Models of Deterministic Systems

by

ARTHUR W. BURKS*

Department of Computer
and Communication Sciences
The University of Michigan
Ann Arbor, Michigan 48104

ABSTRACT

The definition of "model of a system" in terms of a homomorphism on the states of the system is evaluated and an alternative definition in terms of sequence generators is proposed. Sequence generators are finite graphs whose points represent complete states of a system. Sequence generators include finite automata and other information processing systems as special cases. It is shown how to define models in terms of a projection operator which applies to any sequence generator which has an output projection and yields a new sequence generator. A model produced by the projection operator is embedded in the system it models. The notion of embedding is discussed informally and some questions raised about the relations of deterministic, indeterministic, and probabilistic models and systems.

1. Introduction. Zeigler and Weinberg have suggested the following formalization of the notion of a model for deterministic systems such as biological systems ([13], pp. 43–50). Let $A = (S, \tau)$ be a deterministic finite automaton, where S is the set of internal states and τ the transition function. A model of A is another deterministic finite automaton $A' = (S', \tau')$ satisfying the condition: there is a map $h: S \to S'$ from S onto S' such that, for every $s \in S$, $h[\tau(s)] = \tau'[h(s)]$. When h satisfies this condition it is said to be a homomorphism, and the automaton $A' = (S', \tau')$ is said to be a homomorphic image of $A = (S, \tau)$.

If A' is a homomorphic image of A, each state of A is represented by a state of A' in such a way that the law of A (i.e., the transition function τ) is modeled by the law of A' (the transition function τ'). On this ground, Zeigler and Weinberg argue that the notion of "homomorphic image" is a good formalization of the notion of "model of". I think this is a useful formalization, but that further justification of it, and an elaboration of its limitations, are both needed. In my opinion, there are many kinds of modeling relations and this is only a partial formalization of one of them. This point will be argued later. First, we will discuss some limitations of formal definitions of models in general.

The notion "model of" is often treated as a dyadic relation, holding between

* This work was supported by the National Science Foundation under Grant No. GJ-29989X. I wish to thank Andrew Lugg for helpful suggestions.

MATHEMATICAL SYSTEMS THEORY, Vol. 8, No. 4.

a model and the system modeled. But it is really a triadic relation, involving also some respect, purpose, behavior, function, criterion of importance, or point of view. For example, a good model of heart fibrillation need not explain other causes of heart failure, and certainly is not intended to model all the properties and functions of the heart. Likewise, while all properties of gases needed to derive the gas law $PV = TK$ are represented in the kinetic ("billiard ball") model of gases, many properties of gases (in particular, microscopic properties) are not represented in this model.

Thus a model $\mathscr{M}$ of a system $\mathscr{S}$ is not a model simpliciter, but a model of $\mathscr{S}$ in some specified respect $\mathscr{R}$. In no case is the model $\mathscr{M}$ intended to duplicate or model the system $\mathscr{S}$ in every respect, but only in some limited respect $\mathscr{R}$. A system is a model of itself only in a Pickwickian sense. We will show in Section 4 that a model $\mathscr{M}$ is qualitatively identical to an embedded subsystem of $\mathscr{S}$, the relation $\mathscr{R}$ defining the embedding.

Given a system $\mathscr{S}$, a modeling relation $\mathscr{R}$, and a model $\mathscr{M}$, one can ask whether $\mathscr{M}$ is a good model of $\mathscr{S}$ in respect $\mathscr{R}$. For actual systems and models, this is not a formal matter but a matter of judgment. This question of "fit" can be viewed as a question of size. Model $\mathscr{M}$ is "too small" if it does not contain all the features of system $\mathscr{S}$ needed to account for the specified relation $\mathscr{R}$. The model is "too large" if it has unnecessary features which complicate it. Whether a model $\mathscr{M}$ is the "right size" for a system $\mathscr{S}$ and the relation $\mathscr{R}$ is an informal question.

Let us consider next the formal limitations of the specific notion of a homomorphic model. For many applications the notion of a homomorphism is too strong in that it imposes a real-time requirement on the model. Each time step of the system must be represented by a single time-step of the homomorphic model. But often an actual model has a time-scale different from that of the system it models. For example, when one computer simulates another, the simulating computer generally takes several time-steps to compute one time-step of the simulated computer.

In its real-time requirement, the notion of a homomorphic model is too narrow. In another respect it is too weak. For it is based on a definition of automaton which treats each temporal state as a lump, and hence ignores the spatial or network structure of the actual system. These lumps are too big. No biological phenomenon can be understood at this global level, so biological modeling must be done at a more local level. To represent the local interactions of a system we need a richer kind of automation concept, such as a logical net, a cellular automaton, or a block diagram.

Zeigler and Weinberg find a local structure for their biological model by aggregating the coordinates of their state space. The coordinates are concentrations of various chemicals (amino acids, protein, glucose, etc.), different enzymes, various messenger RNA's, the number of cells, etc. Because the chemicals, enzymes, and messenger RNA's act as pools, the transition function of the whole system reduces to a product of partially independent local transition functions governing the pools. The system is then represented by a block diagram in which each pool is a block, and each block involves transitions to only a few other blocks ([13], pp. 51–54, and [12]).

This concludes our discussion of the limitations of formal definitions of models in general and of the notion of a homomorphic model in particular. A general maxim for handling models is appropriate here. This maxim is: Be aware of the limitations of your model, so you won't draw the wrong conclusions or make the wrong predictions.

In the present paper I will analyze the notion of a homomorphic model further, and suggest a formal notion of model which does not require that the model of a deterministic system be deterministic. For this, the formal universe of discourse must be enlarged to include non-deterministic finite automata. I will make this extension by means of the notion of sequence generator developed by Jesse Wright and me in [6] and [7].

2. Sequence Generators and Homomorphic Models. Sequence generators are finite, complete-state graphs which include finite automata and other information processing systems as special cases. Input, output, and internals states are derived from the complete states of a sequence generator by means of projections.

The basic definitions follow. A *sequence generator* $\Gamma = (S, G, R, P^1, \cdots, P^n)$ consists of a set S (whose elements are called *complete states*), a set G (whose elements are called *generators*), a binary relation R (called the *direct transition relation*), and functions $P^1, \cdots, P^n$ (called *projections*), for some $n = 0, 1, 2, 3, \cdots$, satisfying the conditions: (1) S is finite, (2) G is a subset of S, (3) R is defined on S, and each P^i (for $i = 1, 2, \cdots, n$) is also defined on S. The values of the function P^i, which may be entities of any kind, are called *P^i-states.*

A sequence generator may be represented by a finite directed graph whose vertices denote complete states and whose arrows indicate when the direct transition relation holds between two states. In our diagrams, we will use squares at those vertices which represent generator states and circles at vertices representing complete states which are not also generators; the names of complete states and of P-states are written in the circles and squares.

We are interested here in input-free finite automata, indeterministic as well as deterministic. If such an automaton has no outputs, it is represented by a sequence generator $\Gamma = (S, G, R)$. If the automaton has outputs it is represented by a sequence generator $\Gamma = (S, G, R, \Theta)$, where Θ is the output projection. The set of generators G is non-null. Also, there are no terminal states, i.e., for each $s \in S$, there is an $s' \in S$ such that $R(s, s')$. This sequence generator representation of a finite automation is more general than the usual one in that it allows an arbitrary set of starting states (generators).

The transition relation R covers both the deterministic and the indeterministic case. A sequence generator $\Gamma = (S, G, R)$ is *deterministic* if and only if, for each complete state $s_1 \in S$, there is a unique complete state $s_2 \in S$ such that $R(s_1, s_2)$, i.e., each state has a unique successor; otherwise it is *indeterministic.* (This definition differs from that of [6], p. 153, in that the present definition applies only to the input-free case and does not require that there be exactly one generator.) If Γ is deterministic, there is a function τ such that $R(s_1, s_2) \equiv [s_2 = \tau(s_1)]$. Hence in the deterministic case the transition relation R reduces to the transition function τ. Figure 1 shows a deterministic sequence generator and Figure 2 an indeterministic one. Note that we use upper case letters for the sets S and Θ and lower case letters for the elements of these sets.

We will use $[\alpha]\ (0, k)$, where k is a non-negative integer or ω to denote the sequence $\langle \alpha(0), \alpha(1), \cdots, \alpha(k) \rangle$ when k is finite, and the sequence $\langle \alpha(0), \alpha(1), \alpha(2), \cdots \rangle$ when $k = \omega$. If P is a projection, $P([\alpha]\ (0, k))]$ abbreviates the sequence $\langle P(\alpha(0)), P(\alpha(1)), \cdots, P(\alpha(k)) \rangle$ when k is finite and the sequence $\langle P(\alpha(0)), P(\alpha(1)), P(\alpha(2)), \cdots \rangle$ when $k = \omega$.

We are interested in two kinds of sequences "generated" by sequence generators: sequences of complete states and sequences of output states. These are defined as follows. A *history* of Γ (or "Γ-sequence") is any sequence $[s]\ (0, k)$ of complete states of Γ obtained by starting with a generator and following the direct transition relation; k may be a non-negative integer or ω. $\mathscr{H}(\Gamma)$ is the set of histories of Γ. The *behavior* $\mathscr{B}(\Gamma)$ of a sequence generator $\Gamma = (S, G, R, \Theta)$ is the set $\{\Theta([s]\ (0, k))\}$, where $[s]\ (0, k)$ is a Γ-sequence (finite or infinite).

We can now define "homomorphic model" in sequence generator terms. We start with a tentative definition. Let $\Gamma = (S, G, R)$ and $\dot{\Gamma} = (\dot{S}, \dot{G}, \dot{R})$ be sequence generators, with Γ deterministic. Since Γ is deterministic, the transition relation R is in fact a function, and the ordinary transition function τ is defined by the condition $R(s_1, s_2) \equiv [s_2 = \tau(s_1)]$. The tentative definition is: $\dot{\Gamma}$ is a "homomorphic model" of Γ if and only if

(1) $\dot{\Gamma}$ is deterministic
(2) There is map $\Theta: S \to \dot{S}$ from S onto $\dot{S}$ such that
 (a) Θ maps G onto $\dot{G}$, and
 (b) $\Theta[\tau(s)] = \dot{\tau}[\Theta(s)]$, where $\dot{R}(\dot{s}_1, \dot{s}_2) \equiv [\dot{s}_2 = \dot{\tau}(\dot{s}_1)]$.

Condition (2a) has appeared because all sequence generator histories start from generators.

Condition 2 is objectionable because it presupposes condition (1). We will replace it by a condition which applies to indeterministic as well as deterministic sequence generators. This new condition is based on the following modeling relation between sets of sequences.

Definition 2.1. Let S and $\dot{S}$ be sets of states and let $\{[s]\ (0, k)\}$ and $\{[\dot{s}]\ (0, k)\}$ be sets of sequences of states drawn from S and $\dot{S}$, respectively. $\{[\dot{s}]\ (0, k]\}$ *models* $\{[s]\ (0, k)\}$ if and only if there is a projection $\Theta: S \to \dot{S}$ from S onto $\dot{S}$ such that

(1) For each $[s]\ (0, k)$ there is a unique $[\dot{s}]\ (0, k)$ such that $[\dot{s}]\ (0, k) = \Theta([s]\ (0, k))$.
(2) For each $[\dot{s}]\ (0, k)$ there is at least one $[s]\ (0, k)$ such that $[\dot{s}]\ (0, k) = \Theta([s]\ (0, k))$.

Note that $\mathscr{B}(S, G, R, \Theta)$ always models $\mathscr{H}(S, G, R)$.

The groundwork is now laid for our final definition of "homomorphic model".

Definition 2.2. Let $\Gamma = (S, G, R)$ and $\dot{\Gamma} = (\dot{S}, \dot{G}, \dot{R})$ be sequence generators, with Γ deterministic. $\dot{\Gamma}$ is a *homomorphic model* of Γ if and only if (1) $\dot{\Gamma}$ is deterministic, and (2) $\mathscr{H}(\dot{\Gamma})$ models $\mathscr{H}(\Gamma)$.

This definition makes explicit the warrant for calling a homomorphic model a "model". Condition (2) requires that the histories of the model $\dot{\Gamma}$ model the histories of the system Γ via some modeling projection Θ from S onto $\dot{S}$. Each history of the system Γ is modeled, step-by-step, by a history of $\dot{\Gamma}$, and each history of $\dot{\Gamma}$ models some history of Γ.

We said in the introduction that modeling is essentially triadic: one system $\mathscr{M}$ models another system $\mathscr{S}$ in respect $\mathscr{R}$. The notion of a homomorphic model fits this pattern: $\dot{\Gamma} = (\dot{S}, \dot{G}, \dot{R})$ models $\Gamma = (S, G, R)$ when there is a map Θ from S to $\dot{S}$ which is a homomorphism. Here sequence generator $\dot{\Gamma}$ is the model $\mathscr{M}$, sequence generator Γ is the system $\mathscr{S}$, and map Θ is the modeling relation $\mathscr{R}$. Note that the modeling relation does not appear explicitly in the definiendum "$\dot{\Gamma}$ is a homorphic model of Γ" because it is existentially quantified in the definiens.

Now the map Θ is a behavioral projection applied to the sequence generator (S, G, R). It can be added directly to (S, G, R) to give an enriched sequence generator (S, G, R, Θ). We will show in the next section how to derive the model $(\dot{S}, \dot{G}, \dot{R})$ from (S, G, R, Θ) by a projection operator "Proj". This will give a new way of looking at models of deterministic systems. These models will turn out to be embedded subsystems of the given systems (Section 4).

3. The Projection Operator and Models of Deterministic Systems. We next define a projection operator "Proj (Γ)", which applies to any sequence generator $\Gamma = (S, G, R, \Theta)$ that has an output projection Θ. Proj (Γ) yields a new sequence generator $\dot{\Gamma} = (\dot{S}, \dot{G}, \dot{R})$ whose history set $\mathscr{H}(\Gamma)$ includes the behavior set $\mathscr{B}(\Gamma)$.

Informally, "Proj (Γ)" is defined as follows. The projection Θ partitions the set of complete states S into equivalence classes. These are called "Θ-states"; that is, if $\dot{s} \in \dot{S}$ & $s \in \dot{s}$ & $\Theta(s) = \theta$, then "$\dot{s}$" is named "θ". These Θ-states are the complete states $\dot{S}$ of Proj (Γ). $\dot{G}$ is the set of Θ-states which occur as projections of generators of Γ. The projection Θ then induces a transition relation $\dot{R}$ on $\dot{S}$ in a natural way: $\dot{R}(\theta_1, \theta_2)$ holds when θ_2 directly succeeds θ_1 in an element of $\mathscr{B}(\Gamma)$.

The formal definition of Proj (Γ) is this.

Definition 3.1. Let $\Gamma = (S, G, R, \Theta)$ be a sequence generator and let x range over subsets of S. Proj (Γ) $= \dot{\Gamma} = (\dot{S}, \dot{G}, \dot{R})$ is the sequence generator satisfying the following three conditions;

(1) $\dot{S}$ consists of those non-null subsets of S which agree on their projection Θ:

$$x \in \dot{S} : : \equiv : : (x \neq \varnothing) \,\&\, (s_1, s_2)\, [\{s_1 \in S \,\&\, s_2 \in S\} \supset \{s_1 \in x \,\&\, s_2 \in x\} \equiv \Theta(s_1) = \Theta(s_2)\}].$$

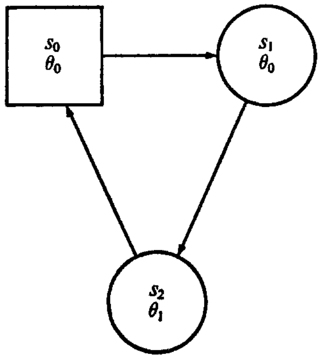
(2) $\dot{G}$ consists of those elements of $\dot{S}$ which contain at least one element of G:

$$x \in \dot{G} \equiv (Es)\,(s \in x \,\&\, x \in \dot{S} \,\&\, s \in G).$$

(3) $\dot{R}(\theta_1, \theta_2)$ holds when θ_2 directly succeeds θ_1 in an element of $\mathscr{B}(\Gamma)$:

$$\dot{R}(\dot{s}_1, \dot{s}_2) \equiv (Es_1, s_2)\,[s_1 \in \dot{s}_1 \,\&\, s_2 \in \dot{s}_2 \,\&\, R(s_1, s_2)].$$

See Figures 1 and 2 for an example of Proj (Γ).

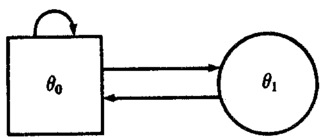


$$S_1 = \{s_0, s_1, s_2\}$$
$$G_1 = \{s_0\}$$
$$R_1 = \{\langle s_0, s_1\rangle, \langle s_1, s_2\rangle, \langle s_2, s_0\rangle\}$$
$$\Theta_1(s_0) = \Theta_1(s_1) = \theta_0, \quad \Theta_1(s_2) = \theta_1$$

Sequence generator $\Gamma_1 = (S_1, G_1, R_1, \Theta_1)$. The history set $\mathscr{H}(\Gamma_1)$ consists of the history $\langle s_0, s_1, s_2, s_0, s_1, s_2, \cdots\rangle$ and all of its initial segments. The behavior $\mathscr{B}(\Gamma)$ consists of the output sequence $\langle \theta_0, \theta_0, \theta_1, \theta_0, \theta_0, \theta_1, \cdots\rangle$ and all its initial segments.

Sequence generator Γ_1 derived from a closed deterministic finite automaton

Figure 1



$$S_2 = \{\theta_0, \theta_1\}$$
$$G_2 = \{\theta_0\}$$
$$R_2 = \{\langle \theta_0, \theta_0\rangle, \langle \theta_0, \theta_1\rangle, \langle \theta_1, \theta_0\rangle\}$$

Sequence generator $\Gamma_2 = (S_2, G_2, R_2) = \text{Proj}\ (\Gamma_1)$. The set of histories $\mathscr{H}(\Gamma_2)$ contains $\langle \theta_0, \theta_0, \theta_1, \theta_0, \theta_0, \theta_1, \cdots\rangle$ and all its initial segments, so $\mathscr{H}(\Gamma_2)$ includes $\mathscr{B}(\Gamma_1)$. $\mathscr{H}(\Gamma_2)$ also contains $\langle \theta_0, \theta_0, \theta_0, \cdots\rangle$, so $\mathscr{H}(\Gamma_2)$ is larger than $\mathscr{B}(\Gamma_1)$.

Minimal sequence generator Γ_2 whose histories $\mathscr{H}(\Gamma_2)$ include $\mathscr{B}(\Gamma_1)$.

Figure 2

An analysis of this definition shows that the behavior of the given sequence generator Γ is included in the history set of the derived sequence generator Proj (Γ). Hence we have

PROPOSITION 3.1. $\mathscr{B}(\Gamma) \subset \mathscr{H}(\text{proj}\ (\Gamma))$.

We show next that Proj (Γ) is the minimal sequence generator for which this inclusion holds. Given a sequence generator $\Gamma = (S, G, R, \Theta)$, there are infinitely many sequence generators $\Gamma^* = (S^*, \dot{G}^*, R^*)$ whose history set $\mathscr{H}(\Gamma^*)$ includes the behavior set $\mathscr{B}(\Gamma)$. We will say that one of these $\dot{\Gamma} = (\dot{S}, \dot{G}, \dot{R})$ is minimal if every S^*, G^*, and R^* is at least as large as $\dot{S}$, $\dot{G}$, and $\dot{R}$ respectively.

PROPOSITION 3.2. Let $\Gamma = (S, G, R, \Theta)$. Proj $(\Gamma) = (\dot{S}, \dot{G}, \dot{R})$ is the minimal sequence generator $\Gamma^* = (S^*, G^*, R^*)$ such that $\mathscr{B}(\Gamma) \subset \mathscr{H}(\Gamma^*)$.

Proof. The inclusion $\mathscr{B}(\Gamma) \subset \mathscr{H}(\Gamma^*)$ requires that:

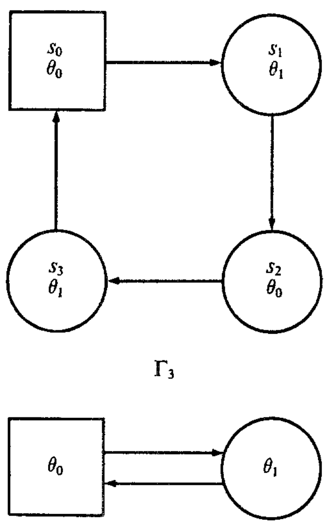
(1′) S^* contain every Θ-state of Γ;
(2′) G^* contain every Θ-state of Γ which occurs on an element of G;
(3′) $R^*(\theta_1, \theta_2)$ contain every pair $\langle \theta_1, \theta_2 \rangle$ such that θ_2 directly succeeds θ_2 in $\mathscr{B}(\Gamma)$.

Hence S^*, G^*, and R^* must be at least as large as $\dot{S}$, $\dot{G}$, and $\dot{R}$ respectively.

The projection operator was so defined that $\mathscr{B}(\Gamma) \subset \mathscr{H}(\text{Proj}\ (\Gamma))$. We next prove two propositions concerning the reverse inclusion.

PROPOSITION 3.3. *If* (Proj (Γ) *is deterministic, then* $\mathscr{H}$ (Proj (Γ)) $\subset \mathscr{B}(\Gamma)$.

Proof. Let $\Gamma = (S, G, R, \Theta)$ and Proj $(\Gamma) = \dot{\Gamma} = (\dot{S}, \dot{G}, \dot{R})$. It is given that Proj (Γ) is deterministic. We need to show that every history of Proj (Γ) is included in $\mathscr{B}(\Gamma)$. Consider any such history $\langle \theta(0), \theta(2), \theta(3), \cdots \rangle$ and two successive states θ_1, θ_2 of it. Since they are successive, $\dot{R}(\theta_1, \theta_2)$. By condition (3) of the definition of the projection operator "Proj", there are two complete states $s_1, s_2 \in S$ such that $R(s_1, s_2)$, $\Theta(s_1) = \theta_1$, and $\Theta(s_2) = \theta_2$. Moreover, because Proj (Γ) is deterministic, for any s_1', $s_2' \in S$ such that $R(s_1', s_2')$ and $\Theta(s_1') = \theta_1$, it must be the case that $\Theta(s_2') = \theta_2$. By a simple induction, the given history of Proj (Γ) can be retraced in Γ, and will be included in $\mathscr{B}(\Gamma)$. This completes the proof. (See Figure 3 for an example.)



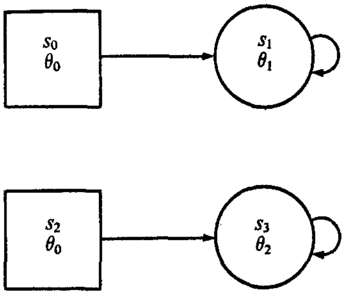
Proj (Γ_3)

Sequence generator Γ_3 with a projection θ and its projection sequence generator Proj (Γ_3). $\mathscr{H}$(Proj (Γ_3)) = $\mathscr{B}(\Gamma_3)$ = {the sequence $\langle \theta_0, \theta_1, \theta_0, \theta_1, \cdots \rangle$ and all its initial segments}.

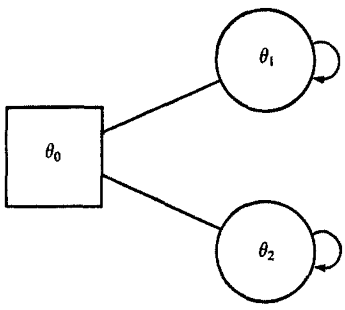
Sequence generator Γ_3 such that $\mathscr{H}$(Proj (Γ_3)) = $\mathscr{B}(\Gamma_3)$

Figure 3

PROPOSITION 3.4. *There is a sequence generator* Γ *such that* $\mathscr{H}$(Proj (Γ)) $= \mathscr{B}(\Gamma)$, *though* Proj (Γ) *is indeterministic.* (*Sequence generator* Γ_4 *of Figure 4 is an example.*)



Γ_4



Proj (Γ_4)

$$\mathscr{B}(\Gamma_4) = \mathscr{H}(\text{Proj}\,(\Gamma_4)) = \left\{\begin{matrix}\langle\theta_0, \theta_1, \theta_1, \cdots\rangle \\ \langle\theta_0, \theta_2, \theta_2, \cdots\rangle \\ \text{and all heads of these}\end{matrix}\right\}$$

Sequence generator Γ_4 such that $\mathscr{H}(\text{Proj}\,(\Gamma_4)) = \mathscr{B}(\Gamma_4)$, but Proj (Γ_4) is indeterministic

Figure 4

Our earlier approach to finding models of a given deterministic system $\Gamma = (S, G, R)$ was this (Section 2): we considered other deterministic systems $\dot{\Gamma} = (\dot{S}, \dot{G}, \dot{R})$ and asked of each whether there is a modeling homomorphism Θ from S to $\dot{S}$. In our new approach we add Θ to Γ to obtain an enriched sequence generator $\Gamma_\Theta = (S, G, R, \Theta)$, form the sequence generator Proj (Γ_Θ), and ask whether Proj (Γ_Θ) is a model of the original system $\Gamma = (S, G, R)$. The next theorem gives a condition for this.

PROPOSITION 3.5. *Let* $\Gamma = (S, G, R)$ *be deterministic and consider* $\Gamma_\Theta = (S, G, R, \Theta)$. Proj (Γ_Θ) *is a homomorphic model of* Γ *if and only if* Proj (Γ_Θ) *is deterministic.*

Proof. Since the definition of "homomorphic model" requires that Proj (Γ_Θ) be deterministic, we need prove only that if Proj (Γ_Θ) is deterministic, then Proj (Γ_Θ) is a homomorphic model of Γ. By propositions 3.1 and 3.2, if Proj (Γ_Θ) is deterministic, then $\mathscr{H}(\text{Proj}\,(\Gamma_\Theta)) = \mathscr{B}(\Gamma_\Theta)$. By virtue of the definition of "behavior", $\mathscr{B}(\Gamma_\Theta)$ models $\mathscr{H}(\Gamma)$, with $\Theta\colon S \to \{\theta\}$ from S of Γ onto the set of

Θ-states of Proj (Γ) being the modeling projection. Since $\mathscr{H}$(Proj (Γ_Θ)) = $\mathscr{B}(\Gamma_\Theta)$ and $\mathscr{B}(\Gamma_\Theta)$ models $\mathscr{H}(\Gamma)$, $\mathscr{H}$(Proj (Γ_Θ)) models $\mathscr{H}(\Gamma)$. This shows that Proj (Γ_Θ) is a homomorphic model of Γ.

We argued in Section 2 that the warrant for calling one sequence generator $\dot{\Gamma}$ a model of another sequence generator Γ is that the history set $\mathscr{H}(\dot{\Gamma})$ models the history set $\mathscr{H}(\Gamma)$. We now take this warrant as the defining condition of a weaker notion "model of".

Definition 3.2. Let $\Gamma = (S, G, R)$ and $\dot{\Gamma} = (\dot{S}, \dot{G}, \dot{R})$ be sequence generators. $\dot{\Gamma}$ is a *model* of Γ if and only if $\mathscr{H}(\dot{\Gamma})$ models $\mathscr{H}(\Gamma)$.

In this sense, a system $\dot{\Gamma}$ models a system Γ if the histories of $\dot{\Gamma}$ model the histories of Γ via some modeling function Θ from S onto $\dot{S}$. Each history of the system is modeled, step-by-step, by a history of $\dot{\Gamma}$, and each history of $\dot{\Gamma}$ models some history of Γ.

This definition of "model" applies to indeterministic as well as deterministic sequence generators, whereas "homomorphic model" applies only to deterministic sequence generators. The *homomorphic* models of a deterministic sequence generator must also be deterministic, but a *model* of a deterministic sequence generator might be either deterministic, or indeterministic as in Figure 4. Moreover, a model of an indeterministic sequence generator may be either deterministic or indeterministic. For example, it is easy to construct a deterministic model Proj (S, G, R, Θ) of an indeterministic sequence generator (S, G, R).

It is natural to ask: Which of these definitions of models (2.2 or 3.2) is correct? Can an indeterministic system really model a deterministic system, or a deterministic system an indeterministic system?

In my opinion, there is no single correct notion of "model", but many. An indeterministic system can model a deterministic system, and vice versa, in a good and basic sense of "model", the sense just defined. For in this sense of model, each history of the model tells us something about the system modeled on a step-by-step basis. We will give further arguments in the next section for our thesis that indeterministic systems can model deterministic systems and vice versa. We argue there that probabilistic systems can model deterministic systems and vice versa.

In some applications even our weaker definition of model is too strong. We require that each history of the model represent some history of the original system. But if one could recognize those histories of the model which do not represent histories of the original system one could ignore them. We also require that one history model another history in a step-by-step fashion. This requirement is sometimes too strong (see p. 305 below).

We will establish one further condition for Proj (Γ) being deterministic. This concerns the determinism of the behavior $\mathscr{B}(\Gamma)$.

Definitions. Let $\{[s](0, \omega)\}$ be a set of infinite sequences of states. This set is *deterministic* if and only if each state s occurring in a sequence of the set has a unique successor. In other words, the set of sequences $\{[s](0, \omega)\}$ is deterministic if and only if, for each state s_1, if s_1 is anywhere followed by s_2 it is always followed by s_2. Let $\Gamma = (S, G, R, \Theta)$ and note that the behavior set $\mathscr{B}(\Gamma)$

includes both finite and infinite sequences. Let $\mathscr{B}^{\omega}(\Gamma)$ be the set of infinite sequences of $\mathscr{B}(\Gamma)$. We can now define the notion of a behavior set being deterministic: the behavior $\mathscr{B}(\Gamma)$ is *deterministic* if and only if $\mathscr{B}^{\omega}(\Gamma)$ is deterministic.

PROPOSITION 3.6. *Let* $\Gamma = (S, G, R, \Theta)$. Proj (Γ) *is deterministic if and only if* $\mathscr{B}(\Gamma)$ *is deterministic.*

Proof. Let Proj $(\Gamma) = \dot{\Gamma} = (\dot{S}, \dot{G}, \dot{R})$. The two directions of the theorem will be proved separately. First, assume Proj (Γ) to be deterministic. The history set $\mathscr{H}(\dot{\Gamma})$ of a deterministic sequence generator $\dot{\Gamma} = (\dot{S}, \dot{G}, \dot{R})$ is deterministic, so $\mathscr{H}$(Proj (Γ)) is deterministic. By Propositions 3.1 and 3.3, $\mathscr{H}$(Proj $(\Gamma)) = \mathscr{B}(\Gamma)$. Hence $\mathscr{B}(\Gamma)$ is deterministic. Next, assume that $\mathscr{B}(\Gamma)$ is deterministic. By the definition of "Proj", $\dot{R}(\theta_1, \theta_2)$ holds when θ_2 directly succeeds θ_1 in an element of $\mathscr{B}(\Gamma)$. Since $\mathscr{B}(\Gamma)$ is deterministic, Proj (Γ) is.

Combining Propositions 3.5 and 3.6, we get the following result concerning homomorphic models. Suppose that we start with a deterministic system $\Gamma = (S, G, R)$, add a projection Θ to it to obtain $\Gamma_{\Theta} = (S, G, R, \Theta)$, and form Proj (Γ_{Θ}). Then Proj (Γ_{Θ}) will be a homomorphic model of Γ if and only if $\mathscr{B}(\Gamma_{\Theta})$ is deterministic. In other words, the projection Θ will produce a homomorphic model if and only if it produces a deterministic behavior.

4. Models and Embedded Subsystems. When the projection operator "Proj" produces a model, that model is embedded in the system it models. We will study this aspect of modeling in the present section.

Start with a sequence generator $\Gamma = (S, G, R)$, add an output projection to obtain $\Gamma_{\Theta} = (S, G, R, \Theta)$, and then form Proj $(\Gamma_{\Theta}) = \dot{\Gamma} = (\dot{S}, \dot{G}, \dot{R})$. The members of both $\dot{S}$ and $\dot{G}$ are subsets of S, so the states of the system $\dot{\Gamma}$ are sets of states of the system Γ, derived by means of the projection Θ. The law or transition relation $\dot{R}$ of the system $\dot{\Gamma}$ is derived from the law or transition relation of the system Γ by means of the same projection Θ.

Suppose now that $\dot{\Gamma}$ is a model of Γ, i.e., that $\mathscr{H}(\dot{\Gamma})$ models $\mathscr{H}(\Gamma)$. Then the derived law is such that for any history $[\dot{s}]$ $(0, k)$ of the model $\dot{\Gamma}$ there is a history $[s]$ $(0, k)$ of the system Γ such that $[\dot{s}]$ $(0, k) = \Theta([s]$ $(0, k))$. Hence each history of the model is a set of histories of the system it models. This shows that the model $\dot{\Gamma}$ is embedded in the given system Γ in this sense: the states of the model are sets of states of the system, and the histories of the model are sets of histories of the system.

More typically a model is separate from the system it models. But often the model is isomorphic to an embedded subsystem of the modeled system. We will give two examples from natural science which illustrate this.

The kinetic theory of gases postulates a mechanical, billiard-ball model of a gas. Consider a gas in a container at equilibrium conditions. The kinetic theory treats this as two distinct systems, an underlying system and an embedded subsystem of it. The underlying system is composed of a tremendously large number of rapidly moving, small, hard particles, which bounce elastically against each other and the walls of the container. The temporal states of this system are the so-called "microscopic states". Each contains the coordinates, velocity, and acceleration of every particle. The underlying system is governed by the laws of moving bodies and of elastic collisions.

The embedded subsystem consists of the gas as defined by its macroscopic states, each with its pressure, volume, temperature, mass, energy, and entropy. This embedded subsystem is governed by the gas law: pressure times volume equals temperature times a constant. The kinetic theory of gases defines the pressure of the subsystem in terms of the forces exerted by the particles on the walls of the container. The temperature of the gas is defined in terms of the average velocity of the particles. The kinetic theory of gases derives the gas law from the mechanical laws of the underlying system and the assumption that the initial states of the particles are randomly distributed.

Galileo's law $s = gt$ describes an idealized system (or "model") of a body falling in a vacuum. This model can be construed as an embedded subsystem of an actual underlying system. The underlying system consists of one or more objects of various sizes, shapes, weights, colors, etc., moving in the earth's gravitational field through a fluid or gas. Galileo's law of falling bodies covers the behavior of a simple system embedded in this more general system. We can derive the Galilean system from the general system by allowing only those initial states in which the buoyancy and friction of the fluid or gas have a negligible effect, and through ignoring differences in the size, shape, color, etc., of objects which have the same mass. Galileo's law of falling bodies follows from Newton's law of mechanics under these special conditions.

In the examples considered thus far the states of the embedded subsystem are sets of states of the underlying system. The next example shows that sometimes the states of the subsystem are sets of sequences of state of the system.

Consider a universal Turing machine $\mathscr{U}$ which simulates or models another Turing machine $\mathscr{M}$. $\mathscr{U}$ is given a description $\mathscr{D}(\mathscr{M})$ of $\mathscr{M}$ and a description of the initial state of $\mathscr{M}$. $\mathscr{U}$ simulates $\mathscr{M}$ by calculating its states for $t = 1, 2, 3, \cdots$. Viewed over all time, $\mathscr{U}$ contains the complete history of $\mathscr{M}$ which results from the given initial state. Moreover, $\mathscr{U}$ can calculate the history of $\mathscr{M}$ for any possible initial state of $\mathscr{M}$. Hence $\mathscr{M}$ is an embedded subsystem of $\mathscr{U}$. Since $\mathscr{U}$ uses several time steps to simulate one time step of $\mathscr{M}$, finite sequences of states of $\mathscr{U}$ represent single states of $\mathscr{M}$.

The preceding examples illustrate the essential aspects of embedding, so let us now abstract these. The histories or possible universes of an *embedded subsystem* are derived from those of the system in which it is embedded by applying either or both of the following operations. First a proper subset of the initial states of the system are chosen as the initial states of the subsystem. This operation restricts the allowed initial states of the system to obtain the initial states of the subsystem. The second operation is: sets of system states or sets of sequences of system states are designated as the states of the subsystem. This operation groups together states or sequences of states of the system to form states of the subsystem. Under these two operations, the law or rule of the system yields a derived law or rule for the subsystem such that the histories of the subsystem are sets of histories of the system.

The relation of embedding is set-theoretic. The states of the subsystem are sets of states (or of sequences of states) of the system, and the histories of the subsystem are sets of histories of the system. Consequently, a complete description of the system yields a complete description of the subsystem. A separable physical part (e.g., the carburetor of an automobile) is an embedded subsystem, but it is a very special case. More generally, an embedded subsystem is distributed throughout the underlying system.

Let us now draw a conclusion concerning the relation of modeling to embedding in the kinds of systems we have been studying. At least in these applications, the concepts of "model" and "embedded subsystem" are qualitatively identical: a model is isomorphic to an embedded subsystem of the system being modeled. Formally speaking, the modeling and embedding relations are the same.

Thus far our analysis of modeling and embedding has been for deterministic and indeterministic systems. A complete theory of modeling and embedding must encompass probabilistic systems as well, so we will conclude with some remarks concerning them.

A probabilistic system is, in essence, a probability assignment to an indeterministic system. An indeterministic transition relation allows more than one state of a system to follow a given state. A probabilistic system may be obtained by assigning probabilities to the different possible transitions.

Modeling and embedding involve two systems, and the laws of these two systems may be very different. We saw that the law of a model may be deterministic while the law of the system is indeterministic, and vice versa (p. 303). On the account of probabilistic systems just given, we would expect that a model may be deterministic while the system modeled is probabilistic, and vice versa. The following examples make this plausible.

A nuclear diffusion process is probabilistic. Suppose it is modeled in a computer by a Monte Carlo process which use pseudo-random numbers as the source of randomness. Here the embedded subsystem (the model of the nuclear diffusion processes) is probabilistic while the underlying system (the computer) is deterministic.

The converse relation is also possible. The Monte Carlo method may be used to solve the differential equations of a deterministic system such as fluid flow. The system described by the differential equation is, in a limiting fashion, an embedded subsystem of the computational system. Random numbers for a Monte Carlo calculation are usually generated by a deterministic algorithm. But one could use a natural source of randomness, such as a device which measures a random electronic effect. In that case the embedded subsystem would be deterministic while the underlying system would be probabilistic.

The notion of embedding plays a crucial role in the foundations of quantum mechanics. Quantum theory describes a probabilistic system, not a deterministic ("causal") system. Moreover, in the opinion of most physicists, quantum mechanics cannot be embedded in a deterministic system. This was the point of John von Neumann's famous "proof" that there are no hidden variables in quantum mechanics ([11], pp. 209–211 and 323–328). Kochen and Specker [10] prove von Neumann's theorem for a wider and better definition of "hidden variable".

A determinist thinks that nature is one large deterministic system. He is therefore committed to the view that every natural probabilistic system is embedded in a deterministic system. It would seem, then, that quantum mechanics refutes determinism. The determinist, however, can point out that the interpretation of quantum theory is controversial. Einstein, for example, held to determinism even in the face of quantum mechanics. He thought that quantum theory is incomplete, and that when a complete theory of quantum phenomena is developed, it will be deterministic. If Einstein was right, present quantum theory describes a probabilistic system embedded in a deterministic system ([8], pp. 176 and 666).

Actually, David Bohm has given a deterministic interpretation of quantum mechanics ([2], [3], [4]). He calls his theory a "hidden variable" theory, but he is using "hidden variable" in a much wider sense than Kochen and Specker, for his theory is not excluded by their proof (Bub [5], Gudder [9]).

Thus, whether or not the probabilistic system described by quantum theory can be embedded in a deterministic system depends on the kind of deterministic system one is willing to consider. Quantum mechanics cannot be embedded in a deterministic system described by a hidden-variable theory of the von Neumann, Kochen, and Specker type, but it can be embedded in a deterministic system with hidden variables of the Bohm type. The real controversy in the foundations of quantum mechanics is over the status and desirability of hidden-variable theories of these different types, and thus over the nature of an acceptable deterministic system.

We mention finally a different embedding relation of quantum mechanics. Neils Bohr's correspondence principle relates this subject to classical mechanics. His principle is: In the limit, where large numbers of quanta are involved, quantum laws lead to classical laws as statistical averages ([1], p. 30). This makes the deterministic system of classical mechanics the limit of a sequence of subsystems embedded in the probabilistic system of quantum mechanics. When this view is combined with Einstein's, a three-layered system of mechanics results. The deterministic system of classical mechanics is embedded in the probabilistic system of quantum mechanics, which is embedded in the deterministic system described by a complete quantum mechanics!

The present paper has been devoted mainly to embedding and modeling relations among deterministic and indeterministic systems. Our last examples show that there are also interesting embedding and modeling relations between deterministic and probabilistic systems. Clearly, then, the formal theory of modeling and embedding must be extended to cover probabilistic systems before it will be complete.

REFERENCES

[1] DAVID BOHM, *Quantum Theory*, Prentice-Hall, New York, 1951.

[2] DAVID BOHM, *Causality and Chance in Modern Physics*, Routledge & Kegan Paul, London, 1957.

[3] DAVID BOHM, Quantum theory in terms of "hidden" variables I, *Physical Review* **85** (1952), 166–179.

[4] DAVID BOHM, A suggested interpretation of quantum theory in terms of "hidden" variables II, *Physical Review* **85** (1952), 180–193.

[5] JEFFERY BUB, What is a hidden variable theory of quantum phenomena? *International Journal of Theoretical Physics* **2** (1969), 101–123.

[6] ARTHUR W. BURKS and JESSE B. WRIGHT, Sequence generators and digital computers, *Recursive Function Theory*, pp. 139–199, American Mathematical Society, Providence, 1962.

[7] ARTHUR W. BURKS and JESSE B. WRIGHT, Sequence generators, graphs, and formal languages, *Information and Control* **5** (1962), 204–212.

[8] ALBERT EINSTEIN, *Albert Einstein: Philosopher Scientist* (edited by P. A. Schillp). Library of Living Philosophers, Evanston, Illinois, 1949.

[9] STANLEY P. GUDDER, On hidden-variable theories, *J. Mathematical Physics* **11** (1970), 431–436.

[10] SIMON KOCHEN and E. P. SPECKER, The problem of hidden variables in quantum mechanics, *J. Mathematics and Mechanics* **17** (1967), 59-87.

[11] JOHN VON NEUMANN, *Mathematical Foundations of Quantum Mechanics*. Princeton