State Aggregation and Population Dynamics in Linear Systems

Jonathan E. Rowe

Michael D. Vose

Alden H. Wright
University of Montana - Missoula, alden.wright@umontana.edu

State Aggregation and Population Dynamics in Linear Systems

Jonathan E. Rowe
School of Computer Science
University of Birmingham
Birmingham B15 2TT
UK
J.E.Rowe@cs.bham.ac.uk

Michael D. Vose
Computer Science Department
University of Tennessee
Knoxville, TN 37996
vose@cs.utk.edu

Alden H. Wright
Department of Computer Science
University of Montana
Missoula, Montana 59812
wright@cs.umt.edu

Abstract We consider complex systems that are composed of many interacting elements, evolving under some dynamics. We are interested in characterizing the ways in which these elements may be grouped into higher-level, macroscopic states in a way that is compatible with those dynamics. Such groupings may then be thought of as naturally emergent properties of the system. We formalize this idea and, in the case that the dynamics are linear, prove necessary and sufficient conditions for this to happen. In cases where there is an underlying symmetry among the components of the system, group theory may be used to provide a strong sufficient condition. These observations are illustrated with some artificial life examples.

Keywords
State aggregation, population dynamics, linear systems, coarse-graining

1 State Aggregation and Dynamical Hierarchies

Many naturally occurring systems are composed of a large collection of components, which interact with each other and possibly with some background environment. These components often cluster into larger units: for example, DNA molecules in a cell nucleus, helium atoms in a balloon, people in a crowd. These units may themselves be components of some larger system. When performing simulations of such systems, rules are specified for the most basic components. It is hoped that, if the rules are correctly specified, then the behavior of the higher-level units will emerge from these low-level interactions. On the one hand, there are practical limitations to naive simulation based on this approach; quoting from [7]:

> The resulting high dimensionality of most biological systems should make . . . the state dynamic . . . completely intractable.

On the other hand, that obstacle might be circumvented:

> . . . the conceptual identification of "units" of biological structure and function . . . is [made possible by] the fortunate fact that an exhaustive . . . state space is not necessary. . . .

An interesting question, which we begin to address in this article, is whether it is possible to *deduce* "units" in terms of which the dynamics can be expressed *exactly*.[1] As a first step, we would

[1] Approximate methods for aggregating states in, say, a Markov process have been considered by other authors (see, for example, [9]).

like to have necessary and sufficient conditions. We would also like to know under what circumstances such structures are nontrivial. For example, helium atoms in a floating balloon and grains of sand in a pile [2] seem intuitively to have at most a highly trivial hierarchical structure. Can this be formalized? We will show why these examples are trivial, and characterize conditions for nontriviality.

We begin, in Section 2, by defining, in a formal way, what is meant by the term "higher-level unit." We suggest that such units could comprise aggregations of the basic components. That is, the set of underlying components will be partitioned into disjoint subsets, and each subset will then be considered as a unit in its own right. If we are given equations specifying the dynamics of the basic components, we want to be able to reconstruct the equations describing the dynamics of the subsets, *from the subsets alone*. If that is possible, we will say the particular way of partitioning components into higher-level units is *compatible* with the dynamics. Whereas approximating the dynamics of higher-level units is both interesting and sometimes the best one can do, the question this article is concerned with is whether one can do better than approximation, and if so, then by which subsets.

We will be modeling the dynamics of our system through a map that takes a *population* (that is, a collection of components in some configuration) at a given time step, and calculates the expected state at the next time step. In this article, we will limit ourselves to the case where this map is *linear* (Section 3). This case includes the description of systems forming a Markov process. The nonlinear case will be investigated in a future article. Aggregation on systems has previously been considered in [8]. However, that article assumes the set of state variables describing a system is *decomposable* as well as *aggregable*. That is, the system is decomposed into a partition of subsets, in which there is little or no interaction between the aggregated states (interaction takes place within the subsets—not between them). We do not hypothesize this restriction, and allow interactions between aggregated states to occur.

Before developing our theory, we present a simple example (Section 4), which should clarify the issues. Section 5 explains how symmetries inherent in the system of underlying components can be described mathematically using elementary group theory. This is one feature that distinguishes our work from that of [7] (another is that our Theorem 2 provides a condition that is necessary as well as sufficient). We then go on to show that the existence of such symmetries can be used to provide strong sufficient conditions under which a given state aggregation will be compatible with the underlying dynamics. This is the *group orbit theorem*, which is proved in Section 6.

We then present two examples that illustrate the application of our theorem. In Section 7 we present a simplified model of molecules passing through a unidirectional membrane. In this example, the equations describing the dynamics can be dramatically simplified by using an appropriate hierarchy of state aggregation. Section 8 looks at the mutation of binary "DNA" strings (such as are typically used in genetic algorithms, for example). In this case we characterize a range of higher-level units that are compatible with the action of mutation. Section 9 presents a final example, which speaks to the generality of our results.

NOTATION: *We follow the convention that a logical expression in square brackets, [expr], evaluates to 1 if the expression is true, and 0 otherwise.*

2 Compatibility of Equivalence Relations

In this section we formalize our concept of aggregation, and define what it means for an aggregation to be compatible with the underlying dynamics of the system. To illustrate the definitions, we will use the following simple example, based on modeling voters' behavior. We suppose there are three political parties: a center party (C), a left wing party (L), and a right wing party (R). Between two elections, there is a chance that a voter will switch allegiance. A supporter of the center party will stay loyal with probability 0.8, and switch to either of the other parties with probability 0.1 each. Left or right wing voters may vote for the center party in the next election with probability 0.4; otherwise, they will stay loyal to their parties. The dynamics of the voters' behavior from election to election is therefore described by the following Markov transition matrix:

	L	*C*	R
L	0.6	0.1	0
C	0.4	0.8	0.4
R	0	0.1	0.6

In general, we suppose that we have a set (or *population*) of N basic components in our system, and that each component may be in one of a number of states. The notion of a state is quite general. A state might be the position of a component in space, a disposition to react in a given way, an indication of how a component might be perceived by others, and so on. In our voting model, a state will be a voting choice (that is, one of $\{L, C, R\}$). We will assume that the set of possible states is finite. To preserve generality, we will simply number the states and refer to the set of all states as $\Omega = \{0, 1, 2, \ldots, n-1\}$. We will characterize a population of components by a *population vector*

$$p = \langle p_0, p_1, \ldots, p_{n-1} \rangle$$

in which p_k is the proportion of components in the population that have state $k \in \Omega$. We follow the convention that a vector delimited by angle brackets denotes a column vector (thus matrices would be multiplied on their right by population vectors). Notice that this vector is independent of the population size N, so that

$$\sum_{k=0}^{n-1} p_k = 1$$

We define the *population state space* to be

$$\Lambda = \left\{ x \in \mathbb{R}^n : \sum_k x_k = 1 \text{ and } x_k \geq 0 \text{ for all } k \right\}$$

so that any population vector is an element of Λ. In the voting model, if an eighth of the voters choose L, half choose C, and the remaining three-eighths choose R, then the population vector is $p = \langle 0.125, 0.5, 0.375 \rangle$.

We suppose that the equations describing the dynamics of the basic components of the system are known. That is, we have a map $\mathcal{G} : \Lambda \to \Lambda$, from which we can reconstruct the dynamics of the system as follows:

1. If the system is *deterministic*, then, given a population p at a given time step, $\mathcal{G}(p)$ is the population at the next time step.
2. If the system is *stochastic*, then, given a population p at a given time step, $\mathcal{G}(p)$ is a *probability distribution* over the set of states Ω. This distribution is sampled N times to give the population at the next time step.

This model is the *random heuristic search* model developed by Vose [11]. In the voting model, the map $\mathcal{G}$ is given by the transition matrix shown above. In other words,

$$\mathcal{G}(x)_L = 0.6x_L + 0.1x_C$$

$$\mathcal{G}(x)_C = 0.4x_L + 0.8x_C + 0.4x_R$$

$$\mathcal{G}(x)_R = 0.1x_C + 0.6x_R$$

We are interested in defining a partition of the components. We begin by assuming there is an *equivalence relation*, $\equiv$, defined on Ω. That is, if $a, b \in \Omega$ are equivalent ($a \equiv b$), then we consider these two states to belong to the same higher-level unit. For example, we may consider that the two extremist parties L and R are sufficiently similar that they can be lumped together into an extremist set X. This is done by declaring that they are equivalent. Such an equivalence relation can be extended to any $x, y \in \mathbb{R}^n$ as follows:

$$x \equiv y \iff \sum_{j\in\Omega}[j \equiv i]x_j = \sum_{j\in\Omega}[j \equiv i]y_j \text{ for all } i$$

It suffices that the summations (above) be equal for equivalence class *representatives* (i.e., for a collection of elements i, one from each equivalence class). Consider the components of a vector $x \in \mathbb{R}^n$ as assigning some weight to each element of Ω (that is, x_k is the weight assigned to k). Then this definition says that two vectors are equivalent if and only if they both assign the same total weight to each of the equivalence classes of Ω. In the voting model, we can say that two distributions of voters are equivalent if the same number of people vote for the extremist parties and the same number of people vote for the center party in each distribution. That is, $x \equiv y$ if $x_C \equiv y_C$ and $x_L + x_R = y_L + y_R$.

We are now in a position to relate the partitioning of Ω (in terms of the equivalence classes determined by $\equiv$) to the dynamics given by $\mathcal{G}$.

DEFINITION 1: *We say that a map $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^n$ is compatible with an equivalence relation $\equiv$ if*

$$x \equiv y \Rightarrow \mathcal{G}(x) \equiv \mathcal{G}(y)$$

for all $x, y \in \mathbb{R}_n$.

Two populations are equivalent if they assign the same weight to each equivalence class. If the map $\mathcal{G}$ is compatible, then we can follow the dynamics of just the equivalence classes (the *higher-level units*) without worrying about their microscopic details. That is, for a compatible map, we can follow the dynamics of the higher-level units as units in their own right. An interesting question, therefore, is, given a map $\mathcal{G}$ that determines the microscopic level of dynamics, what are the equivalence relations with which it is compatible? In other words, what are the naturally emergent higher-level units associated with the dynamics of the microscopic level?

In the voting model, we can directly verify that the aggregation of the two extremist parties is compatible with the dynamics. If $x \equiv y$ are two equivalent distributions, then $\mathcal{G}(x)_C = 0.4(x_L +$

$x_R) + 0.8x_C = 0.4(y_L + y_R) + 0.8y_C = \mathcal{G}(y)_C$. Similarly, $\mathcal{G}(x)_L + \mathcal{G}(x)_R = 0.6x_L + 0.2x_C + 0.6x_R = 0.6(x_L + x_R) + 0.2x_C = 0.6(y_L + y_R) + 0.2y_C = \mathcal{G}(y)_L + \mathcal{G}(y)_R$.

3 Linearity and Markov Processes

In this article we will restrict our attention to the case when the map $\mathcal{G}$ is linear. Notice that we are considering this map to be defined for the whole of $\mathbb{R}^n$ and not just on Λ. By *linear* it is meant that for any vectors $x, y \in \mathbb{R}^n$ and for any real numbers $\lambda, \mu \in \mathbb{R}^n$,

$$\mathcal{G}(\lambda x + \mu y) = \lambda \mathcal{G}(x) + \mu \mathcal{G}(y)$$

While this is an obvious limitation, it does include, for example, the case when the system is a Markov process and the map $\mathcal{G}$ is given by the transition matrix of the Markov chain. We will give three examples of simple artificial life simulations. The general nonlinear case will be the subject of further work [6].

In the linear case, we can make some progress as follows.

DEFINITION 2: *Given an equivalence relation $\equiv$ defined on Ω, let k be the number of equivalence classes. We define Ξ to be the $k \times n$ matrix with i,j entry*

$$\Xi_{i,j} = [i \equiv j]$$

(We take i to range over a set of equivalence class representatives.)

The purpose of this matrix is to map a population, considered as a distribution over the underlying component states Ω, to a corresponding distribution over the higher-level units, given by equivalence classes.

For example, suppose $\Omega = \{0, 1, 2, 3, 4, 5\}$ and our equivalence relation induces a partition $\{0, 1, 2\}$, $\{3, 4\}$, $\{5\}$. We pick the elements 0, 3, 5 to be representatives of the three classes. Then the matrix Ξ is

$$\Xi = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

If we have a population vector $p = \langle 0.1, 0.2, 0.1, 0.3, 0.15, 0.15 \rangle$, then $\Xi p = \langle 0.4, 0.45, 0.15 \rangle$, which is the distribution over the equivalence classes.

Recall that the *kernel* of a linear operator is the subspace of vectors that are all mapped to 0 by the operator. So the kernel of Ξ is the subspace

$$\text{Ker } \Xi = \{x \in \mathbb{R}^n : \Xi x = 0\}$$

(That this set forms a subspace of $\mathbb{R}^n$ can be easily checked.)

We now have the following results:

LEMMA 1: *Let $x, y \in \mathbb{R}^n$. Then $x \equiv y$ if and only if $x - y \in \text{Ker } \Xi$.*

Proof.

$$\begin{aligned} x \equiv y \quad &\iff \quad \sum_{j\in\Omega}[j \equiv i]x_j = \sum_{j\in\Omega}[j \equiv i]y_j \text{ for all representatives } i \\ &\iff \quad (\Xi x)_i = (\Xi y)_i \text{ for all } i \\ &\iff \quad \Xi x = \Xi y \\ &\iff \quad \Xi(x - y) = 0 \\ &\iff \quad x - y \in \text{Ker } \Xi \end{aligned}$$

□

THEOREM 2: *If $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^n$ is a linear map, then it is compatible with $\equiv$ if and only if* Ker Ξ *is an invariant subspace of* $\mathcal{G}$*. That is,* $\mathcal{G}(\text{Ker } \Xi) \subseteq \text{Ker } \Xi$.

Proof.

1. Suppose $\mathcal{G}$ is compatible, and let $v \in \text{Ker } \Xi$. That is, $v \equiv 0$. Therefore, by compatibility, $\mathcal{G}(v) \equiv \mathcal{G}(0)$. But since $\mathcal{G}$ is linear, $\mathcal{G}(0) = 0$. Therefore $\mathcal{G}(v) \equiv 0$, so by the previous lemma, $\mathcal{G}(v) \in \text{Ker } \Xi$.
2. Conversely, suppose that Ker Ξ is an invariant subspace of $\mathcal{G}$. Let $x \equiv y$, so that $x - y \in \text{Ker } \Xi$. By hypothesis, $\mathcal{G}(x - y)$ must also be in Ker Ξ. But $\mathcal{G}$ is linear, so $\mathcal{G}(x) - \mathcal{G}(y) \in \text{Ker } \Xi$. Therefore $\mathcal{G}(x) \equiv \mathcal{G}(y)$ and so $\mathcal{G}$ is compatible with $\equiv$. □

It is helpful to notice that Ker Ξ is the set of all vectors that assign *zero* total weight to each equivalence class.

Returning once more to the voting example, in which the dynamics is given by a matrix (that is, a linear operator), the matrix Ξ takes the form

	L	C	R
C	0	1	0
X	1	0	1

Ker Ξ is the space of all vectors satisfying $x_C = 0$ and $x_L + x_R = 0$. For any vector $x \in \text{Ker } \Xi$, we have

$$\mathcal{G}(x)_C = 0.4(x_L + x_R) + 0.8x_C = 0$$

and

$$\mathcal{G}(x)_L + \mathcal{G}(x)_R = 0.6x_L + 0.2x_C + 0.6x_R + 0.6(x_L + x_R) + 0.2x_C = 0$$

and so $x \in \text{Ker } \Xi$ implies that $\mathcal{G}(x) \in \text{Ker } \Xi$, which proves that the aggregation is compatible with the dynamics.

4 Example: Migration

We take a very simple example to illustrate our definitions. Suppose we have a population of birds that spend most of the year on certain lakes, each bird living at a particular lake. There are some lakes in the north and some lakes in the south. Every winter, some of the birds from the north fly south, while birds from the south fly north. There is a small chance a bird will not migrate. The exact lake that a bird will end up at varies each trip. We might model this process stochastically as follows. For each lake in the north, we assign a probability distribution over the southern lakes. This gives the probability that a bird from that northern lake will end up at any one of the southern likes. Similarly, for each southern lake, there is a probability distribution over the northern lakes. There is also a small probability of a bird remaining at the same lake.

For example, suppose there are five northern lakes N_1, N_2, N_3, N_4, and N_5 and three southern lakes S_1, S_2, and S_3 (see Figure 1). We associate the states of our system with these lakes through a bijection with the set $\Omega = \{0, 1, 2, 3, 4, 5, 6, 7\}$ given by

0	1	2	3	4	5	6	7
N_1	N_2	N_3	N_4	N_5	S_1	S_2	S_3

Our population of birds can be described by a population vector over these eight lakes.

Given a lake in the north, N_1 say, there is a probability distribution over S_1, S_2, and S_3 describing the chance that a bird migrating from N_1 will arrive at each of these lakes. There is also a probability ϵ that the bird will remain at N_1.

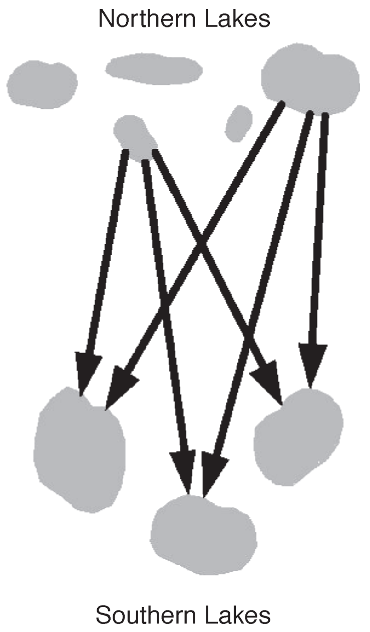

Figure 1. Birds migrate from northern to southern lakes and vice versa.

We can therefore define a linear operator describing this process by

$$M_{N_i,S_j} = \text{probability that bird will migrate from } S_j \text{ to } N_i$$

$$M_{S_j,N_i} = \text{probability that bird will migrate from } N_i \text{ to } S_j$$

$$M_{N_i,N_j} = \epsilon[i = j]$$

$$M_{S_i,S_j} = \epsilon[i = j]$$

If $p \in \Lambda$ describes the population distribution over the lakes at a given time step, then Mp gives us a distribution that, when sampled an appropriate number of times (one for each bird), simulates what happens at the next time step.

It is clear that we can reduce this eight-state system to a much simpler two-state system, by aggregating together the northern and southern lake birds, respectively. If a bird is in the north, there is a probability of remaining in the north. Otherwise it will fly south. Similarly for a bird currently in the south. We can define a linear operator for our two-state system by

	N	S
N	ϵ	$1 - \epsilon$
S	$1 - \epsilon$	ϵ

The reason this aggregation is compatible with the dynamics of the individual birds is as follows. The projection matrix Ξ for this aggregation is

$$\Xi = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

So if we had a distribution of birds $p = \langle 0.1, 0.2, 0.1, 0.05, 0.15, 0.2, 0.0, 0.2 \rangle$, then $\Xi p = \langle 0.6, 0.4 \rangle$ gives the proportions of birds in the north and south, respectively. The kernel of this operator, Ker Ξ, is the set of all vectors giving zero weight to both northern and southern lakes, that is, the set of vectors x with the property $\Xi x = \langle 0,0 \rangle$. Our theorem tells us that the dynamics of the underlying system (given by the matrix M) is compatible with this aggregation provided that if x is any given vector in the kernel of Ξ, then calculating Mx would give another vector, also in the kernel. We can check this by writing the matrix M in block form:

$$M = \left[\begin{array}{c|c} 0 & M_N \\ \hline M_S & 0 \end{array} \right] + \epsilon \mathbf{I}$$

where M_N is a 5×3 matrix giving the probability distributions for birds flying from north to south, and M_S is a 3×5 matrix giving the probability distributions for birds flying from south to north. $\mathbf{I}$ is the 8×8 identity matrix. Let $x \in$ Ker Ξ. We will write $x = \langle x_N, x_S \rangle$, where x_N is a vector giving the distribution of birds in the north, and x_S for the south. The components of each of these sum to zero, since $\Xi x = \langle 0, 0 \rangle$. Now

$$M_x = \left[\begin{array}{c|c} 0 & M_N \\ \hline M_S & 0 \end{array}\right] \left[\begin{array}{c} x_N \\ \hline x_S \end{array}\right] + \left[\begin{array}{c|c} \epsilon\mathbf{I} & 0 \\ \hline 0 & \epsilon\mathbf{I} \end{array}\right] \left[\begin{array}{c} x_N \\ \hline x_S \end{array}\right] = \left[\begin{array}{c} M_N x_S \\ \hline M_S x_N \end{array}\right] + \left[\begin{array}{c} \epsilon x_N \\ \hline \epsilon x_S \end{array}\right]$$

Let us write $\Xi M x = \langle u, v \rangle$. Then

$$\begin{aligned} u &= \sum_i (M_N x_S)_i + \sum_i \epsilon (x_N)_i \\ &= \sum_i \sum_j (M_N)_{i,j} (x_S)_j \\ &= \sum_j (x_S)_j \sum_i (M_N)_{i,j} \\ &= (1 - \epsilon) \sum_j (x_S)_j \\ &= 0 \end{aligned}$$

Similarly, we can show $v = 0$. This confirms that $Mx \in \operatorname{Ker} \Xi$, as required.

5 Symmetry and Group Theory

Suppose that, in the migration example, there were just one probability distribution over the southern lakes. Then it wouldn't matter which of the northern lakes a bird sets off from—the probability of where it arrives is the same. There would then be a symmetry between the northern lakes, meaning that we could interchange them freely and the dynamics of the system would be unaffected. It is clear that all these lakes can be grouped together in that case. Obviously, this is a much stronger condition than we need to achieve a compatible partitioning. But it gives us a clue as to how we might prove some sufficient conditions for achieving compatibility.

Mathematically, symmetry is captured using group theory. One starts with an underlying set—in our case the set is Ω. One then considers permutations of this set (that is, bijections) which somehow leave the "shape" of the set unchanged. In geometry, these permutations might be rotations and reflections, for example. We establish a collection of permutations of Ω, which we will call $L(\Omega)$ and which has the following properties:

1. The identity map is in $L(\Omega)$.
2. If $a \in L(\Omega)$ then $a^{-1} \in L(\Omega)$.
3. If a and b are in $L(\Omega)$, then so is $a \circ b$.

The collection $L(\Omega)$ of permutations forms a *group*, which is said to *act* on Ω.

The permutations represent the natural "folds" of the underlying set. They define a set of equivalence classes, called the *orbits* of the group action. For each $i \in \Omega$ we define its orbit to be

$$L(i) = \{ j \in \Omega : j = a(i) \text{ for some } a \in L(\Omega) \}$$

Each orbit is an equivalence class of the equivalence relation

$$i \equiv j \quad \Longleftrightarrow \quad j \in L(i)$$

In the following section, we will define a special group of permutations on Ω based on the properties of the map $\mathcal{G}$. This will lead to an equivalence relation, whose equivalence classes are guaranteed to be compatible with $\mathcal{G}$.

6 The Group Orbit Theorem

Let a be a permutation of Ω. We associate with a the matrix σ_a with i, j entry $[i = a(j)]$. It is simple to check that $\sigma_a^T = \sigma_a^{-1} = \sigma_{a^{-1}}$ and that $\sigma_{a \circ b} = \sigma_a \circ \sigma_b$, for all permutations a, b.

Given that $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^n$ is a linear map, define $H(\mathcal{G})$ to be the set of all permutations of Ω that *commute* with $\mathcal{G}$ in the sense that

$$a \in H(\mathcal{G}) \quad \Longleftrightarrow \quad \sigma_a \circ \mathcal{G} = \mathcal{G} \circ \sigma_a$$

Then we have the following:

LEMMA 3: *$H(\mathcal{G})$ is a group action on Ω.*

Proof. The identity is clearly in $H(\mathcal{G})$, since it commutes with any operator. If $a \in H(\mathcal{G})$, then, by definition, $\sigma_a \circ \mathcal{G} = \mathcal{G} \circ \sigma_a$, so by rearranging we see $\sigma_a^{-1} \circ \mathcal{G} = \mathcal{G} \circ \sigma_a^{-1}$ and therefore $\sigma_{a^{-1}} \circ \mathcal{G} = \mathcal{G} \circ \sigma_{a^{-1}}$. Finally, if $a, b \in H(\mathcal{G})$ then

$$\begin{aligned} \sigma_{a \circ b} \circ \mathcal{G} &= \sigma_a \circ \sigma_b \circ \mathcal{G} \\ &= \sigma_a \circ \mathcal{G} \circ \sigma_b \\ &= \mathcal{G} \circ \sigma_a \circ \sigma_b \\ &= \mathcal{G} \circ \sigma_{a \circ b} \end{aligned}$$

□

Let G be a subgroup of $H(\mathcal{G})$, and define an equivalence relation on Ω by

$$i \equiv j \quad \Longleftrightarrow \quad \exists a \in G,\ i = a(j)$$

That is, the equivalence classes are the orbits of the elements of G. We write the orbit of element i as $G(i)$. We extend this to an equivalence relation on $\mathbb{R}^n$:

$$\begin{aligned} x \equiv y \quad &\Longleftrightarrow \quad \forall t, \sum_i [i \equiv t] x_i = \sum_i [i \equiv t] y_i \\ &\Longleftrightarrow \quad \forall t, \sum_i [i \in G(t)] x_i = \sum_i [i \in G(t)] y_i \\ &\Longleftrightarrow \quad \forall t, \sum_i [i = a(t),\ \text{some } a \in G] x_i = \sum_i [i = a(t),\ \text{some } a \in G] y_i \end{aligned}$$

We want to count this sum over elements of G instead of over elements of Ω. However, there is the possibility of double counting. This occurs if there are two permutations a, $b \in G$ such that $a(t) = b(t) = i$. So, given any $i \in \Omega$, we need to know how many permutations map t to i. Write

$$G_{t,i} = \{a \in G : a(t) = i\}$$

Suppose $i \in G(t)$, and let $g \in G_{t,i}$. Consider the function

$$\phi : G_{t,t} \to G_{t,i}$$

given by

$$\phi(a) = g \circ a$$

This map ϕ is a bijection (since if $\phi(a) = \phi(b)$ then $g \circ a = g \circ b$, which gives $a = b$; and if $b \in G_{t,i}$ then $g^{-1} \circ b \in G_{t,t}$ and $\phi(g^{-1} \circ b) = b$). This means that the number of elements of $G_{t,i}$ is the same as the number of elements of $G_{t,t}$. That is, this number is independent of i. (This result is based on a theorem in [1].)

Our condition for equivalence becomes:

$$\begin{aligned} x \equiv y \quad &\iff \quad \forall t,\ \frac{\sum_{a\in G} x_{a(t)}}{|G_{t,t}|} = \frac{\sum_{a\in G} y_{a(t)}}{|G_{t,t}|} \\ &\iff \quad \forall t,\ \sum_{a\in G} x_{a(t)} = \sum_{a\in G} y_{a(t)} \\ &\iff \quad \forall t,\ \sum_{a\in G} (\sigma_{a^{-1}} x)_t = \sum_{a\in G} (\sigma_{a^{-1}} y)_t \\ &\iff \quad \forall t, \left(\sum_{a\in G} \sigma_a x\right)_t = \left(\sum_{a\in G} \sigma_a y\right)_t \\ &\iff \quad \sum_{a\in G} \sigma_a x = \sum_{a\in G} \sigma_a y \end{aligned}$$

Now suppose $x \equiv y$, and consider the effect of applying $\mathcal{G}$:

$$\begin{aligned} \sum_{a\in G} \sigma_a \mathcal{G}(x) &= \sum_{a\in G} \mathcal{G}(\sigma_a x) \\ &= \mathcal{G}\left(\sum_{a\in G} \sigma_a x\right) \\ &= \mathcal{G}\left(\sum_{a\in G} \sigma_a y\right) \\ &= \sum_{a\in G} \mathcal{G}(\sigma_a y) \\ &= \sum_{a\in G} \sigma_a \mathcal{G}(y) \end{aligned}$$

and so $\mathcal{G}(x) \equiv \mathcal{G}(y)$. We have therefore proved

THEOREM 4: *If $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^n$ is linear, and G is a subgroup of $H(\mathcal{G})$, then $\mathcal{G}$ is compatible with the equivalence relation given by the orbits of G acting on Ω.*

If $\mathcal{G}$ commutes with G, then $\mathcal{G}\,\sigma_g = \sigma_g\,\mathcal{G}$ for all $g \in G$. Let e_i denote the ith column of the identity matrix; then for all i and j,

$$\mathcal{G}_{i,j} = e_i^T \mathcal{G} e_j = (\sigma_g e_i)^T \sigma_g \mathcal{G} e_j = (\sigma_g e_i)^T \mathcal{G} \sigma_g e_j = e_g^T \mathcal{G} e_{g(j)} = \mathcal{G}_{g(i),g(j)}$$

It follows that if α and β are equivalence classes (orbits) and $a \in \alpha$, then the collection of probabilities from a to (b in) β (as b varies over β) is independent of a, since, for any $h \in G$,

$$\{\mathcal{G}_{a,g(b)} : g \in G\} = \{\mathcal{G}_{a,h^{-1} \circ g(b)} : g \in G\} = \{\mathcal{G}_{h(a),g(b)} : g \in G\}$$

This is described by saying "the outgoing probabilities are invariant." Similarly, if α and β are equivalence classes and $b \in \beta$, then the collection of probabilities from (a in) α to b (as a varies over α) is independent of b. This is described by saying "the incoming probabilities are invariant."

7 Example: Unidirectional Membranes

In this section we present a simplified model of molecular flow through a unidirectional membrane. This phenomenon occurs, for example, in Golgi bodies [3]. Molecules move around randomly within some confined space. The space is divided by a membrane, which allows the molecules to pass through in one direction only. Obviously, we expect that eventually all molecules will end up on one side of the membrane. We wish to study the dynamics of this process, using our group orbit theorem to simplify the model.

The simulation setup is shown in Figure 2. There is a 10 × 10 grid on which the molecules wander at random. At each time step, a molecule picks a neighboring square (north, south, east, or

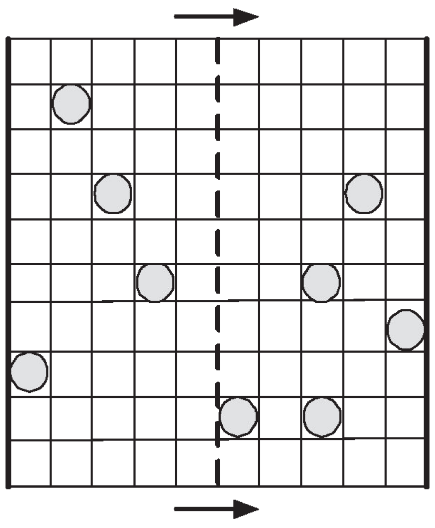

Figure 2. Molecules move around a tissue at random, but can only pass one way through the membrane.

west) to move to, uniformly at random. Molecules next to the left boundary cannot move further left: they have a choice only of three neighbors. Similarly, molecules next to the right boundary cannot move further right. It is assumed, though, that the top and bottom of the grid are connected (so the topology is cylindrical). There is a membrane running from top to bottom, halfway across the grid. Molecules to the left of the membrane can move freely across it. However, molecules to the right are not allowed to pass back. Thus, molecules immediately to the right of the membrane also have a choice of only three neighbors.

Our initial model of this system requires 100 states. We label the squares on the grid from 0 to 99, counting from left to right and from top to bottom. The distribution of molecules in the system at any time step is given by a population vector $p = \langle p_0, p_1, \ldots, p_{99}\rangle$ where p_k is the proportion of molecules in square k. Each square k has a set of neighboring squares, denoted $\nu(k)$, which are the moves allowed for a molecule at k. For example, $\nu(0) = \{1, 10, 90\}$. We describe the evolution of the system by a linear map A given by

$$A_{i,j} = \frac{[i \in \nu(j)]}{|\nu(j)|}$$

which gives the probability that a molecule moves from square j to square i. The matrix A is 100 by 100. We seek to illustrate our theory by reducing this number of states by exploiting the symmetries in the system.

For each $m = 0, 1, \ldots, 9$ we define a bijection of $\Omega = \{0, 1, \ldots, 99\}$ by

$$a_m(k) = \begin{cases} k & \text{if } k \bmod 10 \neq m \\ k + 10 \bmod 100 & \text{otherwise} \end{cases}$$

The idea is that the number m indexes a column of the grid. The application of a_m to square k is to do nothing unless the square is in column m, in which case we shift down to the square beneath (wrapping around to the top if k is on the bottom row).

The collection $a_0, a_1, \ldots, a_9$ does not itself form a group. We also need to include the effects of repeatedly applying each map (a_m^r means apply a_m repeatedly r times) and of combining maps together ($a_m \circ a_{m'}$ means rotate both columns m and m'). This gives us the group *generated* by $a_0, a_1, \ldots, a_9$ (including the identity map). It is not hard to see that this group is commutative (that is, it does not matter in what order the group elements are applied). Therefore if the map A commutes with each of $a_0, a_1, \ldots, a_9$, then it commutes with the whole group. To ease notation we will denote by σ_m the matrix with i,j entry $[i = a_m(j)]$. Now

$$\begin{aligned}(\sigma_m A)_{i,j} &= \sum_k [i = a_m(k)] \frac{[k \in \nu(j)]}{|\nu(j)|} \\ &= \sum_k [a_m^{-1}(i) = k] \frac{[k \in \nu(j)]}{|\nu(j)|} \\ &= \frac{[a_m^{-1}(i) \in \nu(j)]}{|\nu(j)|}\end{aligned}$$

and

$$\begin{aligned}(A\sigma_m)_{i,j} &= \sum_k \frac{[i \in \nu(k)]}{|\nu(k)|}[k = a_m(j)] \\ &= \frac{[i \in \nu(a_m(j))]}{|\nu(a_m(j))|}\end{aligned}$$

It is clear from the rotational action of a_m that

$$|\nu(j)| = |\nu(a_m(j))|$$

and that

$$a_m(\nu(j)) = \nu(a_m(j))$$

and therefore $\sigma_m A = A\sigma_m$.

We therefore know that A commutes with the whole group and so will be compatible with its orbits. But what are the orbits of the group action? Applying group members to a particular square either leaves it alone (if it is not in the appropriate column) or rotates it downwards. The orbits are therefore the columns of the grid. Our theorem tells us, therefore, that we can consider each column to be a high-level unit in its own right, and that we can write down the dynamics of the system in terms of these units. Accordingly, we number the columns giving a new set $\Omega = \{0, 1, \ldots, 9\}$. The dynamics of the system at this new level of description is given by the linear operator

$$C = \begin{bmatrix} 2/3 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/2 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 1/2 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/4 & 1/2 & 1/4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 2/3 & 1/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3 & 1/2 & 1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/2 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/2 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 2/3 \end{bmatrix}$$

We have successfully reduced the number of states of the system from 100 to 10, by aggregating states into higher-level units. However, we can go further still. Note that an invariant subspace of C is spanned by $\{e_5, e_6, e_7, e_8, e_9\}$. Moreover, the orthogonal complement of $\langle 1, \ldots, 1\rangle$ is also an invariant subspace, since C is column stochastic. It follows that the intersection S of these spaces is invariant. By Theorem 2, therefore, we could aggregate further by choosing Ξ such that its kernel is contained in S,

$$\Xi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Hence we can aggregate together all of the columns to the right of the membrane, and remain compatible with the dynamics. Our group orbit theorem does not apply here, however, as the probability of moving from column 4 to column 5 (that is, 1/4) is not identical to the probability of moving to column 6, say (which is zero). Remember that the group orbit theorem gives a *sufficient* condition for compatibility, which is rather strong—it requires all the incoming and outgoing probabilities to be invariant.

Aggregation according to Ξ reduces the number of states to six. We number these 0, 1, 2, 3, 4, and 5, where state 5 represents all the columns to the right of the membrane; the other states are the corresponding left columns as before. The resulting dynamics is given by the matrix $R = \Xi\, CD^{-1}\Xi^T$, where the diagonal matrix D has its ith diagonal entry equal to the cardinality of the equivalence class containing i (the details of why the dynamics of the aggregation would take this form are laid out in [11] and, using different language, in [7]). Therefore,

$$R = \begin{bmatrix} 2/3 & 1/4 & 0 & 0 & 0 & 0 \\ 1/3 & 1/2 & 1/4 & 0 & 0 & 0 \\ 0 & 1/4 & 1/2 & 1/4 & 0 & 0 \\ 0 & 0 & 1/4 & 1/2 & 1/4 & 0 \\ 0 & 0 & 0 & 1/4 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1 \end{bmatrix}$$

We can then use standard Markov chain theory to predict the expected time for a molecule to cross the membrane, depending on which column it is starting in:

Initial state	0	1	2	3	4
Time to cross	55	52	45	34	19

8 Example: Mutation of Binary DNA

Our next example relates to the mutation of "DNA," which we represent as a fixed-length binary string (for example, this is common in genetic algorithms). Each bit in a string will be mutated with some fixed probability u. Our population comprises a pool of such strings, and we wish to track their evolution under this mutation operator.

The set of all binary strings of length ℓ can be conveniently identified with the set of integers $\Omega = \{0, 1, \ldots, 2^\ell - 1\}$ by interpreting the strings as being integers written in base 2. The population at a given time step is therefore given by a vector $p = \langle p_0, p_1, \ldots, p_{2^\ell-1}\rangle$. The dynamics of this system is given by a matrix U defined by

$$U_{i,j} = u^{h(i,j)}(1-u)^{\ell-h(i,j)}$$

where $h(i, j)$ is the Hamming distance between strings i and j (that is, the number of bits at which they differ in value). Put another way, $i \oplus m$ is the result of mutating i, where $\oplus$ denotes bitwise exclusive OR (of bit strings), m is chosen with probability

$$u^{\#m}(1-u)^{\ell-\#m}$$

and $\#m$ denotes the number of bits in m that are equal to 1 (m is commonly referred to as a mutation *mask*).

We wish to establish whether or not the set of all strings can be partitioned into higher-level units which are compatible with these dynamics. We are therefore led to identify the structure of the group $H(U)$.

THEOREM 5: *Suppose Ω is identified with the set of binary strings of length ℓ, and let U be the mutation matrix corresponding to bitwise mutation with rate u. Then $H(U)$ is the group of the automorphisms of the hypercube.*[2]

***Proof*.** The hypercube is the graph that has elements of Ω as vertices, with edges between vertices that differ in only one bit. Let $a \in H(U)$. We can view a as a permutation of vertices of the hypercube. Let $i, j \in \Omega$ be vertices that share an edge, so that $i \oplus j$ contains a single bit. We want to show that $a(i)$ and $a(j)$ also share an edge.

Since U commutes with a, we have $U_{a(i),a(j)} = U_{i,j}$. Hence the probability of picking mask $i \oplus j$ is the same as picking $a(i) \oplus a(j)$. But when mutation is by a rate, this probability depends only on the number of ones in each mask. Thus if i and j differ in one bit, so do $a(i)$ and $a(j)$. This shows that $a \in H(U)$ is an automorphism of the hypercube.

Conversely, suppose π is an automorphism. Then the number of ones in $i \oplus j$ is the same as the number of ones in $\pi(i) \oplus \pi(j)$, for every $i, j \in \Omega$.[3] Therefore the probability of picking $i \oplus j$ as a mask is the same as picking $\pi(i) \oplus \pi(j)$, and so $U_{\pi(i),\pi(j)} = U_{i,j}$. It follows that U commutes with π (using the matrix algebra following Theorem 4). □

The automorphisms of the hypercube are generated by

1. masks (under exclusive-or),
2. permutations of bit positions.

For example, suppose $\ell = 5$, and consider the set of strings

$$\{00000,\ 01000,\ 10000,\ 11000\}$$

2 An *automorphism* of a graph is a permutation of the vertex set such that edges are mapped to edges. See [1].

3 This follows by induction on the number of ones in question and using the fact that the automorphisms form a finite group under composition.

If we let each of these strings act on Ω as a mutation mask, then we obtain a subgroup of the automorphisms. This subgroup generates orbits whose members agree on the final three bits. Such a collection is referred to as a *schemata family* in the genetic algorithm literature [5]. Another subgroup is the set of permutations of bit positions that shuffle the order of the first four bits. The orbits generated by this subgroup comprise sets whose strings contain the same number of ones and zeros in the first four positions. Such collections are sometimes referred to as *unitation* classes. The fact that mutation is compatible with unitation classes can be exploited to investigate the evolution of asexual populations [4, 12]. Other equivalence relations that are compatible with mutation include combinations of schemata and unitation classes. For example, we could collect together strings that agree with each other on bits 2 and 4 and have the same number of ones in bits 1, 3, 5. One such set is $\{01011, 01110, 11010\}$.

It should be noted that aggregation by schemata is also compatible with the dynamics of a variety of standard crossover operators, although here the dynamics are nonlinear [11, 10]. Unitation classes are not compatible with crossover, however.

9 Generality

This section collects some simple observations of a technical sort concerning the generality of our framework (this framework is developed in [11]). There need not be anything special about the coordinate system that determines the matrix of the linear operator $\mathcal{G}$. A change of variable, say $y = Wx$, transforms the system $x' = \mathcal{G}x$ into the equivalent system $y' = W\mathcal{G}W^{-1}y$ in which Ξx has become $\Xi W^{-1}y$. In this new system, the equivalence operator (or "aggregation operator"; call it what you will) is an arbitrary full rank matrix, namely ΞW^{-1} (it is arbitrary to the extent W is).

This process can be reversed. For example, consider the system $y' = Ay$, where

$$A = \begin{bmatrix} 1 & 4 & 0 \\ 2 & 5 & -2 \\ 3 & 6 & -2 \end{bmatrix}$$

Simply *imagine* that Ξ represents an equivalence relation $\equiv$ for some system $x' = \mathcal{G}x$ from which $y' = Ay$ is obtained through some change-of-variables matrix W. Since the kernel of the matrix (with *imagined* name ΞW^{-1})

$$\Xi W^{-1} = \begin{bmatrix} 1 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}$$

is $\{\langle r, 0, r\rangle : r \in \mathbb{R}\}$, which is invariant under A, Theorem 2 can be applied, and implies that the imagined relation $\equiv$ is compatible with $\mathcal{G} = W^{-1}AW$. Expressed in our system $y' = Ay$, this means the variables $\langle u, v\rangle = \langle y_0 + y_1 - y_2, y_2 - y_0\rangle$ are "higher-level units" which have dynamics in their own right that are compatible with A. Choosing W to be *any* full rank matrix making Ξ a partition (let W map—by multiplying on its left—the displayed matrix above to a partition matrix), say

$$W = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

allows our previously imagined objects to be realized:

$$\Xi = \begin{bmatrix} 1 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$$\mathcal{G} = \begin{bmatrix} 1 & 1 & -1 \\ -1 & -1 & 2 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 0 \\ 2 & 5 & -2 \\ 3 & 6 & -2 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

The transformation matrix $\Xi \mathcal{G} D^{-1} \Xi^T$ for $\langle u, v \rangle$ is therefore (this will be explained below)

$$\begin{bmatrix} 3 & 3 \\ 2 & 0 \end{bmatrix}$$

This checks out (as it must), since

$$\begin{bmatrix} 3 & 3 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 0 \\ 2 & 5 & -2 \\ -3 & 6 & -2 \end{bmatrix}$$

This example illustrates how there is no loss of generality in Ξ being a partition matrix. It also provides an example where "higher-level units" (in the system $y' = Ay$) are not intuitively obvious, and are composed of overlapping basic units; $v = y_2 - y_0$ is contained within $u = y_1 - (y_2 - y_0)$.[4]

Three points deserve clarification. First, why can Theorem 2 be applied to A and the general aggregation operator ΞW^{-1} to conclude something about the compatibility of $\equiv$ with $\mathcal{G}$? The justification is the general equivalence

$$A : \mathrm{Ker}(\Xi W^{-1}) \rightarrow \mathrm{Ker}(\Xi W^{-1}) \iff \mathcal{G} : \mathrm{Ker}(\Xi) \rightarrow \mathrm{Ker}(\Xi)$$

which depends only on $\mathcal{G} = W^{-1} A W$. Second, why is the matrix $\Xi \mathcal{G} D^{-1} \Xi^T$ for the reduced system the same for both systems, $y' = Ay$ and $x' = \mathcal{G}x$? The justification is the general equivalence

$$(\Xi \mathcal{G} D^{-1} \Xi^T)(\Xi) = (\Xi)(\mathcal{G}) \iff (\Xi \mathcal{G} D^{-1} \Xi^T)(\Xi W^{-1}) = (\Xi W^{-1})(A)$$

4 We are speaking of the system $y' = Ay$ (in the system $x' = \mathcal{G}x$ there is no overlap).

which depends only on $A = W\mathcal{G}W^{-1}$. Third, since Theorem 2 may be applied directly to A and the general aggregation operator ΞW^{-1} (in the manner illustrated and justified above), why was equivalence not more generally defined? We think it interesting that general aggregation is achieved via partitioning underlying components in a system obtained by a simple change of variable. It is that fact that we wanted to stress. It suggests that the initial coordinate system (or basic components) that define $y' = Ay$ might not be intrinsic; perhaps a different choice (one that admits equivalence by way of partitioning) could be more natural.

10 Conclusions and Further Work

We have presented a formalization for one aspect of what might be meant by "dynamical hierarchy": a hierarchy of system representations at different levels of granularity that are mutually compatible with respect to their dynamics. We view this as the situation in which the microscopic elements of a system can be clustered into higher-level units in such a way that the macroscopic dynamics is compatible with the rules of microscopic behavior. Under these conditions, the higher-level units might be said to be naturally emergent properties of the system. Of course this will not work for arbitrary methods of clustering, and we have begun to investigate conditions under which this can be achieved. In the case where the underlying dynamics can be described as a linear function of the population state space, we have provided a necessary and sufficient condition for compatibility (namely, that the kernel of the corresponding projection is an invariant subspace of the dynamics). We have also proved a strong sufficient condition that exploits symmetries within the set of underlying components, using the language of group theory. In the general case, the microscopic dynamics may be described by a nonlinear map $\mathcal{G}$. In this case, more work needs to be done to provide a characterization of the compatible equivalence classes. The development of this theory will be the subject of a future article.

If we return to the examples of the floating balloon and the sandpile, we can now see exactly why their structures are trivial. In the case of the helium balloon, there are only two possible ways of aggregating the atoms that are compatible with the dynamics. Either each individual helium atom is in an equivalence class of its own, or we take the whole balloon to be a single aggregation. Of course, these two aggregations are always possible, in any system. The helium balloon has only trivial structure, because only trivial aggregations are possible. The situation with the pile of sand is different, however. As long as the sand does not move, the *dynamics* of the system are trivial (that is, $\mathcal{G}$ is simply the identity map). This means that *any* aggregation of states will be compatible with the dynamics. We can partition the grains of sand into any subsets we like, and the resulting states will be compatible with the dynamics. This is the opposite extreme to the floating balloon example, but still renders the example trivial. If all ways of aggregating are possible, then there is no reason to prefer one way of doing it to another. We now see that the two examples are trivial, but for completely different reasons. This observation enables us to characterize nontrivial dynamical hierarchies as being a subset system (ordered by inclusion) of compatible partitions, that lies between these two extremes.

Acknowledgments

Michael D. Vose and Alden Wright completed some of this work while visiting the University of Birmingham. The work was supported by EPSRC grant GR/R47394. We would like to thank the anonymous reviewers for helpful comments.

References

1. Biggs, N. L. (1971). *Finite groups of automorphisms.* Cambridge, UK: Cambridge University Press.
2. Gross, D., & McMullen, B. (2001). Is it the right ansatz? Artificial Life, *7*(4), 355–365.
3. Presley, J. F., Smith, C., Hirschberg, K., Miller, C., & Cole, N. B. (1998). Golgi membrane dynamics. *Molecular Biology of the Cell*, *9*(7), 1617–1626.

4. Rowe, J. E. (1999). Population fixed-points for functions of unitation. In W. Banzhaf & C. Reeves (Eds.), *Foundations of genetic algorithms (FOGA-5)*. San Mateo, CA: Morgan Kaufmann.

5. Rowe, J. E., Vose, M. D., & Wright, A. H. (2002). Group properties of crossover and mutation. *Evolutionary Computation*, *10*(2), 151–184.

6. Rowe, J. E., Vose, M. D., & Wright, A. H. (2005). Coarse-graining selection and mutation. *Foundations of genetic algorithms* (Vol. 8, pp. 176 – 191). Berlin: Springer-Verlag.

7. Shpak, M., Stadler, P. F., Wagner, G. P., & Hermisson, J. (2003). Aggregation of variables and system decomposition: Applications to fitness landscape analysis (Technical report 03-04-025). Santa Fe Institute.

8. Simon, H., & Ando, J. (1961). Aggregation of variables in dynamic systems. *Econometrica*, *29*, 111–138.

9. Spears, W. M., & Jong, K. D. (1997). Analysing GAs using Markov models with semantically ordered and lumped states. In R. K. Belew & M. D. Vose (Eds.), *Foundations of Genetic Algorithms* (Vol. 4, pp. 85–100). San Mateo, CA: Morgan Kaufmann.

10. Stephens, C. R., & Waelbroeck, H. (1999). Schemata evolution and building blocks. *Evolutionary Computation,* *7*(2), 109–124.

11. Vose, M. D. (1999). *The simple genetic algorithm: Foundations and theory.* Cambridge, MA: MIT Press.

12. Vose, M. D., & Rowe, J. E. (2000). Random heuristic search: Applications to GAs and functions of unitation. *Computer Methods in Applied Mechanics and Engineering*, *186*(2–4), 195–220.