concatenation*, $(w, w') \mapsto w \cdot w'$ or ww',

$$X_0^* \times X_0^* \to X_0^* : (\langle x_1 \cdots x_m \rangle, \langle x_1' \cdots x_n' \rangle) \mapsto \langle x_1 \cdots x_m x_1' \cdots x_n' \rangle$$

which is clearly associative with identity Λ.

We then inductively define $\delta^*: Q \times X_0^* \to Q$ to justify the statement, "if M starts in state q in Q and receives input string w in X_0^*, then it will end up in state $\delta^*(q, w)$".

$$\delta^*(q, \Lambda) = q \qquad \text{for all } q \text{ in } Q,$$

$$\delta^*(q, wx) = \delta(\delta^*(q, w), x) \quad \text{for all } q \text{ in } Q, \quad w \text{ in } X_0^*, \quad x \text{ in } X_0,$$

(where we have written wx as an abbreviation for $w \cdot \langle x \rangle$).

It is then easy to verify that

$$\delta^*(q, ww') = \delta^*(\delta^*(q, w), w') \quad \text{for all } q \text{ in } Q, \quad w, w' \text{ in } X_0^*.$$

δ^* is called the *run function* of M.

We next define

$$M_q: X_0^* \to Y : w \mapsto \beta[\delta^*(q, w)]$$

so that "if M starts in state q in Q and receives input string w in X_0^*, then it will end up by emitting output $M_q(w)$ in Y." We call M_q the *response function* or *behavior* of M started in state q.

Hence if we are given an initialized sequential machine $M = (X_0, Q, \delta, q_0, Y, \beta)$, we call M_{q_0} the *behavior* of M. It is this function $X_0^* \to Y$ which answers the question, "how does the machine process input strings?"

In § 2, we shall consider the converse problem of going from a function $f: X_0^* \to Y$ to a machine which behaves in the specified way. A further study of sequential machines can be found in [3] and the references cited there.

1.2. Linear systems. Control theorists have studied the problem of controlling a system to behave in some desired way. In many applications, the input, state and output sets of the system were finite-dimensional Euclidean vector spaces – $X_0 = \mathbb{R}^m$, $Q = \mathbb{R}^n$ and $Y = \mathbb{R}^p$ – and the dynamics of the system was described

[1] A *monoid* is a set together with an associative binary operation which possesses an identity (see, e.g., [3, § 2.3]).

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

by a differential equation:

$$\dot{q}(t) = \phi(q(t), x(t)).$$

Usually coordinates were so chosen that if the system were in its 0 state and received 0 input, it would not change state

$$\phi(0, 0) = 0$$

and thus, ϕ, if sufficiently smooth, would have a Taylor series expansion of the form

$$\phi(q, x) = Aq + Bx + \text{higher-order terms}$$

for suitable matrices A and B. Making a similar approximation for the output map, we can see the interest of control theorists in *linear systems* of the form

$$\dot{q}(t) = Aq(t) + Bx(t),$$
$$y(t) = Hq(t),$$

so long as ϕ is sufficiently smooth for the first order approximation to be meaningful. Now, let us choose a sufficiently small time unit so that $x(\tau)$ is essentially constant from t to $t + 1$. Then

$$q(t + 1) = e^A q(t) + \left[\int_t^{t+1} e^{A(t+1-\tau)} B \, d\tau\right] x(t)$$
$$= Fq(t) + Gx(t),$$

say. We thus find control theorists interested in what we may for now consider as a special case of a sequential machine.

DEFINITION. A *linear machine* is a sextuple (X_0, Q, F, G, Y, H), where X_0, Q and Y are modules over the same ring R, and

$$F: Q \to Q, \quad G: X_0 \to Q \quad \text{and} \quad H: Q \to Y$$

are module homomorphisms.

We have made the generalization from finite-dimensional vector spaces to R-modules,[2] and thus from matrices to module homomorphisms, because the increase in generality adds no algebraic work to much of the theory of linear machines; for duality theory, which we formulate generally in [5] and [6], an arbitrary ring will not suffice.

It is surprising that an algebraic analysis of the linear machine underlying a dynamical system is fruitful in real-life applications. We refer the reader to [27, Chap. 1] for more information.

A linear machine (F, G, H) becomes an initialized sequential machine M with dynamics $\delta(q, x) = Fq + Gx$, output $\beta = H$ and initial state 0. It is easy to compute that the behavior is given by a convolution

$$f(\langle x_n \cdots x_0 \rangle) = \sum_{j=0}^{n} HF^j G x_j.$$

[2] An R-module is like a vector space, except that the scalars lie in some ring R rather than in a field. See, for example, [33, Chap. 6].

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We note that $f(\langle 0\rangle w) = f(w)$ for all w in X_0^*. Let, then, $X_0^{\S}$ be the set of all functions

$$w: \mathbb{N} \longrightarrow X_0$$

of finite support (i.e., $w(n)$ is defined for all integers $n \geqq 0$, but is nonzero for only finitely many of them). We turn $X_0^{\S}$ into an R-module by defining

$$(rw + r'w')(n) = rw(n) + r'w'(n).$$

We may then regard f as a function $X_0^{\S} \longrightarrow Y$, i.e.,

$$f(w) = \sum_{\{n \mid w(n) \neq 0\}} HF^n Gw(n)$$

and it is then obvious that $f: X_0^{\S} \longrightarrow Y$ is linear. Thus for linear machines, the converse problem is to go from a *linear* $f: X_0^{\S} \longrightarrow Y$ to a linear machine whose zero-initial-state response it is.

1.3. Stochastic automata. In § 1.1, we considered a sequential machine as one for which the current q and x completely determine the next q' and output y. In this subsection, we consider the more general situation in which, for each q and x, we can determine a priori only the probability that any pair (q', y) will occur. One may study the probability that a given output sequence is emitted while the automaton receives a specified input sequence (which, in a sequel to [1] and [7], we shall identify as a nondeterministic generalized sequential machine), but here we shall only consider the simpler marginal distribution of final output symbols corresponding to an input sequence (treated further in [7]).

Given sets S, T, an S by T *stochastic matrix* is an $(S \times T)$-tuple (a_{st}) of nonnegative real numbers, such that for all $s \in S$, $a_{st} = 0$ for all but finitely many t, and $\sum_{t \in T} a_{st} = 1$. The set of all such we denote by $\mathscr{S}^{S \times T}$.

We introduce a $Q \times Q$ stochastic matrix $\delta(x)_{qq'}$ with the interpretation, "if the machine receives input x at time t, then the probability of a transition from state q to q' at time $t + 1$ is $\delta(x)_{qq'}$."

We introduce a $Q \times Y$ stochastic matrix $\beta_{qy'}$ with the interpretation, "if the machine is in state q at time t, it will emit output y' at time t with probability $\beta_{qy'}$."

DEFINITION. A *stochastic automaton* is a quintuple $(X_0, Y, Q, \delta, \beta)$, where $\delta: X_0 \longrightarrow \mathscr{S}^{Q \times Q}$ and $\beta \in \mathscr{S}^{Q \times Y}$.

We then define

$$\delta^*: X_0^* \longrightarrow \mathscr{S}^{Q \times Q}$$

inductively by

$$\delta^*(\Lambda) = I_Q, \quad \text{the } Q \times Q \text{ identity matrix,}$$

$$\delta^*(wz) = \delta^*(w)\delta(z) \quad \text{for all } w \in X_0^* \quad \text{and} \quad z \in X_0.$$

In other words, $\Pr(q_0 \overset{wz}{\longmapsto} q_j) = \sum_{q \in Q} \Pr(q_0 \overset{w}{\longmapsto} q) \Pr(q \overset{z}{\longmapsto} q_j)$.

δ^* is the *run function*. Let τ_q be "the probability that the machine starts in state q," that is, let $\tau \in \mathscr{S}^{1 \times Q}$ be the initial state. The *response* is the map

$$X_0^* \longrightarrow \mathscr{S}^Y: w \longmapsto \tau \cdot \delta^*(w) \cdot \beta.$$

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$(\tau \cdot \delta^*(w) \cdot \beta)_{y'}$ is "the probability of achieving output y' given the initial state distribution τ and the knowledge that the particular input sequence w was applied." For more on stochastic machines, see [40].

1.4. Tree automata. Suppose that we are given the rules $S \mapsto ab$ and $S \mapsto aSb$ (an S may be replaced by one of the strings ab or aSb) and are asked, "what strings of $\{a, b\}^*$ may be obtained from S by multiple use of the rules?" It is clear that a string may be so obtained only if it is the "frontier" of a tree (i.e., the ordered string of labels of the "leaves") of the type shown in Fig. 1, i.e., it is of the form $a^n b^n$

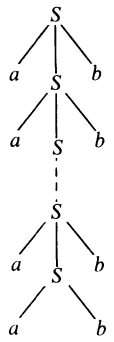


FIG. 1

for some $n \geqq 1$. In other words, the context-free languages of Chomsky-style linguistics [17] and ALGOL-style programming theory may be interpreted as part of a theory of *tree-processing*. It has thus come to be accepted that much of the theory of these languages can be deduced from the study of tree automata, which are defined as follows.

We represent trees by a set of addresses for their nodes; a (finite) tree is then a subset T of $\mathbb{N}^*$ (thus, it is a set of sequences of natural numbers) with the property that for each wx in T (with w in $\mathbb{N}^*$, x in $\mathbb{N}$) we must also have w in T, and wy in T for $0 \leqq y \leqq x$. See the example in Fig. 2, where Λ is the empty string in N^*.

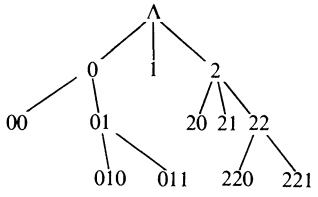


FIG. 2

We say wx (for $x \in \mathbb{N}$) is a *successor* of w in T if both w and wx belong to T; thus w has $n \geqq 1$ successors if $w(n-1)$ is in T, but wn is not in T. A node w is *terminal* if it has zero successors, i.e., w is in T, but $w0$ is not.

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

A group may be thought of as a set with 3 operators, a binary operation labeled $\cdot$ (we say the label $\cdot$ has *arity* 2 since $\cdot$ labels a 2-*ary* operator); a unary operation labeled $^{-1}$ (which has arity 1), and a constant labeled e (we say e has arity 0, and refer to constants as nullary operators). Generalizing, we have the following.

Let Ω be a set of operator labels, together with an arity function $\nu:\Omega \to \mathbb{N}$, so that ω can label an operator of the form $Q^n \to Q$ just in case $\nu(\omega) = n$. An Ω-*algebra* is a pair (Q, δ), where Q is a set, and δ assigns to each *label* $\omega \in \Omega$ an *operator* (map) $\delta_\omega : Q^{\nu(\omega)} \to Q$.

An Ω-tree is a tree correctly labeled with labels from Ω (i.e., a node with n successors may be labeled ω only if $\nu(\omega) = n$), while an Ω_Q-tree may also have a terminal node labeled by an element of Q. More formally, we let $\Omega_n = \nu^{-1}(n)$ for each $n \in \mathbb{N}$, and then say, for each tree T, that a map $h: T \to \Omega$ is an Ω-*tree* if each t in T has $\nu(h(t))$ successors; while an Ω_Q-*tree* is a map $h: T \to \Omega \cup Q$, where $\Omega_0 \cup Q$ is the set of nullary labels which may thus only occur on "leaves" of trees, while otherwise t has $h(t) \in \Omega$ with $\nu(h(t))$ successors.

Then a tree automaton is one with state set Q that works on Ω_Q-trees according to the rules specified by some Ω-algebra. For example, if $\Omega = \{+, \times\}$ with $\nu(+) = \nu(\times) = 2$, and if $Q = \mathbb{N}$, with δ_+ being numerical addition, and $\delta_\times$ being numerical multiplication, the corresponding tree automaton would work up the tree shown in Fig. 3 to produce the numerical result 20 (intermediate results being shown in parentheses).

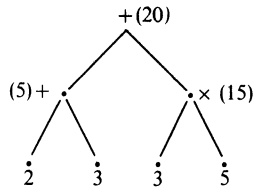


FIG. 3

DEFINITION. A *tree automaton* is a quintuple

$$M = (\Omega, Q, \delta, Y, \beta),$$

where Ω is a label set, (Q, δ) is an Ω-algebra, and $\beta: Q \to Y$ is the output function.

Ω plays the role that the input set plays for sequential machines, except that we interpret elements of Ω_0 as initial state labels rather than as input labels.

Given an Ω_Q-tree $h: T \to \Omega \cup Q$, we define M's processing of h to be the tree $\hat{h}: T \to Q$ which satisfies the following inductive definition.

(i) If $h(t) \in Q$, then $\hat{h}(t) = h(t)$.

(ii) If $h(t) \in \Omega_0$, then $\hat{h}(t) = \delta_{h(t)}$.

(iii) If $\nu(h(t)) = n + 1$ then $\hat{h}(t) = \delta_{h(t)}(\hat{h}(t0), \cdots, \hat{h}(tn))$.

Let $\mathcal{T}_{\Omega,Q}$ denote the set of Ω_Q-trees. The *run of* M is the map

$$\mathcal{T}_{\Omega,Q} \xrightarrow{\xi} Q : h \to \hat{h}(\Lambda)$$

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

(in the previous example, ξ assigns the number "20" and forgets its derivation). Since each Ω-algebra has a distinguished constant for each element of Ω_0, we think of Ω_0 as "the set of canonical initial states" – $\omega \in \Omega_0$ puts initial state $\delta_{h(\omega)}$ on a node – and regard tree automata as already initialized. The *response of* M is

$$\mathscr{T}_{\Omega,\varnothing} \xrightarrow{f} Y : h \longrightarrow \beta(\hat{h}(\Lambda))$$

which makes sense because $\mathscr{T}_{\Omega,\varnothing} \subset \mathscr{T}_{\Omega,Q}$.

Initialized sequential machines are recaptured by $\Omega_0 = \{0\}$, $\Omega_1 = X_0$, $\Omega_n = \varnothing$ for $n > 1$. Then an Ω-algebra (Q, δ) is a sequential dynamics $q, x \longmapsto \delta_x(q)$ together with an initial state $q_0 = \delta_0$. We have $\mathscr{T}_{\Omega,\varnothing} \cong X_0^*$, and the run and response maps reduce to those defined earlier. For more on tree automata, see [13, Chap. 4].

2. What are we trying to generalize? A fundamental problem in sequential machine theory is to "optimally compute" an arbitrary function $X_0^* \xrightarrow{f} Y$ by an initialized machine M_f of form $(X_0, Q_f, \delta_f, q_f, Y, \beta_f)$. Specifically, we require that M_f *realizes* (or computes) f, that is, its response is f; that every state is used, i.e., δ_f^* is onto (surjective); and that no two states do the same thing, meaning $q \longmapsto (M_f)_q$ is one-to-one (injective). Such M_f always exists, is unique up to isomorphism of initialized machines, and can be constructed in various ways [3, § 3.4]. One particular such construction, due to Nerode [39, Lemma 2], was popularized by Rabin and Scott in what is perhaps the first paper on "modern" sequential machines [41, Thm. 2]. (See also the related result of Myhill [38, Thm. 10].)

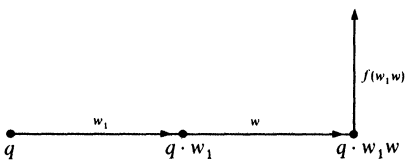


FIG. 4

Given a state q with response function $f : X_0^* \longrightarrow Y$, it is clear (Fig. 4) that the state $q \cdot w_1$ to which an input string w_1 sends q has response $f(w_1 w)$ to an input string w. Two strings w_1 and w_2 thus send q to states with the same response if and only if $f(w_1 w) = f(w_2 w)$ for all $w \in X_0^*$.

We thus define the *Nerode equivalence relation on* X_0^* for a given $f : X_0^* \longrightarrow Y$ by $E_f = \{(w_1, w_2)$: for all $w \in X_0^*$, $f(w_1 w) = f(w_2 w)\}$. A minimal realization is defined by

$$Q_f = X_0^*/E_f \quad \text{with} \quad q_f = [\Lambda],$$

$$\delta_f([w], x) = [wx],$$

and

$$\beta_f[w] = f(w).$$

In other words, the strategy has been to form a minimal realization by taking

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

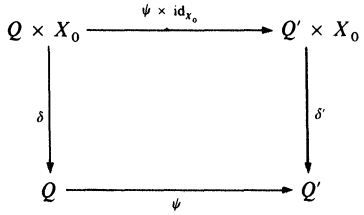
one state for each distinct response function for states reachable from the initial state. The concept of reachability is well-defined for all our examples as follows.

A linear M is *reachable* if the linear map

$$X_0^{\S} \longrightarrow Q : w \longrightarrow \sum_{\{n \mid w(n) \neq 0\}} F^n G w(n)$$

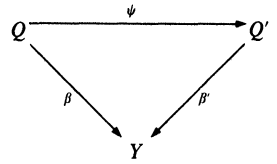
is onto. A stochastic M is *reachable* if there is nonzero probability of reaching all states, that is, for all $q \in Q$, there exists $w \in (X_0 \times Y)^*$ with $(\tau \cdot \delta^*(w))_q \neq 0$. A tree automaton M is *reachable* if the restriction of ξ to $\mathcal{T}_{\Omega,\varnothing}$ is onto. Reachable has thus been taken to mean "all states are used."

In a later paper [6] we shall introduce "state-behavior" machines as a general class of machines (including the machines in a closed category of Goguen [24] – our model has interesting points of similarity with the approach taken by Bainbridge [10]) for which the reduction of states with equal response makes sense. However, the machines in an arbitrary category which we present in § 4 are far more general than state-behavior machines, and for them the concept of "equality of response" is not generally available. Happily, there is another approach to minimization. For sequential machines, it is easy to check that M_f is characterized as a reachable realization of f which admits a unique machine map $M \xrightarrow{\psi} M_f$ from every other reachable realization, M of f; we simply map the state $q_0 \cdot w$ to $[w]$ of M_f, where $q_0 \cdot w$ is reachable by applying w to M in the initial q_0. By a *machine map* $M \xrightarrow{\psi} M'$, we mean a map of dynamics (where id_{X_0} is the identity map $x \longmapsto x$ of X_0)



$$\delta'(\psi(q), x) = \psi\delta(q, x)$$

which preserves the initial state ($\psi(q_0) = q_0'$) and commutes with output



$$\beta'\psi = \beta.$$

The role of ψ is to identify states that induce the same response. For linear machines, ψ must also be R-linear (though preserving 0 comes for free). In the stochastic case, $\psi \in \mathcal{S}^{Q \times Q}$ must preserve initial state ($\tau \cdot \psi = \tau'$), commute with output ($\psi \cdot \beta' = \beta$) and be a map of dynamics (for all $(x, y) \in X_0 \times Y$, $\delta(x, y) \cdot \psi = \psi \cdot \delta'(x, y)$). For tree automata, ψ is just an Ω-homomorphism (i.e., $\psi(\delta_w(q_1, \cdots, q_n)) = \delta'_w(\psi(q_1) \cdots \psi(q_n))$) which commutes with output. (Preserving the initial states is guaranteed.) We can now define the following.

The *minimal realization* M_f of f is (in all four cases) the reachable realization of f admitting a unique machine map from every other reachable realization

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

of f. M_f, if it exists, is easily checked to be unique up to isomorphism of initialized machines (cf. § 4). Kalman [26] and Arbib–Zeiger [9] constructed M_f for linear machines, and Give'on–Arbib did the same for tree automata [23, § 6.8]. As is well known [40, Thm. 1.3.8] a reduced-form stochastic machine may be state-equivalent to a *nonisomorphic* reduced-form machine, so M_f in the above sense need not exist for stochastic machines. Thus, although realization procedures are known for stochastic input/output functions, they will *not* be special cases of our general M_f procedure of § 5.

3. Basic concepts of category theory. With the above motivation, it is now time to set forth the basic notions of category theory (an excellent text is that of MacLane [32]) required for our general theory.

DEFINITION. A *category* $\mathscr{K}$ is given by a class $A, B, \cdots$ of *objects*; for each A, B we have a set $\mathscr{K}(A, B)$, called the set of *$\mathscr{K}$-morphisms* from A to B. (Note that $\mathscr{K}(A, B)$ may be empty if $A \neq B$.) For A, B, C we have a composition map (corresponding, in the case of functions, to composition of functions acting on the right)

$$\mathscr{K}(A,B) \times \mathscr{K}(B,C) \longrightarrow \mathscr{K}(A,C) : (A \xrightarrow{f} B, B \xrightarrow{g} C) \longmapsto A \xrightarrow{fg} C.$$

For all A, there is a distinguished identity element id_A in $\mathscr{K}(A, A)$. This assemblage satisfies the following axioms:

$$(fg)h = f(gh) \quad \text{for all } A \xrightarrow{f} B, B \xrightarrow{g} C, C \xrightarrow{h} D.$$

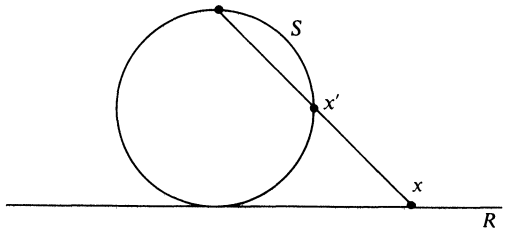
$$A \xrightarrow{\mathrm{id}_A} A \xrightarrow{f} B = A \xrightarrow{f} B = A \xrightarrow{f} B \xrightarrow{\mathrm{id}_B} B.$$

In looking for, say, a minimal automaton, we cannot expect to distinguish objects which differ only in the naming of their elements. So we need a concept of isomorphism to make precise the extent of this reasonable indistinguishability. We know that two *sets* X and Y are isomorphic if they satisfy either of these 2 equivalent conditions:

(i) $\exists\, X \xrightarrow{f} Y$ bijective (i.e., surjective and injective);

(ii) $\exists\, X \xrightarrow{f} Y, Y \xrightarrow{g} X$ with $fg = \mathrm{id}_X$, $gf = \mathrm{id}_Y$.

Unhappily, (i) does not make sense in an arbitrary category, and is not even desirable in many categories. For example consider the category $\mathscr{M}$ of metric spaces with uniformly continuous maps, and let S be the circle with the north pole deleted as shown below.

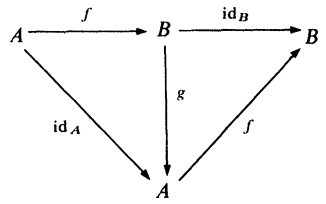


Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Observe that the map $\mathbb{R} \to S : x \mapsto x'$ of the figure is uniformly continuous. However, the sequence $\{1, 2, 3, 4, \cdots\} \subset \mathbb{R}$ is non-Cauchy, while $\{f(n)\}$ is Cauchy but not convergent, and $f^{-1} : S \to \mathbb{R}$ is not uniformly continuous. Thus f is a bijection, but it is not useful to consider it as an isomorphism in this particular category $\mathcal{M}$, and it does not satisfy condition (ii).

For such reasons, it has proved useful to adopt the following general definition.

DEFINITION. In a category $\mathcal{K}$, we say two objects A and B are *isomorphic*, $A \cong B$, iff there exists an *isomorphism* $A \xrightarrow{f} B$, i.e., f is such that there exists $B \xrightarrow{g} A$ with $gf = \mathrm{id}_B$ and $fg = \mathrm{id}_A$. To get into practice for future use, note that we may express the last two equalities by saying that the diagram shown below



commutes, i.e., no matter which directed path we pursue from one node to another, the overall morphism is the same. It is easy to check that $\cong$ is an equivalence relation on objects. We may think, heuristically, that $A \cong B$ iff "A and B have the same 'abstract structure'."

Think of a category as a monoid in which not all products are defined and which has a multiplicity of identities so that a homomorphism must preserve products and map identities to identities. Then we have the following definition.

DEFINITION. A homomorphism $\mathcal{A} \xrightarrow{H} \mathcal{B}$ of categories (called a *functor*) is defined by two functions

$$A \mapsto AH \quad \text{on objects}$$

$$(A \xrightarrow{f} B) \mapsto (AH \xrightarrow{fH} BH) \quad \text{on maps}$$

which satisfy these axioms:

$$(fg)H = (fH)(gH) \quad \text{for all composable } f, g \text{ in } \mathcal{A}$$

and

$$\mathrm{id}_A H = \mathrm{id}_{AH} \quad \text{for all } A \text{ in } \mathcal{A}.$$

Example 1. The identity functor $\mathrm{id}_{\mathcal{A}} : AH = A; fH = f$.

Example 2. $\mathbb{R}$-vect $\to$ Mon$:(X, \lambda, +) \mapsto (X, +)$ which "forgets" the scalar multiplication (represented by the λ) and only "remembers" the additive structure of a real vector space.

Example 3. Let X_0 be a set, and let $A = \mathfrak{S}$, the category of sets and arbitrary maps.

We may view $\mathfrak{S} \xrightarrow{- \times X_0} \mathfrak{S}$ as a functor which sends a set Q to the set $Q \times X_0$, and sends the map $f : Q \to Q'$ to the map, $f \times X_0 : Q \times X_0 \to Q' \times X_0 : (q, x)$

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$\mapsto (qf, x)$. Since $fg \times \mathrm{id}_{X_0} = (f \times \mathrm{id}_{X_0}) \cdot (g \times \mathrm{id}_{X_0})$ and $\mathrm{id}_Q \times \mathrm{id}_{X_0} = (\mathrm{id})_{Q \times X_0}$, $- \times X_0$ is indeed a functor. (Note that we now write functions on the right.)

We say a functor $\mathscr{A} \xrightarrow{H} \mathscr{B}$ is an *isomorphism* iff it satisfies either of these equivalent conditions:

(i) $A \mapsto AH$ is a bijection on objects, and $f \mapsto fH$ is a bijection on morphisms.

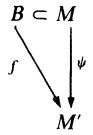
(ii) $\exists\, \mathscr{B} \xrightarrow{H^{-1}} \mathscr{A}$ such that $\mathscr{A} \xrightarrow{H} \mathscr{B} \xrightarrow{H^{-1}} \mathscr{A} = \mathrm{id}_{\mathscr{A}}$ and $\mathscr{B} \xrightarrow{H^{-1}} \mathscr{A} \xrightarrow{H} \mathscr{B} = \mathrm{id}_{\mathscr{B}}$.

The final ingredient we need here for our general theory of automata is the concept of *freeness*. We start with two examples.

Example 4. We may characterize free monoids in 2 equivalent ways; M is a *free monoid* on a basis B of generators iff either

(i) $B \subset M$, and for all w in M, either w equals the identity element e, or there exists exactly one n and exactly one sequence $(b_1, \cdots, b_n) \in B^n$ such that $w = b_1 \cdots b_n$, or

(ii) for all monoids M', and for all set maps $B \xrightarrow{f} M'$, there exists exactly one monoid homomorphism $M \xrightarrow{\psi} M'$ such that



commutes, namely $\psi(b_1 \cdots b_n) = \psi(b_1) \cdots \psi(b_n) = f(b_1) \cdots f(b_n)$.

That (i) $\Rightarrow$ (ii) is simple to verify, but (ii) $\Rightarrow$ (i) requires special properties of monoids (see, e.g., [13, § 9.1]).

Example 5. A free $\mathbb{R}$-vector space X with basis B may be characterized either by the fact that

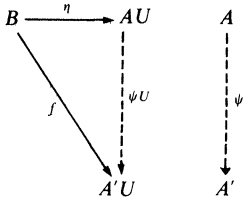
(i) $B \subset X$, and for all $x \in X$ there exists a unique representation $\sum_{b \in B} \lambda_b b$ with $\{b | \lambda_b \neq 0\}$ finite, or

(ii) a condition just as in (ii) for monoids, but using linear maps instead of homomorphisms.

This suggests that the second approach in each case yields the appropriate general notion (we provide heuristics after the definition).

DEFINITION. Given a functor $\mathscr{A} \xrightarrow{U} \mathscr{B}$, and an object $B \in \mathscr{B}$ we say (A, η) is *free* over B with respect to U if the following universal property holds.

For any $A' \in \mathscr{A}$ and any $f: B \longrightarrow A'U$ in $\mathscr{B}$, there is exactly one $\psi: A \longrightarrow A'$ such that $\eta \cdot \psi U = f$, i.e., such that the following commutes:

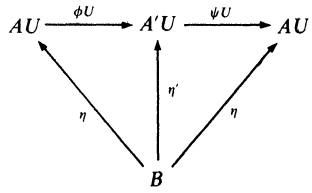


Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Heuristic meanings. Think of $\mathscr{A}$ as a category of $\mathscr{B}$-objects with additional structure and of U as the forgetful functor that throws away this structure. If B is a "basis" object and $A \in \mathscr{A}$, we generalize "B is a subset of A," which *really* says "B is a subset of AU," by specifying a map $B \xrightarrow{\eta} AU$; then we think of A as a "free object with η as inclusion of the generators," though η need not be "one-to-one" in the general setting. Our universal property then asserts that every map on B extends uniquely to an $\mathscr{A}$-morphism. We shall want cases in which (A, η) is in some sense constructible from B. Meanwhile, here are some properties.

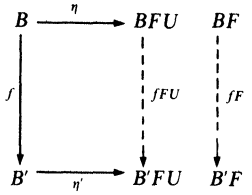
THEOREM. *If (A, η), (A', η') are both free over B with respect to U, then they are isomorphic in the sense that there exists an isomorphism $\phi : A \to A'$ such that $\eta' = \eta \cdot \phi U$.*

Proof. There exist unique $A \xrightarrow{\phi} A'$, $A' \xrightarrow{\psi} A$ such that



Using both category axioms on $\mathscr{A}$, the fact that U is functorial and the uniqueness assertion of the universal property, $\phi\psi = \mathrm{id}_A$. Similarly, $\psi\phi = \mathrm{id}_{A'}$. □

THEOREM. *Let $\mathscr{A} \xrightarrow{U} \mathscr{B}$ be such that there exists (A, η) free over B for every object B in $\mathscr{B}$. Define an object mapping $B \mapsto BF$ by letting (BF, η) be our choice of a free pair over B with respect to U. Then if $B \xrightarrow{f} B'$ in $\mathscr{B}$, define fF by the universal property shown below.*



Then $(B \mapsto BF, f \mapsto fF)$ is a functor, as can be checked from the uniqueness property of universality.

We call the functor F so defined the *left adjoint* of U, so that the functor $U : \mathscr{A} \to \mathscr{B}$ *has a left adjoint* just in case there exists (A, η) free over B for every object B in $\mathscr{B}$. Categorically speaking, we have only scratched the surface. Consult [8] for a gentle introduction to category theory with many examples, and MacLane [32] and the bibliography there for a thoroughgoing treatment of the subject.

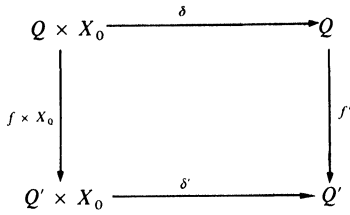
4. The general definition of machines in a category. We shall now show how the notion of a sequential machine may be embedded in a categorical framework (presented here for the first time) sufficiently rich to support all the examples of § 1.

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Working in the category $\mathfrak{S}$ of sets and maps, let X_0 be a fixed input set. We then define Dyn (X_0) to be the category with

$$\text{objects } (Q, \delta): Q \times X_0 \xrightarrow{\delta} Q \qquad (X_0\text{-dynamics})^3$$

and morphisms $(Q, \delta) \longrightarrow (Q', \delta')$ being maps $Q \xrightarrow{f} Q'$ for which



(f is an X_0-*dynamorphism*)

We then make the following claim.

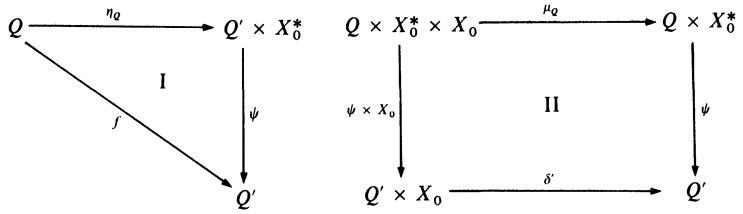
FUNDAMENTAL OBSERVATION. *The forgetful functor*

$$\text{Dyn}\,(X_0) \xrightarrow{U} \mathfrak{S}: (Q, \delta) \longmapsto Q$$

has a left adjoint.

Proof. Given any set Q, let us define an X_0-dynamics $\mu_Q: (Q \times X_0^*) \times X_0 \longrightarrow Q \times X_0^*$ by setting $((q, w), x)\mu_Q = (q, wx)$, i.e., it is just like the free monoid X_0^*, but with a passive Q-register attached. Let us define a map $\eta_Q: Q \longrightarrow Q \times X_0^*$ by $q\eta_Q = (q, \Lambda)$. Note that $Q \times X_0^* = (\mu_Q)U$. We shall verify that the pair (μ_Q, η_Q) is free over Q with respect to U.

In the display below,



I commutes $\Leftrightarrow (q, \Lambda)\psi = (q)f$ for all $q \in Q$,

II commutes $\Leftrightarrow (q, wx)\psi = ((q, w)\psi, x)\delta'$ for all $q \in Q$, $w \in X_0^*$, and $x \in X_0$.

Thus I gives us the basis step for the definition of ψ, and II yields the induction step for defining ψ, and we recognize it as

$$(q, w)\psi = ((q)f, w)(\delta')^*. \quad \square$$

We make the validity of the above theorem the axiom for our general definition.

---

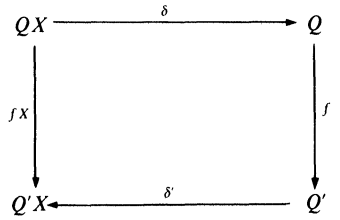
[3] Since a morphism comes equipped with a domain and codomain, the notation (Q, δ) is redundant; if Q is the codomain of a morphism δ, then it is an X_0-dynamics just in case its domain is $Q \times X_0$. Thus we may refer to a dynamics either as (Q, δ) or simply as δ when the codomain is clear from context.

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We first need a crucial change in viewpoint which is one of the major contributions of this paper: *We think of the input not as the set X_0 but as the* **process** *(functor) $- \times X_0 : \mathfrak{S} \to \mathfrak{S}$ which transforms the state set Q into the set $Q \times X_0$ upon which a dynamics δ acts.* We reserve the symbol X for the entire "process" $Q \to Q \times X_0$, explaining the zero subscript appearing in § 1. This viewpoint yields the following immediate generalization.

DEFINITION. A *process* in an arbitrary category $\mathscr{K}$ is a functor $X : \mathscr{K} \to \mathscr{K}$.

Dyn (X) denotes the *category of X-dynamics* whose objects are pairs (Q, δ), with Q an object of $\mathscr{K}$ and $\delta : QX \to Q$ a morphism in $\mathscr{K}$ (note that for each Q there may be many δ's), and whose morphisms $(Q, \delta) \xrightarrow{f} (Q', \delta')$ are *X-dynamorphisms*, being $\mathscr{K}$-morphisms $Q \xrightarrow{f} Q'$ rendering



commutative (so that an $f : Q \to Q'$ may be a dynamorphism $(Q, \delta_1) \to (Q', \delta_1')$ and yet not be a dynamorphism $(Q, \delta_2) \to (Q', \delta_2')$).

Composition and identities are defined as in $\mathscr{K}$, and it is straightforward to check that Dyn (X) is a category equipped with a natural underlying $\mathscr{K}$-object functor Dyn $(X) \to \mathscr{K} : (Q, \delta) \mapsto Q$.

Henceforth we shall only consider processes which share with $- \times X_0$ the property that there exist free dynamics.

DEFINITION. X is an *input* process if "X-machines can run," that is, just in case the forgetful functor Dyn $(X) \to \mathscr{K} : (Q, \delta) \mapsto Q$ has a left adjoint. In this case, the free dynamics on Q will be denoted $(QX^{@}, \mu_Q : QX^{@}X \to QX^{@})$ (which thus has state object $QX^{@}$) with "inclusion of generators" $\eta_Q : Q \to QX^{@}$.

Note that for the case $X = - \times X_0$, we have $X^{@} = - \times X_0^*$; $(X^{@} : \mathscr{K} \to \mathscr{K}$ is always a functor by the last theorem of § 3).

DEFINITION. A *machine* in the category $\mathscr{K}$ is a 7-tuple

$$M = (X, Q, \delta, I, \tau, Y, \beta),$$

where X is an input process,

$(Q, \delta) \in$ Dyn (X) – we call Q the *state object*,

I is an object of $\mathscr{K}$, the *initial state object*,

$I \xrightarrow{\tau} Q$ is a $\mathscr{K}$-morphism called the *initial state*,

Y is an object of $\mathscr{K}$, the *output object*,

$Q \xrightarrow{\beta} Y$ is the *output map*.

(In ordinary automata theory, we usually take $I = \{0\}$ and refer to $\tau(0) = q_0$ as *the* initial state.)

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

It is easy to describe linear systems in our language. Let $\mathscr{K}$ be the category of R-modules and R-linear maps and let $X:\mathscr{K} \to \mathscr{K}$ be defined by $QX = Q \oplus X_0$. Then a dynamics $QX \xrightarrow{\delta} Q$ decomposes uniquely into $\begin{pmatrix} F \\ G \end{pmatrix}$, where $\begin{pmatrix} F \\ G \end{pmatrix}(q, x) = Fq + Gx$, defining $F:Q \to Q$ and $G:X_0 \to Q$. The free dynamics on Q is $Q^{\S} \oplus X_0^{\S}$. When $I = \{0\}$ ("zero-state response"), the machine runs as expected. The details are spelled out in [39a, Chap. 8]. A subsequent paper [5] on "decomposable" machines will point out that we can recapture the theory of linear machines even more simply by taking X to be the identity functor in $\mathscr{K}$! Given this, it is hardly surprising that the constructions underlying the realization theories for linear systems are available in almost all familiar categories. The important structural fact is that $X_0^{\S}$, when equipped with the obvious "zero" $X_0 \xrightarrow{\text{in}_0} X_0^{\S}:x \mapsto (\cdots, 0, \cdots, 0,x)$ and "left shift" $X_0^{\S} \xrightarrow{z} X_0^{\S}:(\cdots, x_1, x_0) \mapsto (\cdots, x_1, x_0, 0)$, has the universal property of simple recursion [31, p. 1907, Axiom 3], and hence linear machines run more simply than sequential machines since simple recursion is simpler than recursion on strings. Recalling from § 1.2 that $X_0^{\S}$ is essentially X_0^* as a set, except that we identify sequences that differ only by leading zeros, it is tempting to conjecture that "$X_0^{\S} = X_0^*$ as sets" is true more generally. For example, if X_0 is a group, so is X_0^* (in the same way as for R-modules). But given a "linear" system (F, G, H) in the base category of groups, Arbib observed [4] that the response map $X_0^* \to Y$ need not always be a group homomorphism. His coproduct construction, which solves problems raised by Brockett and Willsky [14], falls under the formalism of [5] but not of [24].

We may include stochastic automata in our formalism as follows. Let the base category $\mathscr{K}$ have sets $S, T, \cdots$ as objects, but stochastic matrices

$$(a_{st}: s \in S, t \in T): S \to T$$

as morphisms. $\mathscr{K}$ is a category via the usual definitions of identity matrix and matrix multiplication. Then given a stochastic machine $(X_0, Y, Q, \delta, \beta)$, $S:X_0 \to \mathscr{S}^{Q \times Q}$ may be captured as the $\mathscr{K}$-morphism $Q \times X_0 \to Q$ whose (q, x, q')th component is $\Pr(q'|q, x)$. In a subsequent paper [7], we will show that because $\mathscr{K}$ is what is known (see [28], [32, p. 143], [34, § 1.2]) as the Kleisli category of an algebraic theory over which the input process $- \times X_0$ in $\mathfrak{S}$ distributes [12], the object function $- \times X_0$ becomes an input process in $\mathscr{K}$. So long as the initial state object has one element, the machines will run as advertised in § 1.3.

Tree automata, though appearing "nonsequential," can be recaptured, too. Given (Ω, ν) as before, define the process X in the base category $\mathfrak{S}$ by

$$QX = \coprod_{\omega \in \Omega} Q^{\nu(\omega)},$$

where $\coprod$ denotes "disjoint union." Then $\mathrm{Dyn}(X)$ is the category of Ω-algebras and Ω-homomorphisms (as defined in § 1.4) and X is an input process by a well-known theorem of universal algebra that free Ω-algebras exist (e.g., see [18, Thm. III, 2.6]); indeed, $QX^{@} = \mathscr{T}_{\Omega,Q}$ with the obvious Ω-operations and inclusion $Q \to \mathscr{T}_{\Omega,Q}$. To recapture the other details of § 1.4, use the empty set $\varnothing$ as the initial state object.

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

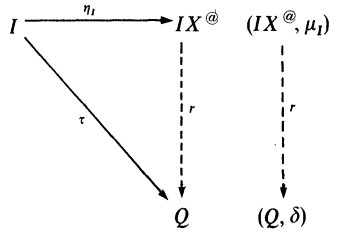
We now return to an arbitrary machine M. $IX^{@}$ is called the *object of inputs.* In the case $X = - \times X_0$ in $\mathfrak{S}$,

$$IX^{@} = I \times X_0^* \cong X_0^* \quad \text{if } |I| = 1.$$

Just as in sequential machine theory, we have the "reachability map"

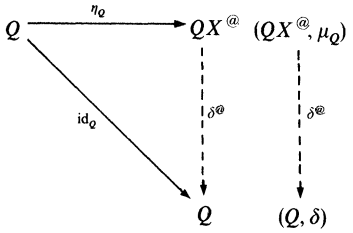
$$r : X_0^* \longrightarrow Q : w \longmapsto \delta^*(q_0, w),$$

so in our general theory we can use our freeness axiom to form the *reachability map* r, as shown below,



as the unique dynamorphic extension of τ. (Strictly speaking, we should write $rU : (IX^{@}, \mu_I)U \longrightarrow (Q, \delta)U$ in the triangle on the left in conformity with the picture of the universal property displayed in the definition of left adjoint. In this particular case, $(IX^{@}, \mu_I)U = IX^{@}$, $(Q, \delta)U = Q$ and the same symbol r is used in Dyn (X) and in $\mathscr{K}$ in accordance with the definition of X-dynamorphism.)

Again, by freeness, we may define the *run map* $\delta^{@} : QX^{@} \longrightarrow Q$ as the unique dynamorphic extension of the identity map as shown below.



This does indeed reduce to $\delta^* : Q \times X_0^* \longrightarrow Q$ when $X = - \times X_0$ in $\mathfrak{S}$. The reader may check his or her understanding of the uniqueness condition in our freeness axiom by verifying that

$$r = (\tau X^{@}) \cdot \delta^{@}.$$

By the *response* (or *system behavior*) *of* M we mean

$$IX^{@} \xrightarrow{r} Q \xrightarrow{\beta} Y.$$

This just reduces to $M_{q_0} : X_0^* \longrightarrow Y$ in the classical case.

For fixed I, Y and X, a *response morphism* is a $\mathscr{K}$-morphism

$$IX^{@} \xrightarrow{f} Y.$$

We say M *realizes* (or *is a realization of*) f just in case $f = r\beta$. In the next section,

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

we shall present very general conditions under which an f has a minimal realization, but here we devote the rest of this section to a discussion of reachability.

In the classical case, we say that M is *reachable* if the map $r: X_0^* \to Q$ is *onto*; i.e., just in case every state q of Q is representable in the form $\delta^*(q_0, w)$ for some w in X_0^*, and hence is reachable from the initial state. Clearly, then, our definition of reachability will depend on how we define *onto* in our category $\mathscr{K}$.

We make the following definitions.

DEFINITION. In a category $\mathscr{K}$, we say that a morphism $f: X \to Y$ is

(i) an *epimorphism* (or f is *epimorphic*) if whenever $g: Y \to Z$ and $h: Y \to Z$ satisfy $fg = fh$, it must follow that $g = h$;

(ii) a *split-epimorphism* if there exists a morphism $s: Y \to X$ such that $Y \xrightarrow{s} X \xrightarrow{f} Y = \mathrm{id}_Y$.

Clearly, every split-epimorphism is an epimorphism, since $sf = \mathrm{id}_Y$, and $fg = fh$ implies

$$g = \mathrm{id}_Y \cdot g = (sf)g = s(fg) = s(fh) = (sf)h = \mathrm{id}_Y h = h.$$

In $\mathfrak{S}$, the axiom of choice implies that whenever f is onto, it is a split epimorphism, for it lets us go from the assertion that for all $y \in Y$ there is an x in X for which $xf = y$ to the assertion that there is a map $s: Y \to X$ which does the job of taking us from y to $x: (ys)f = y$ for all y in Y.

However, in such categories as groups, monoids and lattices, onto morphisms are rarely split; for example, the group homomorphism $\mathbb{Z} \to \mathbb{Z}_2$ which sends $2n$ to 0 and $2n + 1$ to 1 is clearly not split (because every homomorphism from $\mathbb{Z}_2$ to $\mathbb{Z}$ is constantly 0).

This is as distinct from $\mathfrak{S}$, where split-epimorphism = epimorphism = onto; we have already seen that split-epimorphism is epimorphic in any $\mathscr{K}$, and that onto is split-epimorphic in $\mathfrak{S}$. It only remains to verify that epimorphic is onto in $\mathfrak{S}$. Given $f: X \to Y$, consider two characteristic functions from Y to $\{0, 1\}: \chi_Y$, the constant function with value 1, and $\chi_{f(X)}$, the characteristic function of the image of f. We then have that

$$X \xrightarrow{f} Y \xrightarrow{\chi_X} \{0, 1\} = X \xrightarrow{f} Y \xrightarrow{\chi_{f(X)}} \{0, 1\},$$

so that if f is epimorphic, we must have $\chi_Y = \chi_{f(X)}$, so that $Y = f(X)$, and f is indeed onto.

While no single definition of "epimorphism" seems to work for all of mathematics, experience leads us to expect that in each case, each split epimorphism is "epi" and each "epi" is, at least, epimorphic. There is generally a lot of room in between. For example, in the category of monoids and monoid homomorphisms, every split epimorphism is onto, although not all onto homomorphisms are split (e.g., $\mathbb{Z} \to \mathbb{Z}_2$ above); all onto homomorphisms are epimorphisms although the inclusion of the natural number $\mathbb{N}$ into $\mathbb{Z}$ is an epimorphism which is not surjective. Since all of the various definitions of epimorphism are equivalent in $\mathfrak{S}$, it was possible to speak of a machine as being reachable without specifying in what sense r was an epimorphism. However, in our general theory, it will be necessary to select some class $\mathscr{E}$ of epimorphisms which is closed under composition and contains all isomorphisms (in fact, in the next section, we shall be led to

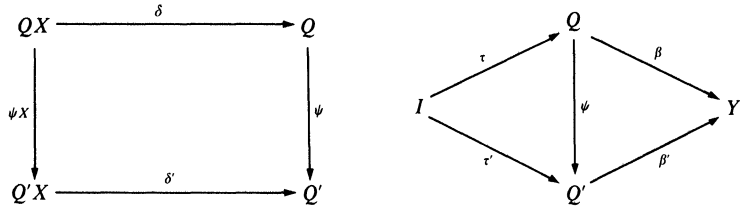
Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

choose yet another class, the coequalizers) and then make our definitions as follows.

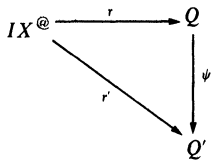
A machine M in K is *$\mathscr{E}$-reachable* if $IX^{@} \xrightarrow{r} Q$ is in $\mathscr{E}$. We then define the category $\mathscr{E}_f$ of $\mathscr{E}$-reachable realizations of f to have

Objects: $\mathscr{E}$-reachable machines M which realize f ($f = r\beta$).

Morphisms: A morphism $M \xrightarrow{\psi} M'$ is a dynamorphism $(Q, \delta) \xrightarrow{\psi} (Q', \delta')$ compatible with the input and output maps, i.e., $\psi : Q \to Q'$ is such that the two diagrams below commute.



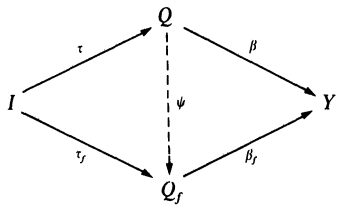
Notice that this implies that the diagram shown below



commutes, because the two dynamorphisms $r\psi$ and r' agree on the generators: $\eta_I \cdot r\psi = \tau\psi = \tau' = \eta_I \cdot r'$.

In any category, an object T is *terminal* if for all objects K there exists a unique morphism $K \to T$. Note that if T and T' are terminal, then $T \xrightarrow{\alpha} T' \xrightarrow{\beta} T \Rightarrow \alpha\beta = \mathrm{id}_T$, $\beta\alpha = \mathrm{id}_{T'}$, so $T \cong T'$. We say J is *initial* if for all objects K, there exists a unique morphism $J \to K$.

The *$\mathscr{E}$-minimal realization* M_f of f (if it exists) is then defined to be the terminal object (unique up to isomorphism) of $\mathscr{E}_f$, i.e., for all M in $\mathscr{E}_f$ there exists exactly one dynamorphism $\psi : M \to M_f$ such that



Axioms on $\mathscr{E}$ will guarantee that for any morphism $M \xrightarrow{\psi} M'$ in $\mathscr{E}_f$, the underlying $Q \xrightarrow{\psi} Q'$ will be in $\mathscr{E}$. (Specifically, we usually expect $\mathscr{E}$ to satisfy "$fg \in \mathscr{E}$

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

implies $g \in \mathscr{E}$," a familiar property of onto maps.) Thus, we may regard Q_f as obtained by "merging states" of any other machine in $\mathscr{E}_f$.

In the case of sequential, linear and tree automata (with $\mathscr{E}$ = "onto maps" in all cases) this definition of M_f agrees with the definitions of § 2.

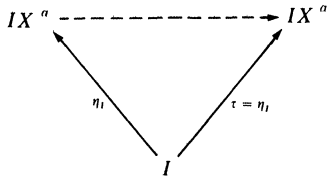
Before giving a general theory for finding M_f in the next section, we close by showing, easily, that any f always has at least one realization whenever X is an input process.

THEOREM. *For any $f: IX^{@} \to Y$ for which X is an input process, $\mathscr{E}_f$ is non-empty. Further, $\mathscr{E}_f$ has an initial object, the "free realization"*

$$F_f = (X, IX^{@}, \mu_I, I, \eta_I, Y, f)$$

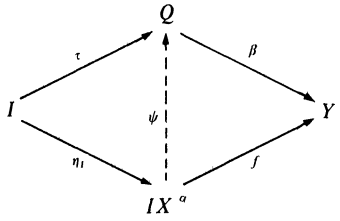
with input process X, state object $IX^{@}$, free dynamics $\mu_I: (IX^{@})X \to IX^{@}$; input morphism $\eta_I: I \to IX^{@}$ and output morphism $f: IX^{@} \to Y$.

Proof. Since the reachability map of F_f is the unique extension in the diagram below,



it must be id_{IX^a}. Thus the system behavior of F_f is indeed f.

To check that F_f is initial, we pick any M in $\mathscr{E}_f$ and look at the following diagram, where ψ is defined to be the reachability map of M.



The left-hand side of the diagram already commutes, and $\psi\beta = f$ since M is in $\mathscr{E}_f$. □

5. A general theory of machine realization. Now that we have seen how a functor $X: \mathscr{K} \to \mathscr{K}$ which is an input process may replace the $- \times X_0$ of "classical" sequential machine theory, we present in this section a way of restating the classical theory in categorical terms which has proved of sufficient generality to solve the machine realization problem in all cases (other than the stochastic; see [40]) presented in this paper and many more besides. We divide this section into alternating subsections, with "Motivation" (a transmutation of the classical theory) preceding "Theory" (the general categorical statement).

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

5.1. Motivation. In the classic case, we are given $f: X_0^* \longrightarrow Y$, and define the Nerode equivalence relation $E_f \subset X_0^* \times X_0^*$ by the rule

$$(1) \qquad (w_1, w_2) \in E_f \Leftrightarrow (w_1 w)f = (w_2 w)f \quad \text{for all } w \text{ in } X_0^*.$$

Noting that $X_0^* = \bigcup_{n \geq 0} X_0^n$, we may certainly replace this condition by

$$(2) \qquad (w_1, w_2) \in E_f \Leftrightarrow [(w_1 w)f = (w_2 w)f \text{ for all } w \text{ in } X_0^n] \quad \text{for all } n \in \mathbb{N}.$$

Defining the map $\mu_0: X_0^* \times X_0 \longrightarrow X_0^* : (w, x) \longmapsto w\langle x \rangle$, we may extend it to $X_0^* \times X_0^n$ by the rule

$$\mu_0^{(n)}: X_0^* \times X_0^n \longrightarrow X_0^*$$

$$= (X_0^* \times X_0) \times X_0^{n-1} \xrightarrow{\mu_0 \times \mathrm{id}_{X_0^{n-1}}} X_0^* \times X_0^{n-1} \xrightarrow{\mu_0 \times \mathrm{id}_{X_0^{n-2}}} \cdots \xrightarrow{\mu_0 \times \mathrm{id}_{X_0}}$$

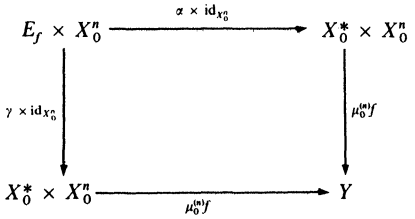
$$\times X_0^* \times X_0 \xrightarrow{\mu_0} X_0 : (w, x_1 x_2 \cdots x_n) \longmapsto (w x_1, x_2 \cdots x_n) \longmapsto \cdots$$

$$\longmapsto (w x_1 \cdots x_{n-1}, x_n) \longmapsto w x_1 x_2 \cdots x_n,$$

so that $\mu_0^{(n+1)} = \mu_0 X^n \cdot \mu_0^{(n)}$, where X is the functor $- \times X_0$. Since $\mu_0 X^n = \mu_0 \times \mathrm{id}_{X_0^n}$, (2) may be re-expressed in the form

$$(4) \qquad (w_1, w_2) \in E_f \Leftrightarrow [(w_1, w)\mu_0^{(n)} f = (w_2, w)\mu_0^{(n)} f \text{ for all } w \text{ in } X_0^n] \quad \text{for all } n \in \mathbb{N}.$$

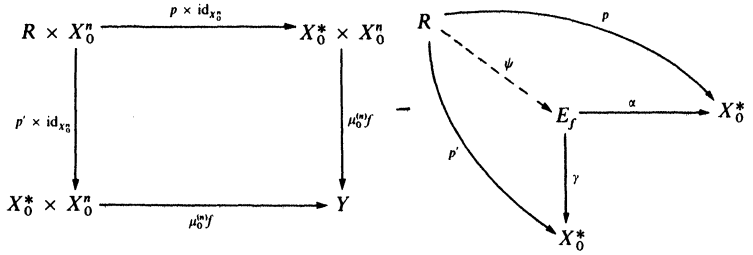
Let now $E_f \xrightarrow{\alpha} X_0^*$ and $E_f \xrightarrow{\gamma} X_0^*$ be the respective projections $(w_1, w_2) \longmapsto w_1$ and $(w_1, w_2) \longmapsto w_2$. Then it is clear that the "$\Rightarrow$" of (4) is expressed by the commutativity of the following diagram for all $n \in \mathbb{N}$.



What about the "$\Leftarrow$" of (4)? It says that if we are given two functions $p, p': R \longrightarrow X_0^*$ and r in R such that $[(rp, w)\mu_0^{(n)} f = (rp', w)\mu_0^{(n)} f$ for all w in $X_0^n]$ holds for all $n \in \mathbb{N}$, then (rp, rp') must be in E_f. A map $\psi: R \longrightarrow E_f$ is then defined by the rule $r\psi = (rp, rp')$. This justifies the following purely arrow-theoretic re-presentation of the definition (1).

E_f is defined by a diagram $E_f \overset{\alpha}{\underset{\gamma}{\rightrightarrows}} X_0^*$ such that $(\alpha \times \mathrm{id}_{X_0^n})\mu_0^{(n)} f = (\gamma \times \mathrm{id}_{X_0^n})\mu_0^{(n)} f$ for all $n \in \mathbb{N}$, and such that whenever $R \overset{p}{\underset{p'}{\rightrightarrows}} X_0^*$ satisfies $(p \times \mathrm{id}_{X_0^n})\mu_0^{(n)} f = (p' \times \mathrm{id}_{X_0^n}) \cdot \mu_0^{(n)} f$ for all $n \in \mathbb{N}$, (see the square below) there exists a unique morphism $\psi: R \longrightarrow E_f$ rendering the triangles shown below in (5) commutative.

(5)



Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

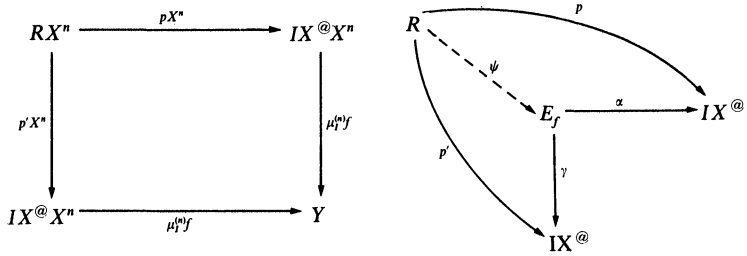
To pass to the general case, we simply replace X_0^* by $IX^@$ and $- \times X_0^n$ by X^n. However, whereas in $\mathfrak{S}$ we know that (5) always defines an equivalence relation in our general theory, we must *postulate* that the corresponding diagram can indeed be properly formed.

5.2. Theory. Let $X : \mathscr{K} \to \mathscr{K}$ be an input process, $IX^@X \xrightarrow{\mu_I} IX^@$ the free dynamics over I in $\mathscr{K}$, and let $f : IX^@ \to Y$ be a response morphism. For any $QX \xrightarrow{\delta} Q$ let us define $\delta^{(n)} : QX^n \to Q$ inductively, for $n \in \mathbb{N}$, by the rule,

$$\delta^{(0)} = \mathrm{id}_Q; \qquad \delta^{(n+1)} = QX^{n+1} \xrightarrow{\delta X^n} QX^n \xrightarrow{\delta^{(n)}} Q.$$

Postulate 1. Then we say that f satisfies Postulate 1 if there exists a diagram $E_f \overset{\alpha}{\underset{\gamma}{\rightrightarrows}} IX^@$ such that $\alpha X^n \cdot \mu_I^{(n)} f = \gamma X^n \cdot \mu_I^{(n)} f$ for all $n \in \mathbb{N}$, and such that whenever $R \overset{p}{\underset{p'}{\rightrightarrows}} IX^@$ satisfies $pX^n \cdot \mu_I^{(n)} \cdot f = p'X^n \cdot \mu_I^{(n)} \cdot f$ for all $n \in \mathbb{N}$ (see the square below), there exists a unique morphism $\psi : R \to E_f$ rendering the triangles shown below commutative.

(6)



Note that, if such an $E_f \overset{\alpha}{\underset{\gamma}{\rightrightarrows}} IX^@$ exists, then the uniqueness condition of ψ of (6) ensures that it is unique up to isomorphism; for if $R \overset{p}{\underset{p'}{\rightrightarrows}} IX^@$ is another candidate, the ψ from (6) and the ψ' obtained by interchanging the roles of R and E_f must satisfy $\psi\psi' = \mathrm{id}_R$ and $\psi'\psi = \mathrm{id}_{E_f}$, and hence are isomorphisms.

5.3. Motivation. Having defined the Nerode equivalence E_f, our next step in the classical theory was to form the minimal state-space as

$$Q_f = X_0^*/E_f = \{[w]_f | w \in X_0^*\}.$$

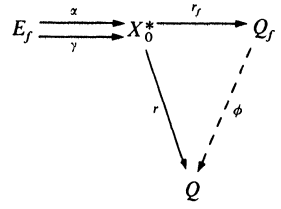
Let us see if we can define the map $X_0^* \xrightarrow{r_f} Q_f$ in some way other than by the specification $w \mapsto [w]$ of elements. We have that

$$w_1 r_f = w_2 r_f \Leftrightarrow [w_1] = [w_2]$$
$$\Leftrightarrow (w_1, w_2) \in E_f.$$

Suppose, then, that $r : X_0^* \to Q$ is any map with the property that $w_1 r = w_2 r$ whenever $(w_1, w_2) \in E_f$. It is clear that we can then define a map $Q_f \to Q$ by $[w]\phi = wr$, since $[w_1] = [w_2]$ implies $w_1 r = w_2 r$. In other words, we have that $\alpha r_f = \gamma r_f$ (where, recall, $\alpha, \gamma : E_f \to X_0^*$ are the restricted coordinate projections),

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

while for all $X_0^* \xrightarrow{r} Q$ which satisfy $\alpha r = \gamma r$, we have that there exists exactly one ϕ such that the following diagram commutes.

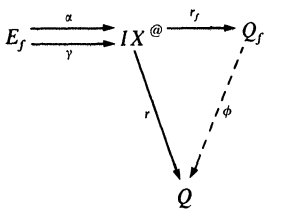


In category theoretic terms, we say r_f is the *coequalizer* of α and γ, and write $r_f = \text{coeq}(\alpha, \gamma)$. (It is easy to verify that the diagram determines ϕ uniquely up to isomorphism.) This motivates the second postulate of our general theory. The fourth postulate is a technical one to make our general proof work, and reduces in the classical case to the obvious fact that if $h: A \to B$ is onto in $\mathfrak{S}$, then so is $h \times \text{id}_{X_0}: A \times X_0 \to B \times X_0$. Postulate 3 is a "cop-out" for which we shall atone in § 5.5 below.

5.4. Theory. Let $E_f \overset{\alpha}{\underset{\gamma}{\rightrightarrows}} IX^@$ be the diagram whereby an $f: IX^@ \to Y$ satisfies Postulate 1.

Postulate 2. We say that f also satisfies Postulate 2 if E_f has a coequalizer, and we then define $IX^@ \xrightarrow{r_f} Q_f$ to be $\text{coeq}(\alpha, \gamma)$; i.e., we have that $\alpha r_f = \gamma r_f$ while for all $IX^@ \xrightarrow{r} Q$ which satisfy $\alpha r = \gamma r$, we have that there exists exactly one ϕ such that the following diagram commutes.

(7)



Postulate 3. We say f satisfies Postulate 3 if in addition to Postulates 1 and 2, we have that there exists $\delta_f: Q_f X \to Q_f$ making $(IX^@, \mu_I) \xrightarrow{r_f} (Q_f, \delta_f)$ a dynamorphism.

Postulate 4. We say f satisfies Postulate 4 if whenever M is a realization of f whose reachability map r is a coequalizer, rX is epimorphic.

We say that a map c is a *coequalizer* if $c = \text{coeq}(a, b)$ for some (a, b).

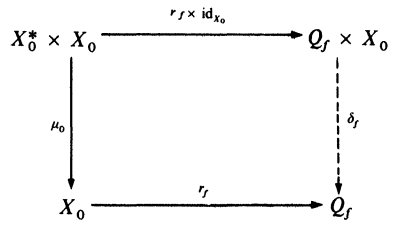
Let $\mathfrak{E}$ be the class of all coequalizers. Observe that every split epimorphism is in $\mathfrak{E}$ (if $A \xrightarrow{d} B \xrightarrow{s} A = \text{id}_A$, then $s = \text{coeq}(\text{id}_B, sd)$). Also, if c is a coequalizer, then c is epimorphic (if $ct = cu$, then t and u are both the unique map induced by a universal property). In the category of monoids, $X \xrightarrow{f} Y$ is a coequalizer if and only if f is a monoid homomorphism onto.

5.5. Motivation. In the classic theory, we define

$$\delta_f: Q_f \times X_0 \to Q_f: ([w], x) \mapsto [wx],$$

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

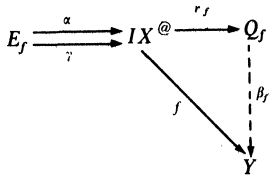
which is the unique map which completes the diagram below.



We define $\beta_f : Q_f \longrightarrow Y : [w] \longmapsto f(w)$.

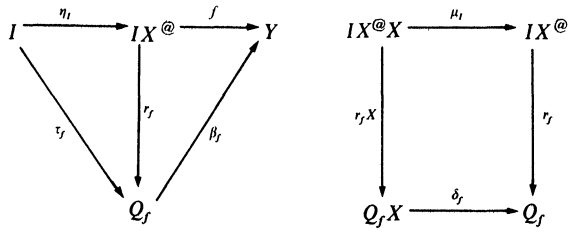
5.6. Theory. Postulate 3 provides $Q_f X \xrightarrow{\delta_f} Q_f$. We may define β_f by

(8)



because $\alpha f = \gamma f$ by (6) for $n = 0$.

If we define τ_f to be $\eta_I r_f$ we have the diagrams below,



so that r_f, a priori only defined as the coequalizer of $\alpha, \gamma : E_f \longrightarrow IX^@$, is indeed the reachability map of

$$M_f = (X, Q_f, \delta_f, I, \tau_f, Y, \beta_f).$$

5.7. Motivation. In the classical theory, we show that M_f is minimal by showing that if $M = (X_0, Q, \delta, q_0, Y, \beta)$ is any other realization of f which is reachable (i.e., $\delta^*(q_0, X_0^*) = Q$), then the map $Q \longrightarrow Q_f : \delta^*(q_0, w) \longmapsto [w]$ is indeed well-defined and a homomorphism. The proof is no harder in the general case, except that a little care is required to handle the reachability condition in the form that the reachability map of M is a coequalizer.

5.8. Theory. Turning to theory, we now prove the main theorem of this paper.

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

NERODE REALIZATION THEOREM. *Let $\mathscr{K}$ be an arbitrary category, and let X be an input process in $\mathscr{K}$. Fix an initial state object I and an output object Y, and let $f: IX^{@} \to Y$ be a response morphism. Assume that f satisfies the four postulates discussed earlier, specifically,*

Postulate 1. *The Nerode equivalence $E_f \overset{\alpha}{\underset{\gamma}{\rightrightarrows}} IX^{@}$ of f exists.*

Postulate 2. *$r_f: IX^{@} \to Q_f = \operatorname{coeq}(\alpha, \gamma)$ exists in $\mathscr{K}$.*

Postulate 3. *There exists a $\mathscr{K}$-morphism $\delta_f: Q_f X \to Q_f$ with respect to which $r_f: (IX^{@}, \mu_I) \to (Q_f, \delta_f)$ is an X-dynamorphism.*

Postulate 4. *Given an X-machine M which realizes f, if the reachability map r of M is a coequalizer in $\mathscr{K}$, then rX is epimorphic in $\mathscr{K}$.*

Then there exists a unique morphism $\beta_f: Q_f \to Y$ such that $r_f \cdot \beta_f = f$, and

$$M_f = (X, Q_f, \delta_f, I, \tau_f = \eta_I \cdot r_f, Y, \beta_f)$$

is the minimal coequalizer-reachable realization of f (i.e., is a terminal object in $\mathscr{E}_f$, where $\mathscr{E}$ is the class of all coequalizers in $\mathscr{K}$).

Proof. That M_f realizes f is just (8): $f = r_f \cdot \beta_f$, while the reachability map $IX^{@} \xrightarrow{r_f} Q_f$ is a coequalizer, by definition. It only remains to prove that M_f is minimal.

Suppose that $M = (X, Q, \delta, I, \tau, Y, \beta)$ is a coequalizer-reachable realization of f, so that the r defined by

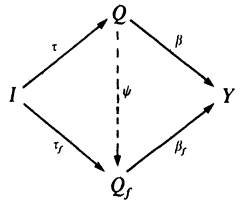
$$\eta_I r = \tau, \quad rX \cdot \delta = \mu_I r$$

satisfies a coequalizer diagram

$$E \overset{\alpha'}{\underset{\gamma'}{\rightrightarrows}} IX^{@} \xrightarrow{r} Q$$

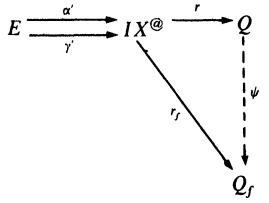
as well as the equality $f = r\beta$.

We have to prove that there exists a unique dynamorphism $\psi: M \to M_f$ such that the diagram below commutes.



To be a dynamorphism, ψ must certainly satisfy $r\psi = r_f$. By hypothesis, there exists (α', γ') such that $r = \operatorname{coeq}(\alpha', \gamma')$. The obvious strategy is thus to exploit this coequalizer property, and take the unique ψ which yields commutativity in the following diagram,

(9)



Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

but to do this we must first verify that $\alpha' r_f = \gamma' r_f$. Now $\mu_I^{(n)} f = rX^n \cdot \delta^{(n)}\beta$ (as can easily be seen by induction). Thus

$$\begin{aligned}\alpha' X^n \cdot \mu_I^{(n)} f &= (\alpha' r) X^n \cdot \delta^{(n)}\beta \\ &= (\gamma' r) X^n \cdot \delta^{(n)}\beta \\ &= \gamma' X^n \cdot \mu_I^{(n)} f.\end{aligned}$$

Thus, by an application of (6), there is a unique ψ' such that

$$\alpha' = \psi'\alpha \quad \text{and} \quad \gamma' = \psi'\gamma,$$

and so $\alpha' r_f = \psi'\alpha r_f = \psi'\gamma r_f = \gamma' r_f$, as required to obtain ψ uniquely from (9). We have to check

$$\tau\psi = \tau_f; \quad \psi\beta_f = \beta; \quad \text{and} \quad \psi X \cdot \delta_f = \delta\psi.$$

First,

$$\tau\psi = \eta_I r\psi = \eta_I r_f = \tau_f.$$

Second,

$$r\psi\beta_f = r_f\beta_f = f = r\beta,$$

whence $\psi\beta_f = \beta$, since r is a coequalizer, and hence epimorphic. Finally,

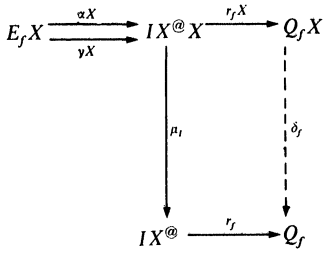
$$rX \cdot \psi X \cdot \delta_f = r_f X \cdot \delta_f = \mu_I r_f = \mu_I r\psi = rX \cdot \delta\psi$$

whence $\psi X \cdot \delta_f = \delta\psi$ since, by Postulate 4, rX is epimorphic. □

5.9. To demonstrate the power of this theorem, we recover the Arbib–Zeiger M_f [9, § 4] for linear systems and the Give'on–Arbib M_f [23, § 6.8] for tree automata. We first make an observation concerning Postulates 3 and 4.

LEMMA. *If f satisfies Postulates* 1 *and* 2 *and if X preserves coequalizer diagrams (that is, cX = coeq (aX, bX) given c = coeq (a, b)) then f satisfies Postulates* 3 *and* 4.

Proof. Postulate 4 is clear since coequalizers are epimorphic. Since r_f = coeq (α, γ) by definition, our hypothesis on X yields that $r_f X$ = coeq $(\alpha X, \gamma X)$.



For Postulate 3, it is then sufficient to prove that $\alpha X \cdot \mu_I \cdot r_f = \gamma X \cdot \mu_I \cdot r_f$, and this is always true as follows. Applying (6) above to $p = \alpha X \cdot \mu_I$, $p' = \gamma X \cdot \mu_I$, one easily computes that

$$pX^n \cdot \mu_I^{(n)} f = \alpha X^{n+1} \cdot \mu_I^{(n+1)} f = p' X^n \cdot \mu_I^{(n)} f,$$

so that there exists unique ψ with $\alpha X \cdot \mu_I = \psi\alpha$ and $\gamma X \cdot \mu_I = \psi\gamma$. Since $\alpha r_f = \gamma r_f$, we are done. □

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

While it is often a lot to ask of an input process that it preserve coequalizer diagrams to even the extent required in the previous lemma, one checks easily that $- \times X_0$ in $\mathfrak{S}$ and $- \oplus X_0$ in R-modules preserve all coequalizer diagrams. It is also true that in both $\mathfrak{S}$ and R-modules, any pair $V \overset{a}{\underset{b}{\rightrightarrows}} W$ has a coequalizer. (In $\mathfrak{S}$, divide out by the equivalence relation generated by $\{(va, vb): v \in V\}$; in R-modules, divide out by the image of $a - b$.) In particular, to apply the Nerode realization theorem to a linear map $X_0^{\S} \xrightarrow{f} Y$, we need only check that f satisfies Postulate 1. To this end, we report (as is proved quite generally in [5]) that for arbitrary maps $p, p': R \rightrightarrows IX^{@} \cdot pX^n \cdot \mu_I^{(n)} f = p'X^n \cdot \mu_I^{(n)} f$ (as in (6) above) if and only if $pz^n f = p'z^n f$, where $X_0^{\S} \xrightarrow{z} X_0^{\S}$ was defined in § 4. It is therefore clear that

$$\{E_f = (w, w'): wz^n f = w'z^n f \quad \text{for all } n \geqq 0\} \subset X_0^{\S} \times X_0^{\S},$$

α, γ being the restriction of the projections.

Without going into too much detail, we sketch Postulate 1 in the tree automata case. Think of $QX^{@}$ as the set of all derived Ω-operations with variables in Q [18, § III.7]. QX^n consists of some of the derived operations of tree depth n (e.g., if $\Omega = \{+, \cdot\}$ and $x, y, a, b \in Q$, then $xy + ab + \cdot$ is in QX^2 and $xy + a \cdot$ is in $(QX \cup Q)X$, but is not in QX^2). $\varnothing X^{@}$ is the set of all derived nullary operations (e.g., if $\Omega = \{1, +\}$, then $1, 11+$ and $111++$ are elements of $\varnothing X^{@}$). If $Q \subset \varnothing X^{@} \times \varnothing X^{@}$ with projections $\alpha, \gamma: Q \rightrightarrows \varnothing X^{@}$, then

$$\alpha X^n \cdot \mu_I^{(n)} \cdot f = \gamma X^n \cdot \mu_I^{(n)} \cdot f$$

as in (6) above occurs if and only if for each derived operation $(q_1 \cdots q_n)\Delta \in QX^n$, we have

$$(a_1, b_1), \cdots, (a_m, b_m) \in Q \Rightarrow (a_1 \cdots a_m)\Delta f = (b_1 \cdots b_m)\Delta f. \tag{10}$$

If (10) holds for all choices of (a_1, b_1), Δ and n, then Q is a congruence since all basic Ω-operations are in QX; in particular, (10) holds whenever Δ is any derived operation, for instance, a unary one. Conversely, capitalizing on the fact that elements of $\varnothing X^{@}$ are nullary operations, we see that in the context of (10), the elementary translation $(a_1, \cdots, a_{k-1}, q_k, a_{k+1}, \cdots, a_m)\Delta$ is in fact a derived unary operation Δ'. By using m suitable Δ''s for each Δ, we have that

$$E_f = \{(a, b) \in \varnothing X^{@}: a\Delta f = b\Delta f \text{ for all derived unary operations } \Delta\}$$

satisfies Postulate 1 and is a congruence. Postulates 2 and 3 are now standard universal algebra [18, § II.3.5]. Postulate 4 is trivial since in $\mathfrak{S}$, coequalizers are split epimorphic and all functors preserve split epimorphisms.

Thus our general result does indeed show that only extremely weak conditions are required on the input process X which determines a class of machines in a category for the corresponding response functions $IX^{@} \longrightarrow Y$ to have a Nerode-style minimal realization theory.

With this, we complete our introductory exposition of our theory of machines in an arbitrary category. The theory is developing rapidly in a number of directions. In [6] we classify "state-behavior" machines (see also the related work of Bainbridge [10]) which exclude tree and stochastic automata but include the closed category machines of Goguen [24]. State-behavior machines in a given

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

category transform into state-behavior machines in the dual category, giving rise to a new and interesting duality theory (a technique not available to Goguen because the dual of a closed category is not always closed). An especially simple class of state-behavior machines (those for which X is the identity functor) is isolated for study in [5], the key technique being based on Lawvere's concept of simple recursion [31]. The duality theory of these "decomposable" machines generalizes Kalman's module-theoretic approach to linear systems ([26], [27, Chap. 10]) to machines in almost any category. In fact, we present the startling thesis that linearity has little to do with linear machines, and that certain peculiarities of the category of real vector spaces have acted to obscure the conventional view of the way linear machines run!

The intuition behind our view of dynamics as a morphism $QX \longrightarrow Q$ has been that "all machines are sequential and deterministic." Among the four examples of the first section, stochastic automata are usually viewed as "nondeterministic." A precise formulation of "state-blurring" is presented in [7], which identifies nondeterministic machines in the original (arbitrary) category with deterministic machines in an associated "blurred" category.

The nature of our approach ultimately introduces algebraic semantics [20], [35] into automata theory. The fundamental concepts of [7] are, in fact, the "monads" or "triples" of algebraic semantics [32, Chap. VI] and the distributive laws of [12]. In an entirely different way, monads first appeared in the automata theory literature as the tree automata generalization of the monoids usually associated with sequential machines [22]. Since Barr [11, § 5] has shown that our free machines (the $X^{@}$ of § 4) are often coextensive with free monads, we may also cite the recent results of Dubuc [19], which explicate how free monads are constructed from their generators. By exploiting the monad transformations already in use in algebraic semantics, Alagić [1] has developed an elegant theory of state transformation for generalized machines in an arbitrary category, in particular recapturing and simplifying the generalized[2] sequential machines of Thatcher [45].

Given this vigorous growth, we may expect the study of machines in a category both to feed back into the study of algebraic structure per se and also to have repercussions in many phases of the computer, information and system sciences from programming language studies to control theory.

REFERENCES

[1] S. ALAGIĆ, *Natural state transformations*, COINS Tech. Rep. 73B-2, Univ. of Mass., Amherst, 1973.

[2] M. A. ARBIB, *Automatia theory and control theory: A rapprochement*, Automatica, 3 (1966), pp. 161–189.

[3] ———, *Theories of Abstract Automata*, Prentice-Hall, Englewood Cliffs, N.J., 1969.

[4] ———, *Coproducts and decomposable machines*, J. Comput. Syst. Sci., 7 (1973), pp. 278–287.

[5] M. A. ARBIB AND E. G. MANES, *Foundation of system theory*, Automatica–J. IFAC, May 1974.

[6] ———, *Adjoint machines, state-behavior machines and duality*, COINS Tech. Rep. 73B-1, Univ. of Mass., Amherst, 1973.

[7] ———, *Kleisli Machines*, to appear.

[8] ———, *Category Theory Without Tears*, to appear.

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

[9] M. A. ARBIB AND H. P. ZEIGER, *On the relevance of abstract algebra to control theory*, Automatica–J. IFAC, (1969), pp. 589–606.

[10] E. S. BAINBRIDGE, *A unified minimal realization theory with duality*, Tech. Rep., Computer and Communication Sci. Dept., Univ. of Mich., Ann Arbor, 1972.

[11] M. BARR, *Coequalizers and free triples*, Math., Z., 116 (1970), pp. 307–322.

[12] J. BECK, *Distributive laws*, Lecture Notes in Mathematics, Springer-Verlag, New York, 80 (1969), pp. 119–140.

[13] L. S. BOBROW AND M. A. ARBIB, *Discrete Mathematics: Applied Algebra for Computer and Information Science*, Saunders, Philadelphia, 1974.

[14] R. BROCKETT AND A. H. WILLSKY, *Finite-state group-homomorphic sequential machines*, IEEE Trans. Automatic Control, AC-17 (1972), pp. 483–490.

[15] J. R. BÜCHI AND J. B. WRIGHT, *Mathematical Theory of automata*, course notes of Communications Sciences 403, Univ. of Mich., Ann Arbor, 1960.

[16] J. CARLYLE, *On the external probability structure of finite state channels*, Information and Control, 7 (1964), pp. 385–397.

[17] N. CHOMSKY, *On certain formal properties of grammar*, Ibid., 2 (1959), pp. 137–167.

[18] P. M. COHN, *Universal Algebra*, Harper and Row, New York, 1965.

[19] E. DUBUC, *Free Monoids*, J. Algebra, to appear.

[20] B. ECKMANN, ed., *Seminar on triples and categorical homology theory*, Lecture Notes in Mathematics, Springer-Verlag, New York, 80 (1969).

[21] H. EHRIG, H.-J. KREOWSKI, W. KUHNEL AND K. D. KIERMEIER, *Axiomatic theory of systems and systematics*, to appear.

[22] S. EILENBERG AND J. B. WRIGHT, *Automata in general algebras*, Information and Control, 11 (1967), pp. 452–470.

[23] Y. GIVE'ON AND M. A. ARBIB, *Algebra automata II: The categorical framework for dynamic analysis*, Ibid., 12 (1968), pp. 346–370.

[24] J. A. GOGUEN, *Minimal realization of machines in closed categories*, Bull. Amer. Math. Soc., 78 (1972), pp. 777–783.

[25] B. L. HO AND R. E. KALMAN, *Effective construction of linear state-variable models from input/output functions*, Regelungstech. Prozess–Datenverarbeit., 14 (1968), pp. 545–548.

[26] R. E. KALMAN, *Algebraic structure of linear dynamical systems. I, the module of* Σ. Proc. Nat. Acad. Sci. U.S.A., 54 (1965), pp. 1503–1508.

[27] R. E. KALMAN, P. L. FALB AND M. A. ARBIB, *Topics in Mathematical System Theory*, McGraw-Hill, New York, 1969.

[28] H. KLEISLI, *Every standard construction is induced by a pair of adjoint functors*, Proc. Amer. Math. Soc., 16 (1965), pp. 544–546 ; for an exposition, see [32, Thm. 1, p. 143].

[29] F. W. LAWVERE, *Functorial semantics of algebraic theories*, Doctoral dissertation, Columbia Univ., New York, 1963.

[30] ———, *An elementary theory of the category of sets*, mimeographed notes, Univ. of Chicago; condensed version in Proc. Nat. Acad. of Sci. U.S.A., 52 (1964), pp. 1506–1511.

[31] ———, *Adjointness in Foundations*, Dialectica, 23 (1969), pp. 281–296.

[32] S. MAC LANE, *Categories for the Working Mathematician*, Springer-Verlag, New York, 1972.

[33] S. MAC LANE AND G. BIRKHOFF, *Algebra*, Macmillan, New York, 1967.

[34] E. G. MANES, *A triple miscellany: Some aspects of the theory of algebras over a triple*, Wesleyan Univ., 1967.

[35] ———, *Algebraic Theories*, Springer-Verlag, New York, to appear.

[36] W. S. MCCULLOCH AND W. H. PITTS, *A logical calculus of the ideas immanent in nervous activity*, Bull. Math. Biophys., 5 (1943), pp. 115–133.

[37] E. F. MOORE, *Gedanken experiments on sequential machines*, Automata Studies, Princeton Univ. Press, Princeton, N.J., pp. 129–153.

[38] J. MYHILL, *Finite automata and the representation of events*, WADD Tech. Note 57–624, Wright-Patterson AFB, Ohio, 1957.

[39] A. NERODE, *Linear automaton transformations*, Proc. Amer. Math. Soc., 9 (1958), pp. 541–544.

[39a] L. PADULO AND M. A. ARBIB, *System Theory: A Unified State-Space Approach to Continuous and Discrete Systems*, Saunders, Philadelphia, 1974.

[40] A. PAZ, *Introduction to Probabilistic Automata*, Academic Press, New York, 1971.
[41] M. O. RABIN AND D. SCOTT, *Finite automata and their decision problems*, IBM J. Res. Develop., 3 (1959), pp. 114–125.
[42] E. C. ROUNDS, *Mappings and grammars on trees*, Math. Syst. Theory, 4 (1970), pp. 257–287.
[43] C. E. SHANNON, *A mathematical theory of communication*, Bell System Tech. J., 27 (1948), pp. 379–423 and pp. 623–656.
[44] J. W. THATCHER, *Transformations and translations for the point of view of generalized finite automata theory*, Proc. ACM Symp. Theory of Computing, 1969, pp. 129–142.
[45] ———, *Generalized² Sequential Machine Maps*, J. Comput. Syst. Sci., 4(1970), pp. 339–367.
[46] J. W. THATCHER AND J. B. WRIGHT, *Generalized finite automata theory*, Math. Systems Theory, 2 (1968), pp. 57–81.
[47] A. M. TURING, *On computable numbers with an application to the Entscheidungsproblem*, Proc. London Math. Soc. Ser. 2, 42 (1936), pp. 230–265 and 43 (1936–37), pp. 544–546.
[48] L. ZADEH AND C. A. DESOER, *Linear System Theory*, McGraw-Hill, New York, 1963.

Downloaded 11/28/14 to 129.120.242.61. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php