BULLETIN OF
MATHEMATICAL BIOPHYSICS
VOLUME 20, 1958

THE REPRESENTATION OF BIOLOGICAL SYSTEMS FROM THE STANDPOINT OF THE THEORY OF CATEGORIES

ROBERT ROSEN
COMMITTEE ON MATHEMATICAL BIOLOGY
THE UNIVERSITY OF CHICAGO

A mathematical framework for a rigorous theory of general systems is constructed, using the notions of the theory of Categories and Functors introduced by Eilenberg and MacLane (1945, *Trans. Am. Math. Soc.*, **58**, 231-94). A short discussion of the basic ideas is given, and their possible application to the theory of biological systems is discussed. On the basis of these considerations, a number of results are proved, including the possibility of selecting a unique representative (a "canonical form") from a family of mathematical objects, all of which represent the same system. As an example, the representation of the neural net and the finite automaton is constructed in terms of our general theory.

I. *Introduction.* In a previous paper (Rosen, 1958), we attempted to investigate some of the aspects of a general theory of biological systems and to point out several of the possible applications which such a theory might have. We remarked at that time that although our treatment was an intuitively reasonable one, it was not yet of sufficient scope to provide the foundation for a general theory. It is the purpose of the present paper to remove the ambiguities contained in our earlier approach and to show how a precise mathematical theory of systems may be constructed.

Before we undertake the construction of our general theory, it may be of value to point out some of the difficulties which arise in an attempt to formalize our earlier treatment. We recall that an arbitrary system might be decomposed into a collection of smaller objects called *components* and that these components could be arranged in an oriented graph (the *block diagram*) as follows: two components M_i, M_j are to be connected by an oriented edge $M_i \rightarrow M_j$ if an output of M_i is an input to M_j. However, such an oriented graph is unable to provide an adequate representation for many situations and may hence lead to a distorted picture of the actual input-output relations obtaining between the components of a system. For example, the number of *distinct* outputs produced by a component M_i is not necessarily the same as the number of oriented edges for which M_i is origin in the block diagram, since more than one component may receive the same input from M_i. This will be the case, e.g., if M_i were an endocrine gland that produces a hormone which affects several different organs. On the other hand, a component M_i might provide a component M_j with more than one distinct output, although the block diagram will only indicate one oriented edge connecting M_i and M_j. To illustrate this, we may once again take M_i to be an endocrine gland, such as the pituitary, which provides a number of different hormones to the same organ.

Further, it is necessary for formal purposes to adjoin to the block diagram a formal vertex which represents the environment. This vertex must be connected as origin to all components which receive environmental inputs and as terminus to all components which produce environmental outputs. Since this vertex is not itself a component, its introduction necessitates a troublesome and artificial attention to special cases in many arguments.

Even more important than the foregoing is the necessity of having an adequate means for discussing the behavior of the various time lags which occur in the operation of a biological system. We remarked in our earlier paper (*loc. cit.*) that the operation lags of the various components of a system will depend in general upon the particular inputs with which the component is supplied. There is no obvious and easy means of accomplishing such a representation within the framework of a pure graph theory. Likewise, the requirement that a component producing several distinct outputs will in general possess a different lag at each output is slightly troublesome.

Although we may to a certain extent overcome the difficulties we have mentioned by the introduction of a number of technical devices, the theory which results will have lost the intuitional clarity which constituted a large part of its appeal. It is therefore evident that a suitable representation theory of biological systems will require an entirely different point of view than the one adopted above and will utilize a new set of mathematical tools. The representation theory which we present below will be seen to fulfill these conditions. The mathematical tools which we find appropriate are contained in the comparatively recent theory of categories and functors (Eilenberg and MacLane, 1945).

Since this theory is largely unfamiliar to non-mathematicians and since the emphasis which is placed on the theory in the mathematical literature is quite different from the one which we will require, we enter into a brief discussion of its basic notions before we construct our representation. It will be seen that, although the theory which results seems at the outset to be considerably more complicated than our previous treatment, we can formulate our results, and even our definitions, in a simpler, more intelligible and more precise fashion than is possible through any refinement of our other approach.

II. *The Formal Representation.* Throughout our previous discussion, we have regarded a component of a system M as a "black box," capable of receiving a number of inputs and of emitting a number of outputs. We observe that this behavior implies that each component acts *selectively* upon its environment at each input; that is, each input to a component must belong to a fixed set of *admissible* inputs in order to be accepted by the component. Likewise, each output of a component is an element of a fixed set of admissible outputs. Both of these sets are completely determined by the component itself. Thus, for example, an ordinary audio-amplifier will accept as input only the elements of a certain set of electric currents and will produce as output another element in this set. Likewise, only the elements of a definite set of chemical compounds can serve as inputs to a given enzyme; this set is termed the set of *substrates* of the enzyme; the outputs of the enzyme will similarly belong to fixed sets of chemical compounds.

With these ideas in mind, let us attempt to find a suitable representation for the simplest type of component. We consider a component M which receives a number m of inputs and emits a single output. In a block diagram, M would be represented as a vertex which serves as the terminus for m directed edges $\rho_1, \ldots, \rho_m$ and as origin for a single arrow ρ. According to our discussion above, to each arrow ρ_i in the block diagram there corresponds a definite set, the elements of which are capable of serving as inputs to M. Let us designate the set corresponding to ρ_i by A_i for each i such that $1 \leqslant i \leqslant m$, and the set corresponding to ρ by B. Then we can regard the effect of M as a *mapping* or transformation, which assigns to every m-tuple $(a_1, \ldots, a_m)$, $a_i \epsilon A_i$, a definite object $b \epsilon B$. In formal terminology, we assign to the component M a mapping f, where

$$f: A_1 \times A_2 \times \ldots \times A_m \rightarrow B.$$

In words, M is to be represented by a mapping f, the domain of which is the "cartesian product" of the sets of admissible inputs to M, and the range of which is the class of admissible outputs of M.

In the general case, where we allow the component M to emit $n > 1$ outputs, we must in general assign a mapping f_k to each output ρ_k of M. Thus, if B_k is the set of admissible output objects produced by M at ρ_k, we write

$$f_k: A_1 \times A_2 \times \ldots \times A_m \rightarrow B_k.$$

Hence a general component is to be represented by an n-tuple of mappings, where n is the number of outputs of the component. These mappings all possess a common domain; namely, the set $A_1 \times A_2 \times \ldots \times A_m$, but their ranges will differ, in general.

These simple remarks provide the main conceptual basis for the ensuing discussion; the remainder of the paper is largely an elaboration and formalization of these remarks. We may observe that regarding a biological system as a set of mappings incorporates most of our intuitive notions about these systems in an extremely natural way. Further, we see that most of the difficulties mentioned in the Introduction have disappeared. Thus, if a component M provides the same input to more than one component, then this merely means that some of the mappings which represent M are the same. Likewise, it is seen that no difficulty can be caused by a component M providing a second component with more than one distinct input.

With this background in mind, let us proceed to introduce the mathematical tools which will provide the formal basis for our representation. We shall proceed axiomatically. The above discussion, together with some elementary examples which we introduce to fix the ideas, should serve to motivate our presentation.

The first notion which we introduce is that of a *category*. Explicitly, a category consists of the following data:

1. A collection of *objects*, which we shall designate by A, A', ...

2. A function assigning to each pair (A, A') of objects in the category, a set denoted by $H(A, A')$, the elements of which are called *mappings* or *transformations*. This set may be empty for some pairs (A, A').

If $f \epsilon H(A, A')$, then we shall call the object A the *domain* of f; the object A' will be called the *range* of f.

The data above represent the irreducible minimum which must be given in order to construct a theory of a set of mappings; namely, we must be given both the mathematical objects on which the mappings act and the mappings themselves. We must now introduce a means of combining, or *composing*, the mappings which we are given. Hence, we further require of a category:

3. There exists a function (called *composition*) which assigns to pairs (f, g) of mappings such that $f \epsilon H(A, A')$, $g \epsilon H(A', A'')$, a mapping (denoted by gf) in the set $H(A, A'')$.

The composition operation may perhaps be more clearly represented by means of a *diagram* of mappings of the type shown below:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "gf"'] & A' \arrow[d, "g"] \\ & A'' \end{tikzcd}$$

We shall have occasion to make further use of diagrams of this type throughout the ensuing discussion. A diagram such as the above, in which any two directed paths of mappings with the same origin and terminus yield the same resultant, is called a *commutative* diagram.

As an example of a category, we may consider the following: let the objects of the category consist of all groups, with the mappings of each set $H(A, A')$ being the group homomorphisms of A *onto* A'. Or, we may take for the objects of the category the totality of all topological spaces, with the mappings of each $H(A, A')$ taken to be continuous mappings. For many other examples of such structures, we refer the reader to the paper of Eilenberg and MacLane (1945).

It will be seen from these concrete examples that the data 1–3 above must be subjected to certain axioms in order that our abstract categories may enjoy properties suggested by these examples. Thus, we shall require the following:

Cat. 1: $H(A, A') \cap H(A_1, A_1') = \emptyset$
whenever $A \neq A_1$ and $A' \neq A_1'$.

Here $\emptyset$ is the empty set. This axiom merely states that any mapping has exactly one domain and exactly one range.

Cat. 2: If $f \epsilon H(A, A')$, $g \epsilon H(A', A'')$, and $h \epsilon H(A'', A''')$, then $h(gf) = (hg)f$.

This axiom states that the composition of mappings satisfies the associative law.

Cat. 3: For each object A in the category, there exists a mapping $i_A \epsilon H(A, A)$, such that if A' is any other object and $f \epsilon H(A, A')$, then $fi_A = f$; for any mapping $g \epsilon H(A', A)$ we have $i_A g = g$.

The mapping i_A is called the *identity map* of the object A. We may remark at this point that since there exists by (Cat. 3) a one-to-one correspondence between the objects of a category and their identity maps and since by (Cat. 1) each mapping of the category uniquely determines its domain and range, we could have dispensed with the objects entirely and defined the category solely in terms of its mappings. We shall find that this is the generalization of a remark made earlier with reference to biological systems.

Let us now denote by $\mathbf{A}$ an arbitrary category. We define a *subcategory* $\mathbf{A}_0$ of $\mathbf{A}$ to be a collection of objects and mappings of $\mathbf{A}$ satisfying the following conditions:

Sub. 1: If $A \epsilon \mathbf{A}_0$, then $i_A \epsilon \mathbf{A}_0$.

Sub. 2: If f and g are mappings in $\mathbf{A}_0$ such that the composition gf is defined, then $gf \epsilon \mathbf{A}_0$.

Sub. 3: If f is a mapping in $\mathbf{A}_0$, then the domain of f and the range of f are objects in $\mathbf{A}_0$.

It is readily verified that the objects and mappings of a subcategory themselves comprise a category.

We now introduce the notion of *functor*. In general, whenever a mathematical structure has been defined axiomatically, we desire a means of comparing, in some sense, different objects which bear the structure in question. Most frequently, we proceed by constructing various structure-preserving mappings between such objects. For example, groups may be compared with one another by means of homomorphisms, and topological spaces by means of continuous mappings. Categories, then, may be compared by means of some generalized type of mapping; such a mapping is called a *functor*. We now proceed to give a precise definition.

Let **A** and **B** be categories. A *covariant functor* T: **A** $\to$ **B** is a pair of mappings (the two denoted by T, the first of which assigns to each object $A \in$ **A** an object, which we shall write as $T(A)$, in **B**; and the second of which assigns, to each mapping $f \in H(A, A')$ in **A**, a mapping which will be written as $T(f) \in$ **B**, $T(f) \in H[T(A), T(A')]$. We shall require the following conditions to be satisfied:

Fun. 1: $T(gf) = T(g)\,T(f)$ whenever gf is defined.

Fun. 2: $T(i_A) = i_{T(A)}$ for each $A \in$ **A**.

We remark that it is possible to define *contravariant functors*, which behave precisely as do covariant functors, except for the fact that they reverse the directions of the mappings. Throughout the following discussion we shall deal exclusively with covariant functors which we shall refer to simply as *functors*.

A functor T is called *faithful* if the following conditions are fulfilled:

Fid. 1: If f and g belong to $H(A, A')$ and $T(f) = T(g)$, then $f = g$.

Fid. 2: If $f \in H(A, A')$ and $g \in H(A', A)$ are mappings such that $gf = i_A$; and if $T(A) = T(A')$, $T(f) = i_{T(A)}$; then $A = A'$.

Mappings f, g which satisfy the conditions mentioned in the postulate (Fid. 2) are called *equivalences* of the objects A, A'; in the case where **A** is a category of groups, its equivalences are isomorphisms; in a category of topological spaces, the equivalences are homeomorphisms.

It is easily verified that the image of a category **A** under a functor T: **A** $\to$ **B** is a subcategory of **B**. If T is faithful, then this image, which we may denote symbolically as T(**A**), is abstractly indistinguishable from **A**, and we may in this fashion regard **A** as being *embedded* by T as a subcategory of **B**.

With this background, we may now state a theorem which will be of great importance to our theory of representation of systems:

Theorem 1. Any abstract category **A** can be embedded as a subcategory of the category **S**, the objects of which consist of all sets and the mappings of which are the totality of all set-theoretical many-one mappings of sets.

Proof. The proof of this theorem is given in the paper of Eilenberg and MacLane (1945) and consists in the explicit construction of a faithful functor from **A** to **S**. We refer the reader to the original paper for the details.

As a consequence of Theorem 1, we may by choosing a *definite* faithful functor T: **A** $\longrightarrow$ **S**, regard the objects of an arbitrary category **A** as ordinary sets and the mappings of **A** as ordinary set-theoretical mappings. Hence in this manner the usual set-theoretical terminology will make sense in an arbitrary category. In particular, we may speak of unions, intersections, inclusions, and cartesian products of objects in a category. We shall have more to say about this aspect in Section IV below; we merely observe here that in general an arbitrary category will not be *closed* under these set-theoretical operations. For our purposes, however, we shall require from the outset that the categories with which we deal shall be closed under the formation of cartesian products.

This completes our digression on the mathematical tools which we shall utilize in the representation of biological systems; we now turn to the construction of the representation itself. Our point of departure is the observation made earlier that, given an arbitrary category **A**, we can form *oriented graphs* (called *diagrams* on p. 317) by selecting a collection or objects $\{A_i\}$ in the category and a collection of mappings from the appropriate sets $H(A_i, A_j)$. Two objects in the diagram will be connected by an oriented edge if and only if some mapping of the collection has one of the objects as range and the other as domain. For our purposes, however, we shall require a slightly more general type of diagram in which the objects of the diagram merely *contain* the domains and ranges of the mappings.

Let us agree to denote by $d(f)$ and $r(f)$ the domain and range, respectively, of an arbitrary mapping in a category. Then we shall say that a collection of objects $\{A_i\}$ and a collection $\{f\}$ of mappings in a category comprise an *abstract block diagram* if the following conditions are satisfied:

A.B.D. 1: If f is a mapping in the collection, then there exist objects A, A' in the collection such that $d(f) \subseteq A$, $r(f) \subseteq A'$.

A.B.D. 2: If A is an object of the collection and A is of the form $A_1 \times A_2 \times \ldots \times A_m$, then A_i is in the collection, for $1 \leqslant i \leqslant m$.

A.B.D. 3: If $A = A_1 \times A_2 \times \ldots \times A_m$ and there exist mappings f_i in the collection such that $r(f_i) \subseteq A_i$ for each i, then for any mapping g in the collection such that $d(g) \subseteq A$, we have

$$d(g) \cap [r(f_1) \times r(f_2) \times \ldots \times r(f_m)] \neq \emptyset.$$

These abstract block diagrams can serve as a vehicle for the representation of biological systems. To clarify the above definition, let us once again recall the situation which we are attempting to formalize. We desire that each object A in the abstract diagram shall represent a definite class of physical materials, capable of serving as inputs or outputs to a component of some system, to which the diagram may correspond. (Since not every such diagram may correspond to a physical system we use the adjective *abstract.*) Each component of the system is itself represented by a collection of mappings in the diagram, all of which possess a common domain. Since it is in general possible for a component to accept a wider class of inputs than may be provided by other components of the system, we have allowed the inclusions in our definition, instead of requiring strict equality.

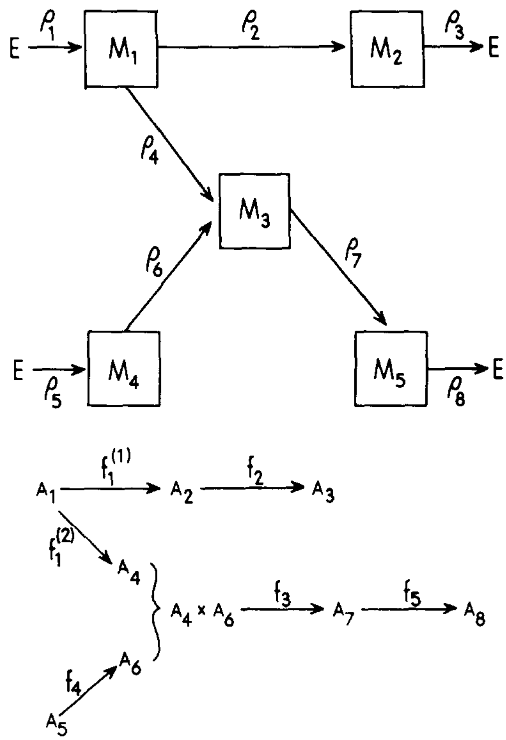
The three axioms we have imposed on an abstract block diagram will be seen to have simple interpretations in this context. The first axiom expresses in concise terms the statement that every component of a system shall have at least one input and at least one output. The second axiom is essentially a kind of closure condition and states explicitly that if a component receives an input, then that input must be represented within the system. The third axiom ensures that the system will actually be capable of stable operation, in the sense that the mappings corresponding to an arbitrary component will actually be *defined* on at least some of the possible inputs to the component.

Before we proceed to consider a specific example, let us pause for a moment to comment upon some of the aspects of the representation theory outlined above. First, we notice that the great utility of Theorem 1 lies in enabling us to consider all the classes A_i which we may encounter as being sets in the mathematical sense, and thereby enjoying all the properties and operations of sets, while allowing us to sidestep the dangerously metaphysical question of whether, and in what sense, these classes "really" are sets. Second, we observe that, although our representation takes the form of an oriented graph, just as did the cruder representation which we described in the Introduction, we find a peculiar inversion of vertices and oriented edges between our abstract diagrams and the block diagrams with which we started. That is, a vertex in our original block diagram was thought of as representing a component of the system, whereas in our abstract diagrams the components are represented by mappings, i.e., by oriented edges. Likewise, the oriented edges of the original block diagram were to represent the inputs and outputs of the system, whereas in our abstract diagrams we find that these inputs and outputs are represented by objects in a category and hence are vertices in the diagram.

To illustrate the inversion between edges and vertices mentioned above, as well as to bring out clearly the manner in which our representation theory operates, let us consider as a particular example the representation of the system **M**, the hypothetical resolution of which into components (block diagram) is shown in Figure 1.

According to our representation theory, there is a set of mappings corresponding to each component M_i. Each of the components M_i except for M_1, emit a single output; hence each of these will be represented by a single mapping which we shall designate by f_i. The component M_1, however, produces two outputs and hence is to be represented by a pair of mappings (one for each output), which we shall write as $f_1^{(1)}$ and $f_1^{(2)}$ (Figure 2).

Next, we must consider the representation of the links ρ_i shown in Figure 1. Let us suppose that ρ_i is an output of a component M_j and an input to a component M_k. We recognize two possible cases, according to whether ρ_i is the only input to M_k or not. If there are no other inputs to M_k, then we may represent the link ρ_i by the set $A_i \equiv r(f_j) \cup [\bigcup_\alpha d(f_{k_\alpha})]$, where f_j and f_{k_α} are mappings corresponding to the components M_j, M_k respectively. If M_k receives other inputs besides f_j, then for each α we find that $d(f_{k_\alpha})$ is a cartesian product, such that $r(f_j)$ is contained in one of the sets which constitute the factors of this product. We may then, for each α, project $d(f_{k_\alpha})$ onto the set containing $r(f_j)$, in the sense of Cartesian Products (see e.g., Chevalley, 1956, p. 15); let us denote this projection by $d_j(f_{k_\alpha})$. Finally, we define the set



FIGURES 1 and 2. The component M_1 in Figure 1 corresponds to the mappings $f_1^{(1)}$ and $f_1^{(2)}$ in Figure 2. The component M_2 corresponds to f_2; M_3 – to f_3; etc. The input ρ_1 in Figure 1 corresponds to the "object" A_1 in Figure 2. The inputs ρ_4 and ρ_6 into the component M_1 correspond each to the objects A_4 and A_6, respectively. The total input into component M_3 in Figure 1 thus corresponds to the Cartesian product $A_4 \times A_6$ in Figure 2.

A_i which represents the link ρ_i in the analogous manner as we did in the simple case; namely,

$$A_i \equiv r(f_j) \cup \left[\bigcup_{\alpha} d_j(f_{k_\alpha}) \right].$$

The reader can verify for himself without much trouble that this choice of mappings and objects constitutes an abstract block diagram, as we have defined it above.

All these considerations lead from the block diagram of Figure 1 to the network of mappings shown in Figure 2. We observe that this representation has removed the difficulties mentioned above concerning the representation of the environment E. We have taken E into account in a natural fashion among the A_i. Also, the inversion of vertices and edges between the diagrams of Figure 1 and Figure 2 is clearly shown; we remark that care must be taken to avoid confusion which can easily arise due to this inversion.

Since the same lines of reasoning which led from Figure 1 to Figure 2 may be applied to any biological system, we obtain our Representation Theorem.

Theorem 2. Given any system M and a resolution of M into components, it is possible to find an abstract block diagram which represents M and which consists of a collection of suitable objects and mappings from the category S of all sets.

Proof. Completely contained in the above discussion.

III. *Canonical Forms.* Our construction of an abstract block diagram to represent a given system, which we outlined in the previous section, was contingent upon a particular decomposition of the system into components. Since it is in general possible to decompose a given system in more than one manner, it therefore follows that we may correspondingly find a large number of abstract block diagrams which represent the system. In this manner, we may introduce an equivalence relation into the collection of abstract block diagrams which we may form from the objects and mappings of a given category, calling two abstract block diagrams *equivalent* if they are both representatives of the same system. It is the purpose of the present section to investigate some methods by which we may select a definite representative from each such equivalence class of abstract block diagrams; this is called a problem of determining *canonical forms* for block diagrams.

We shall first introduce some customary set-theoretic terminology which we shall find useful in formulating our results. If we are given a collection S of sets and a 1-1 onto mapping between S and the elements of some set I, then we shall say that the mapping *indexes* the collection S by the elements of I (the set I is then called an *index set* for S). We shall write an indexed collection as $(A_i)_{i \epsilon I}$. In what follows, we shall always restrict ourselves to finite collections. If the collection contains n sets, then we shall take the index set I to consist of the first n positive integers. This being so, we shall assume that the idea of the cartesian product of a family of sets $(A_i)_{i \epsilon I}$ is known. We shall write the cartesian product of an indexed collection of sets as $\prod_{i \epsilon I} A_i$.

If I is an index set, we shall call a 1-1 mapping of I onto itself as a *permutation* of I, and we shall denote such a permutation by $\pi(I)$. There is a nautral equivalence (1-1 onto mapping) between the cartesian products $\prod_{i\epsilon I} A_i$ and $\prod_{i\epsilon\pi(I)} A_i$ for any permutation π; this equivalence merely expresses the general commutativity law for cartesian products. A *subproduct* of a cartesian product $\prod_{i\epsilon I} A_i$ is obtained by writing $\prod_{j\epsilon J} A_j$, where $J \subset I$. Let us suppose that I contains n elements and J contains $m < n$ elements. We shall define the *projection* of $\prod_{i\epsilon I} A_i$ on $\prod_{j\epsilon J} A_j$ in the following manner: Choose a permutation of I such that the elements of the set J become precisely the first m elements of $\pi(I)$. Then map any element $x \epsilon \prod_{i\epsilon I} A_i$ onto its first m elements; this will be a well-determined element in $\prod_{j\epsilon J} A_j$.

This procedure of projecting a cartesian product upon a subproduct may be expressed by means of the following diagram:

$$\begin{tikzcd} \prod_{i\epsilon I} A_i \arrow[r, "\theta"] \arrow[dr, "P"'] & \prod_{i\epsilon \pi(I)} A_i \arrow[d, "\bar{P}"] \\ & \prod_{j\epsilon J} A_j \end{tikzcd}$$

Here P represents the desired projection, θ is the natural equivalence between the product and the permuted product, and $\overline{P}$ is the mapping described above which sends each element of the permuted product onto its first m elements. It is clear that $P = \overline{P} \cdot \theta$, in the sense of composition of mappings; i.e., the diagram is commutative.

Finally, if A, B, C are objects in a given category **A**, and $f : A \rightarrow B$ is a mapping in the category, then f is said to be *factorable* through C if there exists mappings $\varphi : A \rightarrow C$, $\psi : C \rightarrow B$ in **A** such that the diagram

$$\begin{tikzcd} A \arrow[r, "\varphi"] \arrow[dr, "f"'] & C \arrow[d, "\psi"] \\ & B \end{tikzcd}$$

is commutative.

With these preliminaries, we may now undertake the main portion of our discussion. To motivate our procedure, let us suppose that we are given a block diagram of a system M; let M_k be an arbitrary component of M with m inputs and n outputs. According to our representation theory, M_k will be represented by a family of n mappings, which we designate as before by f_{k_α}, $\alpha = 1, \dots, n$. The domain of each of these mappings f_{k_α} is contained in the cartesian product $\prod_{i \in I} A_i$, where I is the set of the first m integers. We now proceed to investigate in detail the possible behavior of the mappings f_k.

In particular, it is important for our purposes to investigate the question as to whether it is possible to factor a mapping f_k: $\prod_{i \in I} A_i \to B$ through a subproduct of $\prod_{i \in I} A_i$; i.e., whether it is possible to find a set $J \subset I$ such that the diagram

$$\begin{tikzcd}
\prod A_i \arrow[r, "P"] \arrow[dr, "f_{k_\alpha}"'] & \prod A_j \arrow[d, "\bar{f}_{k_\alpha}"] \\
& B
\end{tikzcd}$$

is commutative, where P is a projection, and $\bar{f}_{k_\alpha}$ is prime with respect to all subproducts of $\prod_{j \in J} A_j$. We remark that such a factorization, if it exists, is unique, as can readily be proved. Physically, the possibility of factorization of a mapping f_{k_α} means of course that the αth output of M_k does not itself require all the stated inputs to M_k in order to be produced, but only a certain subset of them; i.e., certain of the A_i are superfluous in the domain of f_{k_α}.

If we perform these factorizations on each mapping f_{k_α} in the abstract block diagram representing M, we obtain a new collection of mappings $\bar{f}_{k_\alpha}$, each defined on a subset of a certain subproduct of $\prod_{i \in I} A_i$. It may of course happen that factorization is not possible for some α, in which case we have $\bar{f}_{k_\alpha} \equiv f_{k_\alpha}$. We can now collect these mappings into classes, putting two mappings into the same class if and only if they have the same subproduct of $\prod_{i \in I} A_i$ as domain. Physically, it will be seen that the possibility of factorization implies that our original choice of components was too coarse. In other words, the component M_k may be more accurately thought of as consisting of a *number* of components, each of which is specified by a set of factored mappings $\bar{f}_{k\alpha}$ all of which have as common domain a subset of the inputs to M_k, and none of which are "superfluous."

We may carry out this procedure on every component in the block diagram of the system **M**. It is clear that the factored mappings thus obtained, together with the subproducts which constitute their domains and ranges, provide us with a new abstract block diagram which is equivalent to the one with which we started, in the sense defined above. Further, it is obtained by a sequence of canonical operations (i.e., factorizations through subproducts) and hence is in this sense unique. Hence, we have a Canonical Form Theorem.

Theorem 3. Given a block diagram for an arbitrary system **M**, we can find an abstract block diagram representing **M** such that none of the mappings of the abstract diagram is factorable through any subproduct of its domain.

Proof. See the above discussion.

This particular type of canonical form for abstract block diagrams seems to be the most natural one to consider. However, there are other types of canonical decompositions which suggest themselves, of which we mention one in particular, because of its relation to the study of general automata. Its most convenient formulation is the following:

Theorem 4. Given a decomposition of an arbitrary system **M** into components, we may find a further decomposition of **M** such that every component of the new decomposition emits exactly one output.

Proof. Let M_k be a component in the given decomposition of **M**. If M_k already emits exactly one output, there is nothing to be done. Hence, let us suppose that M_k emits $m > 1$ outputs. In the abstract block diagram representing **M**, the component M_k is represented by a family of m mappings $f_{k\alpha}$, $\alpha = 1, \ldots, m$. We now reconstruct a block diagram by allowing each of the mappings $f_{k\alpha}$ to correspond to a new component which we shall call $M_{k\alpha}$ and which is provided with the same inputs as M_k. This process is repeated for every component in the block diagram. The details of the necessary connections required to turn this set of mappings into an abstract block diagram equivalent to the one with which we started are slightly tedious, but elementary and will be omitted.

Let us now briefly turn our attention to the problem of the possible behavior of a system under the termination of an input to a component M_k. This discussion will bring to light certain aspects of our representation theory and at the same time will lead to a new formulation of Theorem 3.

To this end, let M_k be a component which receives m inputs and emits n outputs. Then M_k is to be represented, as usual, by the mappings $f_{k\alpha}$, $\alpha = 1, \ldots, n$. According to Theorem 3, each of these mappings will have a common domain, which we may write as $\prod_{i \in I} A_i$, where I may be taken as the set of the first m integers. Now the suppression of an input to M_k means precisely, in our terminology, the replacement of one of the sets A_i (A_{i_0}, say) by the empty set. This is in turn equivalent to saying that the domain space of the mappings $f_{k\alpha}$ has been reduced to a *subproduct* $\prod_{j \in J} A_j$, where J is the set I with the index i_0 deleted; i.e., $J = I - \{i_0\}$. But since the mappings $f_{k\alpha}$ may by virtue of Theorem 3 be assumed to be *prime* to subproducts of $\prod_{i \in I} A_i$, they are in general *not defined* on any such subproduct. This may perhaps be seen more clearly from a consideration of the analogous fact that a function of n variables will assume a definite value only if a value is given to *each* of the variables. Even in mathematical terms, a cartesian product, one factor of which is the empty set, is itself empty.

However, we know that there exist many systems which contain components that will produce some kind of output, even when one or more of their inputs are terminated. We represent this state of affairs as follows: We consider that a mapping $f_{k\alpha}$, which represents a component M_k, may be of such a nature that a restriction rule to subproducts for this mapping is given along with its definition. Thus, the mapping may still be defined on suitable subproducts $\prod_{j \in J} A_j$ of $\prod_{i \in I} A_i$. Such a mapping will be called *contractible* to the subproduct $\prod_{j \in J} A_j$. We shall call the elements of the range of a contracted mapping *defective outputs*, in line with their physical significance. If a mapping is such that no restriction rule to subproducts is given, the mapping will be called *non-contractible.* It will immediately be seen that this definition of contractibility is precisely the translation of the definition we proposed elsewhere (Rosen, 1958) into the terminology of our representation theory.

We may now restate Theorem 3 in a manner which more clearly shows its relationship to the concepts we have introduced above:

Theorem 5. Given a block diagram for an arbitrary system M, we can find an abstract block diagram representing M, such that if any set A is removed from the diagram, then every mapping f such that A is a factor of $d(f)$ produces defective outputs.

Proof. A restatement of the proof of Theorem 3, using the terminology introduced above. We leave the details to the reader.

The notion of contractibility is the device used in our representation theory to discuss the deeper intrinsic properties of biological systems. The simplest possible case, of course, is that in which all mappings of the diagram are non-contractible; this case was discussed at some length in our previous paper (Rosen, 1958). In the more general case, we notice that one of the important consequences of contractibility is that the range of a mapping will in general be altered by contraction, while the domains of other mappings, for which this domain may serve as factor, are left unaltered. There is no guarantee that these domains and ranges will "match up"; i.e., have a non-empty intersection. Hence a defective input may serve as no input at all to certain components. It will be seen intuitively that the notion of the *poisoning* of components by means of certain inputs is closely related to this type of situation. Also closely related to the above is the general problem of determining the *dependent sets* of the components of an arbitrary system, in the sense defined by us (Rosen, 1958) . These problems seem very difficult.

As a final word in this context, we may remark that it is possible to define a concept of the *expandability* of mappings in an arbitrary abstract block diagram, corresponding to the contractibility we have discussed above. The justification for this notion is the fact that biological systems, when placed in a richer environment, may take on new properties (i.e., emit new outputs). The notion of expandability may perhaps be a useful means for attacking this problem.

IV. *A Theorem on Functors.* It was mentioned at the time of their definition that functors furnish a natural means whereby categories can be compared, much as groups can be compared by homomorphisms and topological spaces by continuous mappings. Similarly, it follows that any structures formed from the objects and mappings in various categories may likewise be compared by functors. Thus, given an abstract block diagram (which we may denote by $\mathbf{M}$) of objects and mappings in a category $\mathbf{A}$, we may apply a functor T: $\mathbf{A} \to \mathbf{B}$ and obtain in the category $\mathbf{B}$ the collection of images under T of the objects and mappings of $\mathbf{M}$. We may write the image of the abstract block diagram $\mathbf{M}$ as $T(\mathbf{M})$. One of the natural questions to ask concerning a given functor T is whether the image of an abstract block diagram under T is again an abstract block diagram. The present section is devoted to a discussion of this question.

First, we observe that Theorem 1 implies that, in the study of any functor T: $\mathbf{A} \to \mathbf{B}$, we may as well assume that $\mathbf{A} \subset \mathbf{S}$, $\mathbf{B} \subset \mathbf{S}$, where $\mathbf{S}$ is the category of sets. Now $\mathbf{S}$ has certain structures of an algebraic nature imposed upon it, and these structures were utilized in the construction of the abstract block diagram. For instance, $\mathbf{S}$ has a local Boolean structure, imposed by the operations of intersection and union. Further, the objects of $\mathbf{S}$ form a partially ordered space under the operation of set inclusion, and a commutative semigroup under the cartesian product operation. Now an arbitrary functor T: $\mathbf{A} \to \mathbf{B}$ need have no relation to these structures, even if T is faithful; just as a 1-1 set-theoretic mapping between two groups need have no relation to the group structures. If we wish these structures to be preserved, then we must impose the necessary restrictions upon the functors which we shall consider.

Let us consider at the outset functors T which map the category of sets into itself; we shall denote this situation by T: $\mathbf{S} \to \mathbf{S}$. We shall say that such a functor is *regular* if it satisfies the following conditions:

Reg. 1: If $A \in \mathbf{S}$ and $A \neq \emptyset$, then $T(A) \neq \emptyset$.
Reg. 2: If $A \subset B$, then $T(A) \subset T(B)$.

The first condition is a non-triviality type restriction. The second condition requires that T preserve the partial ordering placed on the objects of $\mathbf{S}$ by the inclusion operation.

We shall now prove a lemma, which will be required in the proof of the main theorem of this section:

Lemma. If T: $\mathbf{S} \rightarrow \mathbf{S}$ is a regular functor, and $A \cap B \neq \emptyset$, then $T(A) \cap T(B) \neq \emptyset$.

Proof. The hypothesis $A \cap B \neq \emptyset$ means explicitly that there exists a set $C \neq \emptyset$ such that $C \subset A$ and $C \subset B$. By (Reg. 1), $T(C) \neq \emptyset$. By (Reg. 2), we have $T(C) \subset T(A)$ and $T(C) \subset T(B)$. Thus by definition we have that $T(A) \cap T(B) \neq \emptyset$.

Finally, we agree to call a functor T: $\mathbf{S} \rightarrow \mathbf{S}$ *multiplicative* if for any two sets A_1, A_2, we have that $T(A_1 \times A_2) = T(A_1) \times T(A_2)$. Thus, a multiplicative functor preserves the cartesian product operation of $\mathbf{S}$ and may in fact be regarded as a semigroup homomorphism on $\mathbf{S}$.

We are now ready to state the main result:

Theorem 6. Let $\mathbf{M}$ be an abstract block diagram which represents a definite biological system. Let T be a faithful functor. Then $T(\mathbf{M})$ is an abstract block diagram which represents the system if and only if T is regular and multiplicative.

Proof. First, let us assume that T satisfies the conditions of the hypothesis. We must then show that $T(\mathbf{M})$ satisfies the three conditions (A.B.D. 1-3) which we have laid down for an abstract block diagram.

(A.B.D. 1) is satisfied by any functor T.

To verify that (A.B.D. 2) is satisfied, let f: $\prod_{i \in I} A_i \rightarrow B$ be a mapping in $\mathbf{M}$. On applying the functor T, this becomes $T(f)$: $T(\prod_{i \in I} A_i) \rightarrow T(B)$. The multiplicativity of T, however, implies that $T(\prod_{i \in I} A_i) = \prod_{i \in I} [T(A_i)]$. But by (A.B.D. 2) we know that, for each index i, $A_i \in \mathbf{M}$. Hence each $T(A_i)$ is in $T(\mathbf{M})$, and therefore (A.B.D. 2) holds in $T(\mathbf{M})$.

To say that (A.B.D. 3) holds in $T(\mathbf{M})$ is precisely to say that T is regular, as a glance at the definitions and the lemma proved above will verify.

Thus, we have shown that $T(\mathbf{M})$ is an abstract block diagram. That it must represent the same system as $\mathbf{M}$ follows from the faithful character of T. Thus, half our theorem is proved. It remains to show that if $T(\mathbf{M})$ represents the same system as $\mathbf{M}$, then T is faithful, regular, and multiplicative on the objects and mappings of $\mathbf{M}$.

First, we show that T is faithful. The faithfulness of T on the mappings of $\mathbf{M}$ is a consequence of the fact that two components in the system represented by the same mappings in $\mathbf{M}$ must be represented by the same mappings in $T(\mathbf{M})$. The faithfulness of T on objects follows from the fact that, if a component of a system provides the same output to two distinct components, then this output must be represented by equivalent objects in $\mathbf{M}$ and in $T(\mathbf{M})$.

To prove multiplicativity of T, we observe that, if $f\colon \prod_{i\in I} A_i \to B$ is a mapping in $\mathbf{M}$, then the domain of $T(f)$ is $T(\prod_{i\in I} A_i)$. But since $T(\mathbf{M})$ represents the same system as $\mathbf{M}$, the input to the component represented by $T(f)$ must be precisely $\prod_{i\in I}[T(A_i)]$. Comparing these two, we see that the multiplicativity follows.

The regularity of T follows from the same type of argument which proved the multiplicativity.

Hence the entire theorem is proved.

Corollary. If $T\colon \mathbf{S} \to \mathbf{S}$ is a functor such that, for any abstract block diagram $\mathbf{M}$ the objects and mappings of which are in $\mathbf{S}$, $T(\mathbf{M})$ is again an abstract block diagram equivalent to $\mathbf{M}$, then T is faithful, regular, and multiplicative on all of $\mathbf{M}$, and conversely.

Proof. Immediate from Theorem 6.

Theorem 6 is a first step in the very difficult problem which plagues attempts at representation of biological systems; namely, to prove that equivalent biological systems have the same representation and, conversely, to show that two mathematically equivalent representations actually correspond to the same system. We are still far from achieving this result, since among other things we have only considered the block diagram aspect of the structure of systems and have neglected time lags and other aspects of structure. Nevertheless, Theorem 6 tells us at least how we may begin to construct equivalent representations of biological systems.

V. *Example. The McCulloch–Pitts–von Neumann Theory of General Automata.* An *automaton* consists, in the first approximation, of a black box and a finite number of inputs and outputs, each of which satisfies the special property that it is capable of assuming precisely two observable states. Following the terminology of von Neumann (1945) we shall call these states the *stimulated* and *unstimulated* states. An automaton is considered to be completely specified when it is known which of its outputs assume the stimulated state for every stimulation of a subset of its inputs.

On the basis of his general theory, von Neumann constructs every general automaton from a network of appropriate single-output automata (modulo the time lags of these automata, which will be discussed later). Hence it will be sufficient to restrict our attention to this particularly simple type of automaton. Let us denote by $\boldsymbol{U}$ a single-output automaton which receives n inputs. Each of these inputs, by definition, is capable of assuming precisely two states; let these states be denoted by the symbols 0 and 1 which shall correspond to the unstimulated and the stimulated state respectively. In the terminology of our representation we may take, for the set A which represents the set of admissible objects at each input of $\boldsymbol{U}$, the set consisting of exactly two elements 0, 1. This is the familiar "coin-tossing space," and we write $A = \{0, 1\}$.

Our representation further stipulates that the single output automaton $\boldsymbol{U}$ is to be represented by a single mapping f, the domain of which is the cartesian product of the sets of admissible objects corresponding to the inputs of $\boldsymbol{U}$. Thus, the domain of the mapping f may be taken as the cartesian product of A with itself n times. This product may be expressed simply as the set of all sequences of 0 and 1 of length n; or equivalently, as the set of all binary digits of length n.

Since the output of the automaton $\boldsymbol{U}$ is also capable of only two states, the range of the mapping f may be taken to be just the set A itself. Hence, if we introduce the notation $A^n = \prod_{i=1}^{n} A_i$ (where $A_i = A$ for each index i), then we have

$$f: A^n \to A.$$

Explicitly, $f(x) = 1$, $x \in A^n$ means that the total input results in a stimulation of the output of $\boldsymbol{U}$; $f(x) = 0$ means the opposite.

According to our general definitions, a mapping f may be factored down to a subproduct A^m, $m < n$, if and only if a mapping $\bar{f}$

and a projection $P\colon A^n \to A^m$ can be found such that the diagram

$$\begin{tikzcd} A^n \arrow[r, "P"] \arrow[dr, "f"'] & A^m \arrow[d, "\tilde{f}"] \\ & A \end{tikzcd}$$

is commutative. In the special case under consideration, we may embed A^m in A^n in a variety of ways as follows: We have a subset J of the first n integers, the elements of which we shall denote by $i_1, i_2, \ldots, i_m$. We can write this subcontract as

$$A^m_{(J)} = (\ldots 0, A_{i_1}, 0, \ldots 0, A_{i_2}, \ldots 0, A_{i_m}, \ldots),$$

where of course $A_{i_1} = A_{i_2} = \ldots = A_{i_m} = A$. For example, if we consider the subproduct $A^{n-1}_{(i)}$ obtained from A^n by omitting the i^{th} co-ordinate of A^n, we find that this can be expressed as

$$A^{n-1}_{(i)} = (A_1, A_2, \ldots, A_{i-1}, 0, A_{i+1}, \ldots A_n).$$

On this basis, it is now easy to verify explicitly that a necessary and sufficient condition for a mapping f to be factorable through the subproduct $A^{n-1}_{(i)}$ is that for every input $x \in A^n$, we have

$$f(A_1 \ldots A_{i-1}, 0, A_{i+1}, \ldots A_n) = f(A_1 \ldots A_{i-1}, 1, A_{i+1}, \ldots A_n).$$

The situation is analogous for all other subproducts. We shall assume throughout the following discussion, as usual, that f is not factorable.

We now turn our attention to the contractibility of the mapping f; that is, the behavior of the automaton $\boldsymbol{U}$ upon the termination of one or more of its inputs. We have already, in our discussion of factorization, implicitly used the fact that an empty input is equivalent to an input of 0. This means that we can treat this special case as if we did not really drop down to a lower dimensional subproduct upon terminating an input to $\boldsymbol{U}$; and hence the mapping f will always be contractible, and in fact, to every subproduct of its domain. We can write down explicitly the definition of the contracted mappings, as follows: Let $A^m_{(J)}$ be a subproduct of A^n, corresponding to an index set $J = \{i_1, i_2, \ldots, i_m\} \subset I$. Then we define the contraction f_J of the mapping f to $A^m_{(J)}$ as follows:

$$f_J(x) = 1 \text{ if and only if } x \in [f^{-1}(1) \cap A^m_{(J)}],$$
$$f_J(x) = 0 \text{ if and only if } x \in [f^{-1}(0) \cap A^m_{(J)}],$$

where, by the natural embedding discussed above, x can be regarded as an element of both $A^m_{(J)}$ and A^n, thereby giving sense to the definition of f_J.

Let us, finally, suppose that $A^{n-1}_{(i)}$ is the subproduct obtained by terminating the i^{th} input to the automaton $\boldsymbol{U}$. Then according to the definition of the contraction of f to $A^{n-1}_{(i)}$, it follows that the automation will be completely inhibited by this termination [i.e., $f^{-1}_{(i)}(A^{n-1}_{(i)}) = 0$] if and only if the condition $f^{-1}(1) \cap A^{n-1}_{(i)} = \emptyset$ is satisfied and will be completely unaffected by this termination if and only if $f^{-1}(1) \cap A^{n-1}_{(i)} = f^{-1}(1)$. This last condition is equivalent to $f^{-1}(1) \subseteq A^{n-1}_{(i)}$. We further note that our assumption of the non-contractibility of f forces this last inclusion to be *strict*, for if it were the case that $f^{-1}(1) = A^{n-1}$, then it would immediately follow that f would be factorable, through the one-dimensional subproduct $(0, \ldots 0, A_i, 0, \ldots 0)$. The generalization of this discussion of the behavior of a single-output automaton $\boldsymbol{U}$ to the termination of one of its inputs to arbitrary subproducts (i.e., to the termination of more than one input) is straightforward and can be omitted.

We now turn to the problem of the representation of the general automaton. As mentioned earlier, each general automaton can be considered as a network of single-output automata. Our representation theory shows that each single-output automaton can be represented as a mapping the range of which is the "coin-tossing space" A, and the domain of which is a cartesian product of A with itself, taken a number of times equal to the number of inputs to the automaton. Hence it follows that a category $\mathbf{A}$ sufficient for the representation of arbitrary automata is obtained by regarding the set of all finite cartesian products of A with itself as the set of objects in the category. The only mappings of the category will be the sets $H(A^n, A)$ which will be the totality of set-theoretical mappings between A^n and A. It is readily verified that $\mathbf{A}$ is indeed a category and is obviously closed under the cartesian product operation on the sets of the category.

It thus follows from the above discussion that a general automaton can always be represented as an abstract block diagram in the category **A** defined above. Conversely, it follows from Theorem 4 that an abstract block diagram of objects and mappings in the category indeed represents a general automaton. Thus, we obtain a complete characterization (ignoring time lags, for the moment), of all possible general automata. It may be of interest to observe at this point that this last remark and Theorem 6 imply that any functor R: $\mathbf{A} \to \mathbf{A}$, such that the image under R of an abstract block diagram which represents a general automaton is again a general automaton, is necessarily the identity functor on the objects of **A**, for the image of any single-output automaton, represented by the mapping f: $A^n \to A$, is transformed by R into $R(f)$: $R(A^n) \to R(A)$. In order for this image to be a single-output automaton, we must have $r[R(f)] = R(A) = A$; by Theorem 6 we have $R(A^n) = [R(A)]^n$, which by our last observation is just A^n.

The brief discussion presented above is sufficient to indicate how the graphical aspects of the theory of general automata follow in a very simple manner from the formalism which we have developed in the preceding sections. The general automaton is in fact one of the simplest possible illustrations of our formalism, and indeed the entire theory of automata might have been derived, in abstracto, from this viewpoint. The application of category theory to more general kinds of systems becomes correspondingly more complicated, but at the very least, we hope to have indicated in the foregoing that the notion of systems introduced here can be put on a rigorous basis and that the results obtained by using those notions can be formally justified.

The author is indebted to Professor N. Rashevsky for a thorough discussion of the manuscript.

This work was aided by United States Public Health Service Grant RG-5181.

LITERATURE

Chevalley, C. 1956. *Fundamental Concepts of Algebra.* New York: Academic Press.

Eilenberg S. 1957. *Categories and Fiber Bundles.* Mimeographed Lecture Notes, (Unpublished, The University of Chicago.)

Eilenberg, S. and S. MacLane. 1945. "General Theory of Natural Equivalences." *Trans. Am. Math. Soc.* 58, 231–94.

Kleene, S. C. 1955. "Representation of Events in Nerve Nets." Automata Studies, Annals of Mathematics Studies, #34, 3–41. (Ed. C. E. Shannon and J. McCarthy.) Princeton: Princeton University Press.

McCulloch, W. and W. Pitts. 1943. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bull. Math. Biophysics*, **10**, 1–10.

Neumann, J. von. 1951. "The General and Logical Theory of Automata." *Cerebral Mechanisms in Behavior* (Ed. Lloyd A. Jeffress.) New York: John Wiley & Sons.

——— 1955. "Probabilistic Logics and the Synthesis of Reliable Organisms From Unreliable Components." *Automata Studies*, Annals of Mathematics Studies, #34, 43–98. (Ed. C. E. Shannon and J. McCarthy.) Princeton: Princeton University Press.

Rashevsky, N. 1954. "Topology and Life." *Bull. Math. Biophysics*, **16**, 317–48.

Rosen, R. 1958. "A Relational Theory of Biological Systems." *Bull. Math. Biophysics* **20**, 245–60.

RECEIVED 5-30-58