Compositional properties of random Boolean networks

Elena Dubrova and Maxim Teslenko
Department of Microelectronics and Information Technology, Royal Institute of Technology, Electrum 229, 164 40 Kista, Sweden
(Received 28 October 2004; published 23 May 2005)

Random Boolean networks (RBNs) are used in a number of applications, including cell differentiation, immune response, evolution, gene regulatory networks, and neural networks. This paper addresses the problem of computing attractors in RBNs. An RBN with n vertices has up to 2^n states. Therefore, for large n, computing attractors by full enumeration of states is not feasible. The state space can be reduced by removing irrelevant vertices, which have no influence on the network's dynamics. In this paper, we show that attractors of an RBN can be computed compositionally from the attractors of the independent components of the subgraph induced by the relevant vertices of the network. The presented approach reduces the complexity of the problem from $O(2^n)$ to $O(2^l)$, where l is the number of relevant vertices in the largest component.

DOI: 10.1103/PhysRevE.71.056116 PACS number(s): 89.75.Hc, 87.23.Kg, 87.17.Aa

I. INTRODUCTION

This paper studies compositional properties of *random Boolean networks* (RBNs). An RBN is a synchronous Boolean automaton with n vertices. Each vertex has k incoming edges, selected at random, and an associated Boolean function. Functions are selected so that they evaluate to the values 0 and 1 with given probabilities p and $1-p$, respectively. The set of function's values of vertices at a given time characterizes the current state of a network.

RBNs were introduced by Kaufmann [1] in 1969 in the context of gene expression and fitness landscapes. Later, they were applied to the problems of cell differentiation, immune response, evolution, and neural networks [2,3]. They have attracted the interest of physicists [4–6] due to their analogy with the disordered systems studied in statistical mechanics, such as the mean field spin glass.

The parameters k and p determine the dynamics of an RBN. If a vertex controls many other vertices, and the number of controlled vertices grows in time, the RBN is said to be in a *chaotic phase*. Typically such a behavior occurs for large values of $k \sim n$. The next states of the RBN are random with respect to the previous ones. The dynamics of the network is very sensitive to changes in the state of a particular vertex, associated Boolean function, or network connections.

If a vertex controls only a small number of other vertices and their number remains constant in time, the RBN is said to be in a *frozen phase*. Usually, independently on the initial state, after a few steps, the network reaches a stable state. This behavior usually occurs for small values of k, such as $k=0$ or 1.

There is a critical line between the frozen and the chaotic phases, when the number of vertices controlled by a vertex grows in time, but only up to a certain limit. Statistical features of RBNs on the critical line are shown to match the characteristics of real cells and organisms [1,7–10]. The minimal disturbances create typically only slight variations in the network's dynamics. Only some rare perturbations evoke radical changes.

For a given probability p, there is a critical number of inputs k_c below which the network is in the frozen phase and above which the network is in the chaotic phase [4,11]:

$$k_c = \frac{1}{2p(1-p)}.$$

For example, for $p=0.5$, the critical number of inputs is $k_c = 2$.

Since the number of possible states of an RBN is finite, any sequence of consecutive states of a network eventually converges to either a single state, or a cycle of states, called *attractor*. For large RBNs, computing attractors by full enumeration of states is an infeasible task. It is possible to reduce the state space of an RBN by removing vertices belonging to its *stable core*. The stable core is defined by Flyvbjerg [12] as the set of vertices whose output value develops in time to a constant value that is independent of the initial state of the RBN. Bastola and Parisi [13] have observed that the state space can be further reduced by removing, e.g., vertices which have no outputs. They introduced a notion on *relevant vertex*, which is a vertex which has an influence on an RBN's dynamics. Exact and approximate bounds on the size of the set of relevant vertices for different values of k and p have been given [12–16]. In the infinite size limit $n \to \infty$, in the frozen phase, the number of relevant vertices remains finite. In the chaotic phase, the number of relevant vertices is proportional to n. On the critical line, the number of relevant vertices scales as $n^{1/3}$ [17]. Bastola and Parisi [18] have also studied the structural properties the independent components of the subgraph induced by the relevant vertices on an RBN. In the infinite size limit, in the frozen phase, the components are loops of effective connectivity 1. In the chaotic phase, the number of components tends to 1. On the critical line, the average number of components is of order of $\log n$. Derrida and Stauffer [11] have considered the lattice version of RBNs.

In this paper, we show that attractors of an RBN can be computed *compositionally* from the attractors of the independent components of the subgraph induced by the relevant vertices. For an RBN with n vertices, the presented approach reduces the complexity of the problem from $O(2^n)$ to $O(2^l)$, where l is the number of relevant vertices in the largest component.

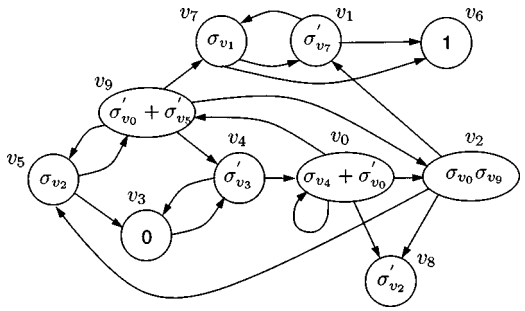

FIG. 1. Example of an RBN. The state of a vertex v_i at time $t+1$ is given by $\sigma_{v_i}(t+1)=f_{v_i}(\sigma_{v_j}(t),\sigma_{v_k}(t))$, where v_j and v_k are the predecessors of v_i, and f_{v_i} is the Boolean function associated to v_i (shown by the Boolean expression inside v_i).

The paper is organized as follows. Section II describes basic notation and definitions. Section III gives the main result of the paper. Section IV concludes the paper and discusses open problems.

II. RANDOM BOOLEAN NETWORKS

A random Boolean network is a directed cyclic graph $G=(V,E)$, where V is the set of vertices and $E\subseteq V\times V$ is the set of edges connecting the vertices.

The set V has n vertices. Each vertex $v\in V$ has k incoming edges, selected at random. The set of *predecessors* of v is defined by

$$P_v=\{u\in V|(u,v)\in E\}.$$

The set of *successors* of v is defined by

$$S_v=\{u\in V|(v,u)\in E\}.$$

Each vertex $v\in V$ has an associated Boolean function, f_v, of type $\{0,1\}^k\rightarrow\{0,1\}$. The state σ_v of a vertex v at time $t+1$ is determined by the states of its predecessors $u_i\in P_v$, $i\in\{1,\ldots,k\}$, as

$$\sigma_v(t+1)=f_v(\sigma_{u_i}(t),\sigma_{u_2}(t),\ldots,\sigma_{u_k}(t)).$$

The vector $\Sigma=(\sigma_{v_1}(t),\sigma_{v_2}(t),\ldots,\sigma_{v_n}(t))$ represents the state of the network at time t. An example of an RBN with ten vertices is shown in Fig. 1.

An infinite sequence of consecutive states of a network is called a *trajectory*. A trajectory is uniquely defined by the initial state. Since the number of possible states is finite, all trajectories eventually converges to either a single state, or a cycle of states, called *attractor*. The *basin of attraction* of A is the set of all trajectories leading to the attractor A. The *attractor length* is the number of states in the attractor's cycle.

Next, we define irrelevant vertices.

Definition 1. A vertex $v\in V$ is irrelevant for an RBN G if the network obtained from G by removing v has the same number and length of attractors.

There are several types of irrelevant vertices. First, all vertices v whose associated function f_v is constant 0 or constant 1 are irrelevant. If u is a successor of an irrelevant vertex v and if after the substitution of the constant value of f_v in f_u the function f_u reduces to a constant, then u is irrelevant, too.

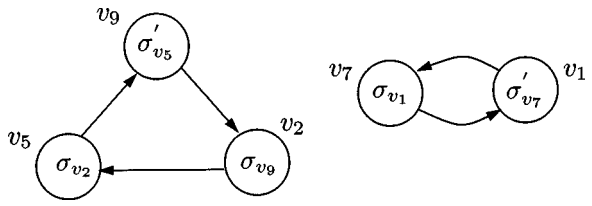

FIG. 2. Reduced network G_R for the RBN in Fig. 1.

Second, all vertices v which have no successors are irrelevant. If u is a predecessor of an irrelevant vertex v and if all successors of u are irrelevant, then u is irrelevant, too.

Third, a vertex can be irrelevant because its associated function f_v has a constant value due to the correlation of its input variables. For example, if a vertex v with an associated OR (AND) function has predecessors u_1 and u_2 with functions $f_{u_1}=\sigma_w$ and $f_{u_2}=\sigma_w'$, then the value of f_v is always 1 (0). This kind of irrelevant vertices are hardest to identify.

Let Z be the set of all irrelevant vertices of an RBN G.

Definition 2. The reduced network $G|_R=(V|_R,E|_R)$ is a subgraph of G defined by $V|_R=V-Z$, and $E|_R=E-\{(u,v)\in E|\,u\in Z$ or $v\in Z\}$.

In Ref. [18], an algorithm for computing the set of all irrelevant vertices has been presented. This algorithm is computationally expensive and therefore is feasible for RBNs with up to a thousand vertices only. The *decimation procedure* presented in Ref. [19] computes only a subset of irrelevant vertices, but it is applicable to large networks. In time linear in the size of an RBN, it finds irrelevant vertices evident from the structure of the network (first and second type). The decimation procedure will not identify the irrelevant vertices whose associated functions have constant values due to the correlation of their input variables (third type).

The independent components of G_R are defined as follows.

Definition 3. Two relevant vertices are in the same component if and only if there is an undirected path between them.

A path is called *undirected* if it ignores the direction of edges. For example, the network in Fig. 2 has two components: $\{v_2,v_5,v_9\}$ and $\{v_1,v_7\}$.

Definition 3 is equivalent to the definition from Ref. [18], which says that "two relevant elements belong to the same module if one of them controls the other one."

Independent components can found in $O(|V|+|E|)$ time, where $|V|$ is the number of vertices and $|E|$ is the number of edges of G_R, using the algorithm [20]. To find a component number i, the function COMPONENTSEARCH(v) is called for a vertex v which has not been assigned to a component yet. COMPONENTSEARCH does nothing if v has been assigned to a component already. Otherwise, COMPONENTSEARCH assigns v to the component i and calls itself recursively for all predecessors and successors of v. The process repeats with the counter i incremented until all vertices of G_R are assigned.

III. COMPUTING ATTRACTORS BY COMPOSITION

In this section, we show that it is possible to compute attractors of a network G compositionally from the attractors

of the connected components of the reduced network G_R.

Let G_A be a connected component of G_R and A_A be an attractor of G_A. An attractor A_A of length L is represented by a sequence of states $\langle \Sigma_0, \Sigma_1, \ldots, \Sigma_{L-1} \rangle$, where $\Sigma_{(i+1) \bmod L}$ is the next state of the state Σ_i, $i \in \{0,1,\ldots,L-1\}$.

The *support set* of an attractor A_A, $\sup(A_A)$, is the set of vertices of G_A. For example, the left-hand side connected component in Fig. 2 has the support set $\{v_2, v_5, v_9\}$.

Definition 4. Given two attractors $A_A = \langle \Sigma_0^A, \Sigma_1^A, \ldots, \Sigma_{L_A}^A \rangle$ and $A_B = \langle \Sigma_0^B, \Sigma_1^B, \ldots, \Sigma_{L_B}^B \rangle$, such that $\sup(A_A) \cap \sup(A_B) = \emptyset$, the composition of A_A and A_B is a set of attractors defined by

$$A_A \circ A_B = \bigcup_{k=0}^{d-1} \{A_k\},$$

where d is the greatest common divisor of L_A and L_B, each attractor A_k is of length m, m is the least common multiple of L_A and L_B, and the ith state of A_k is a concatenation of $(i \bmod L_A)$th state of A_A and $[(i+k) \bmod L_B]$th state of A_B:

$$\Sigma_i^k = \Sigma_{i \bmod L_A}^A \Sigma_{(i+k) \bmod L_B}^B$$

for $k \in \{0,1,\ldots,d-1\}$, $i \in \{0,1,\ldots,m-1\}$ and "mod" is the operation division modulo.

As an example, consider two attractors $A_A = \langle \Sigma_0^A, \Sigma_1^A \rangle$ and $A_B = \langle \Sigma_0^B, \Sigma_1^B, \Sigma_2^B \rangle$. We have $d=1$ and $m=6$, so $A_A \circ A_B = \{A_0\}$, where the states Σ_i^0, $i \in \{0,1,\ldots,5\}$ are defined by

$$\Sigma_0^0 = \Sigma_0^A \Sigma_0^B, \quad \Sigma_3^0 = \Sigma_1^A \Sigma_0^B,$$

$$\Sigma_1^0 = \Sigma_1^A \Sigma_1^B, \quad \Sigma_4^0 = \Sigma_0^A \Sigma_1^B,$$

$$\Sigma_2^0 = \Sigma_0^A \Sigma_2^B, \quad \Sigma_5^0 = \Sigma_1^A \Sigma_2^B.$$

The composition of attractors is extended to the composition of sets of attractors as follows.

Definition 5. Given two sets of attractors $\{A_{11}, A_{12}, \ldots, A_{1L_1}\}$ and $\{A_{21}, A_{22}, \ldots, A_{2L_2}\}$, such that $\sup(A_{1i}) \cap \sup(A_{2j}) = \emptyset$, for all $i \in \{1,2,\ldots,L_1\}$, $j \in \{1,2,\ldots,L_2\}$, the composition of sets is defined by

$$\{A_{11}, A_{12}, \ldots, A_{1L_1}\} \circ \{A_{21}, A_{22}, \ldots, A_{2L_2}\} = \bigcup_{\forall (i_1, i_2) \in \{1,\ldots,L_1\} \times \{1,\ldots,L_2\}} A_{1i1} \circ A_{2i_2},$$

where "×" is the Cartesian product.

Lemma 1. The composition $A_A \circ A_B$ consists of all possible cyclic sequences of states which can be obtained from A_A and A_B.

Proof: By Definition 4, the result of the composition of A_A and A_B is d attractors $\{A_0, A_1, \ldots, A_{d-1}\}$ of length m each, where d is the greatest common divisor of L_A and L_B is m and the least common multiple of L_A and L_B.

Consider any two states of the attractor A_k, Σ_i^k and Σ_j^k, for some i, $j \in \{0,1,\ldots,m-1\}$, $i \neq j$, and some $k \in \{0,1,\ldots,d-1\}$. By Definition 4,

$$\Sigma_i^k = \Sigma_{i \bmod L_A}^A \Sigma_{(i+k) \bmod L_B}^B$$

and

$$\Sigma_j^k = \Sigma_{j \bmod L_A}^A \Sigma_{(j+k) \bmod L_B}^B.$$

We prove that

$$(\Sigma_{i \bmod L_A}^A = \Sigma_{j \bmod L_A}^A) \Rightarrow (\Sigma_{(i+k) \bmod L_B}^B \neq \Sigma_{(j+k) \bmod L_B}^B).$$

If $\Sigma_{i \bmod L_A}^A = \Sigma_{j \bmod L_B}^B$, then we can express j as

$$j = i + XL_A, \tag{1}$$

where X is some constant which satisfies $XL_A < m$.

By substituting j by Eq. (1) in the expression $(j+k) \bmod L_B$, we get

$$(j+k) \bmod L_B = (i + XL_A + k) \bmod L_B. \tag{2}$$

Clearly, if XL_A is not evenly divisible by L_B, then the right-hand side of the expression (2) is not equal to $(i+k) \bmod L_B$. On the other hand, XL_A cannot be evenly divisible by L_B, because $L_A \neq L_B$ and $XL_A < m$. Thus

$$(i + XL_A + k) \bmod L_B \neq (i+k) \bmod L_B$$

and therefore the states $\Sigma_{(j+k) \bmod L_B}^B$ and $\Sigma_{(i+k) \bmod L_B}^B$ are different. Similarly, we can show that

$$(\Sigma_{(i+k) \bmod L_B}^B = \Sigma_{(j+k) \bmod L_B}^B) \Rightarrow (\Sigma_{i \bmod L_A}^A \neq \Sigma_{j \bmod L_A}^A).$$

Therefore, for a given $k \in \{0,1,\ldots,d-1\}$, no two states in the attractor A_k are equal.

Similarly to the above, we can show that no two states in two different attractors can be the same. If the first parts of two states are the same, than the second parts differ due to the property

$$(k + XL_A) \bmod L_B \neq 0$$

for any $k \in \{0,1,\ldots,d-1\}$.

There are $L_A L_B$ different pairs of indexes in the Cartesian product $\{1,\ldots,L_A\} \times \{1,\ldots,L_B\}$. Thus, since $L_A L_B = md$, at least d attractors of length m are necessary to represent all possible combinations. Since no two states of $A_0, A_1, \ldots, A_{d-1}$ are the same, exactly d attractors of length m are sufficient to represent all possible combinations. □

Let $\{G_1, G_2, \ldots, G_p\}$ be the set of components of G_R. Throughout the rest of the section, we use N_i to denote the number of attractors of G_i, A_{ij} to denote jth attractor G_i, and L_{ij} to denote the length of A_{ij}, $i = \{1,2,\ldots,p\}$, $j = \{1,2,\ldots,N_i\}$.

Let $I = I_1 \times I_2 \times \ldots \times I_p$ be the Cartesian product of sets $I_i = \{i_1, i_2, \ldots, i_{N_i}\}$, where p is the number of components of G_R. The set I_i represents indexes of attractors of the component G_i. For example, if $N_i = 3$, then G_i has three attractors: A_{i1}, A_{i2}, and A_{i3}. The set I_i is then $I_i = \{1,2,3\}$. The set I enumerates all possible elements of the sets I_i. For example, if $p=2$, $N_1=2$, and $N_2=3$, then $I = \{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3)\}$.

Theorem 1. The set of attractors $\mathbf{A}$ of the reduced network G_R with p components can be computed as

$$\mathbf{A} = \bigcup_{\forall (i_1,\ldots,i_p) \in I} [(A_{1i_1} \circ A_{2i_2}) \circ \{A_{3i_3}\}] \cdots \circ \{A_{pi_p}\}.$$

Proof (1) The state space of any component is partitioned

into basins of attraction. There are no common states between different basins of attraction. Thus different attractors of the same component have no common states.

(2) Since in any pair of components (G_i, G_j), $i,j = \{1,2,\ldots,p\}$, $i \neq j$, G_i and G_j do not have vertices in common, the support sets of attractors of G_i and G_j do not intersect. Thus different attractors of different components have no common states.

(3) The set I enumerates all possible combinators of p-tuple of indexes of attractors of components. By definition of the Cartesian product, every p-tuple of I differ at least in one position.

(4) From (1), (2), and (3) we can conclude that the set of attractors obtained by the composition $[(A_{1i_1} \circ A_{2i_2}) \circ \{A_{3i_3}\}] \cdots \circ \{A_{pi_p}\}$ for a given $(i_1,\ldots,i_p) \in I$, differs from the set of attractors obtained for any other p-tuple $(i'_1,\ldots,i'_p) \in I$.

(5) From Lemma 1, we know that the composition $A_{1i_1} \circ A_{2i_2}$ represents all possible cyclic sequences of states which can be obtained from A_{1i_1} and A_{2i_2}. We can iteratively apply Lemma 1 to the result of $A_{1i_1} \circ A_{2i_2}$ composed with A_{3i_3}, etc., to show that the composition $[(A_{1i_1} \circ A_{2i_2}) \circ \{A_{3i_3}\}] \cdots \circ \{A_{pi_p}\}$ represents all possible attractors which can be obtained from p attractors A_{ji_1}, $j = \{1,2,\ldots,p\}$.

(6) From (4) and (5) we can conclude that the union of compositions over all p-tuples of I represents the attractors of G_R. □

The following results follow directly from the Theorem 1.

Lemma 2. The total number of attractors in the reduced network G_R with p components is given by

$$N = \sum_{\forall (i_1,\ldots,i_p) \in I} \prod_{j=2}^{p} \{[(L_{1i_1} \star L_{2i_2}) \star L_{3i_3}] \cdots \star L_{j-1 i_{j-1}}\} \diamondsuit L_{ji_j}$$

where "$\star$" is the least common multiple operation and "$\diamondsuit$" is the greatest common divisor operation.

Lemma 3. The maximum length of attractors in the reduced network G_R is given by

$$L_{\max} = \max_{\forall (i_1,\ldots,i_p) \in I} ((L_{1i_1} \star L_{2i_2}) \star L_{3i_3}) \cdots \star L_{pi_p},$$

where "$\star$" is the least common multiple operation.

By Definition 1, by removing irrelevant vertices we do not change the total number and the maximum length of attractors of an RBN. Therefore N and $L_{\max}$ given by Lemmas 2 and 3 are the same for the original network G.

Results similar to Lemma 2 and 3 have been presented by Bastola and Parisi in Ref. [18] without a proof. It was correctly observed that the maximum attractor length equals to the least common multiple of the maximum lengths of the cycles that compose it. However, the total number of attractors was said to be equal to the maximum common divisor of the maximum lengths of the cycles that compose it, which is incorrect.

As an example, consider the network in Fig. 2 with two components: $G_1 = \{v_2, v_5, v_9\}$ and $G_2 = \{v_1, v_7\}$. Their state spaces are shown in Fig. 3. The first component has two attractors: $A_{11} = \langle 011, 100 \rangle$ of length $L_{11} = 2$ and $A_{12} = \langle 000, 001, 101, 111, 110, 010 \rangle$ of length $L_{12} = 6$. The second component has one attractor $A_{21} = \langle 00, 10, 11, 01 \rangle$ of length $L_{21} = 4$.

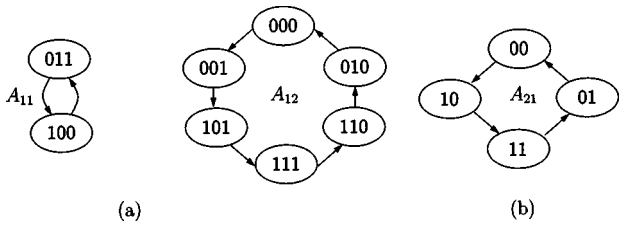

FIG. 3. (a) State space of the component $G_1 = \{v_2, v_5, v_9\}$. There are two attractors, $A_{11} = \langle 011, 100 \rangle$ and $A_{12} = \langle 000, 001, 101, 111, 110, 010 \rangle$. (b) State space of the component $G_2 = \{v_1, v_7\}$. There is one attractor, $A_{21} = \langle 00, 10, 11, 01 \rangle$.

The Cartesian product of $I_1 = \{1,2\}$ and $I_2 = \{1\}$ contains two pairs: $I = \{(1,1),(2,1)\}$. For the pair (1,1) we have $L_{11} \diamondsuit L_{21} = 2 \diamondsuit 4 = 2$ and $L_{11} \star L_{21} = 2 \star 4 = 4$. So, A_{11} and A_{21} compose into two attractors of length 4:

$$A_{11} \circ A_{21} = \{\langle 01100, 10010, 01111, 10001 \rangle, \langle 01110, 10011, 01101, 10000 \rangle\}.$$

The order of vertices in the states is v_2, v_5, v_9, v_1, v_7.

Similarly, for the pair (2, 1) we have $L_{12} \diamondsuit L_{21} = 6 \diamondsuit 4 = 2$ and $L_{12} \star L_{21} = 6 \star 4 = 12$. So, A_{12} and A_{21} compose into two attractors of length 12:

$$A_{12} \circ A_{21} = \{\langle 00000, 00110, 10111, 11101, 11000, 01010, 00011, 00101, 10100, 11110, 11011, 01001 \rangle, \langle 00010, 00111, 10101, 11100, 11010, 01011, 00001, 00100, 10110, 11111, 11001, 01000 \rangle\}.$$

The total number of attractors is $N = 4$. The maximum attractor length is $L_{\max} = 12$.

IV. SUMMARY

In this paper, we show that attractors of an RBN can be computed compositionally from the attractors of the independent components of the subgraph induced by relevant vertices of the network. Previously, for networks whose state spaces were too large to examine exhaustively, the median instead of the exact values of the number of attractors and their length were measured. The presented compositional approach allows us to obtain exact results for larger networks.

In general, independent components occur in RBNs as a result of network's finite connectivity and because the Boolean function assigned to a vertex may not depend on all its inputs. Due to these two factors, some vertices do not control any other vertex in the network. Such vertices serve as "barriers" which prevent exchange of information among the components. It would be interesting to find a relation between the components of an RBN and the biological systems

which RBNs are intended to model, e.g., gene regulatory networks.

Future work includes deriving formulas for the average number and size of the components, as well as the average vertex-to-vertex distance. Another interesting possibility is to apply *redundancy removal* techniques used in logic synthesis [21] to speed up the algorithm for finding irrelevant vertices from Ref. [18].

[1] S. A. Kaufmann, J. Theor. Biol. **22**, 437 (1969).
[2] H. Atlan, F. Fogelman-Soulie, J. Salomon, and G. Weisbuch, Cybernetics and System **12**, 103 (2001).
[3] M. Aldana, S. Coopersmith, and L. P. Kadanoff, adap-org/9305001.
[4] B. Derrida and Y. Pomeau, Biophys. Lett. **1**, 45 (1986).
[5] B. Derrida and H. Flyvbjerg, J. Phys. A **19**, L1103 (1986).
[6] B. Derrida and H. Flyvbjerg, J. Phys. A **20**, L1107 (1987).
[7] S. A. Kaufmann, J. Theor. Biol. **44**, 135 (1974).
[8] S. A. Kaufmann, Physica D **10**, 145 (1984).
[9] S. A. Kaufmann, Physica D **42**, 145 (1990).
[10] S. A. Kaufmann, *Origins of Order* (Oxford University Press, Oxford, 1993).
[11] B. Derrida and D. Stauffer, Europhys. Lett. **1**, 739 (1986).
[12] H. Flyvbjerg, J. Phys. A **21**, L955 (1988).
[13] U. Bastola and G. Parisi, Physica D **115**, 203 (1998).
[14] H. Flyvbjerg and N. J. Kjaer, J. Phys. A **21**, 1695 (1988).
[15] U. Bastola and G. Parisi, J. Theor. Biol. **187**, 117 (1997).
[16] B. Luque and R. Sole, J. Phys. A **31**, 1533 (1998).
[17] J. E. S. Socolar and S. A. Kaufmann, cond-mat/0212306.
[18] U. Bastola and G. Parisi, Physica D **115**, 219 (1998).
[19] S. Bilke and F. Sjunnesson, Phys. Rev. E **65**, 016129 (2001).
[20] R. E. Tarjan, SIAM J. Comput. **1**, 146 (1972).
[21] M. Berkelaar and K. M. van Eijk, in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition* (IEEE Computer Society Press, 2002), p. 1088.