Physica D 69 (1993) 201–208
North-Holland

SDI: 0167-2789(93)E0219-2

PHYSICA D

Antichaos in a class of random boolean cellular automata

James F. Lynch[1]

Department of Mathematics and Computer Science, Clarkson University, Potsdam, NY 13699-5815, USA

Received 14 September 1992
Revised manuscript received 7 June 1993
Accepted 9 June 1993
Communicated by A.C. Newell

A variant of Kauffman's model of cellular metabolism is presented. It is a randomly generated network of boolean gates, identical to Kauffman's except for a small bias in favor of boolean gates that depend on at most one input. The bias is asymptotic to 0 as the number of gates increases. Upper bounds on the time until the network reaches a state cycle and the size of the state cycle, as functions of the number of gates n, are derived. For any $c > 0$, if the bias approaches 0 slowly enough, the state cycles will be smaller than n^c.

1. Introduction

Let n be a natural number. A *boolean cellular automaton* with n gates consists of a directed graph D with vertices $1, \ldots, n$ (referred to as *gates*) and a sequence $b = (b_1, \ldots, b_n)$ of boolean functions. The number of arguments of each function b_i is the same as the indegree of gate i. We say that gate j is an *input* to gate i if (j, i) is an edge of D. A boolean cellular automaton $B = \langle D, b \rangle$ defines a map from $\{0, 1\}^n$ (the set of 0–1 sequences of length n) to $\{0, 1\}^n$ in the following way. For each $i = 1, \ldots, n$ let $j_{i1}, \ldots, j_{ik_i}$ be the inputs of i in increasing order. Given $x = (x_1, \ldots, x_n) \in \{0, 1\}^n$, $B(x) = (b_1(x_{j_{11}}, \ldots, x_{j_{1k_1}}), \ldots, b_n(x_{j_{n1}}, \ldots, x_{j_{nk_n}}))$. B may be regarded as a finite state automaton with state set $\{0, 1\}^n$ and initial state x. That is, its state at time 0 is x, and if its state at time t is $y \in \{0, 1\}^n$ then its state at time $t + 1$ is $B(y)$. We put $B^t(x)$ for the state of B at time t, and $b_i^t(x)$ for the value of its ith component, or gate, at time t. Since the number of states is finite, i.e. 2^n, there exist times t_0 and t_1 such that $B^{t_0}(x) = B^{t_1}(x)$. Let t_1 be the first time at which this occurs. Then $B^{t+t_1-t_0}(x) = B^t$ for all $t \geq t_0$. We refer to the set of states $\{B^t(x) : t \geq t_0\}$ as the *state cycle* of x in $\langle D, b \rangle$, to distinguish it from a cycle of D in the graph-theoretic sense. The *tail* of x in $\langle D, b \rangle$ is $\{B^t(x) : t < t_0\}$.

Kauffman [3] proposed boolean cellular automata as models of cellular metabolism. The gates represent genes within a cell, the state of a gate indicates whether the gene is active or inactive, and $\langle D, b \rangle$ describes how the activity of genes affects other genes. State cycles correspond to the possible behavior patterns that the cell can differentiate into.

In an extensive series of articles (see for example [3–5]), Kauffman described computer simulations on a particular kind of randomly constructed boolean cellular automaton. The number of gates n was fixed, typically in the range of several hundred to several thousand. Every gate had exactly two inputs, chosen independently with equal probability from the $\binom{n}{2}$

[1] Research supported by NSF Grant CCR-9006303.

possibilities. Each gate was also assigned a random boolean function on its two inputs from the 16 equally likely possibilities. Then a random starting state was chosen, again with a uniform distribution on $\{0,1\}^n$. The automaton was simulated, and the trajectory of its state at discrete time intervals was observed.

A striking feature of the behavior of these random boolean cellular automata was their stability. Typically, the tail length and the size of the state cycle were quite small compared to n. Kauffman estimated the median size of the state cycle was on the order of $n^{1/2}$. Also, the automata were very robust. Perturbing the state by flipping the value of one gate usually did not affect the state cycle that was entered.

The behavior of Kauffman's automata are in marked contrast to that of random mappings. It is well known (see e.g. Harris [2]) that the average cycle size and tail length of a random mapping on m points are both asymptotic to $(\frac{1}{8}\pi m)^{1/2}$. It is not difficult to define classes of random boolean cellular automata that behave like random mappings. One way is to construct a network of n gates, where the random functions assigned to the gates have n arguments. In this case, the average state cycle size will be on the order of $2^{n/2}$. The phenomenon where the average state cycle size is superpolynomial in n will be referred to as chaos. On the other hand, the situation when the average state cycle size is bounded by a polynomial in n will be called antichaos. Kauffman has conjectured that his class exhibits antichaotic behavior. We will prove that the class defined in this article is antichaotic.

Although Kauffman's model has been studied empirically for many years, it was only recently that formal mathematical methods have been applied to it (see for example [1,6]). Łuczak and Cohen appear to be the first to prove one of Kauffman's claims – that perturbation of a single gate almost never affects the state cycle that the automaton enters. They also derived a $2^{o(n)}$ upper bound on the size of the state cycle.

In this article, we study a random boolean cellular automaton that is almost the same as Kauffman's (we shall make this precise shortly). Essentially, when boolean functions are assigned to the gates, we give a very small bias to those functions that depend on at most one argument. Depending on the strength of the bias, we will derive polynomial upper bounds on the size of the state cycle entered, and even sublinear upper bounds. We also obtain sublinear bounds on the tail length.

2. Preliminaries

We will use the following notions, introduced by Kauffman [5].

Definition 2.1. Let $f: \{0,1\}^k \to \{0,1\}$ be a boolean function of k arguments. Then f is said to be *canalyzing* if there is some $m = 1, \ldots, k$ and some values $u, v \in \{0,1\}$ such that for all $x \in \{0,1\}^k$, if $x_m = u$ then $f(x) = v$. Argument m of f is said to be a *forcing argument* with *forcing value* u and *forced value* v. Likewise, if $\langle D, b\rangle$ is a boolean cellular automaton and b_i is a canalyzing function with forcing argument m, forcing value u and forced value v, then input j_{im} is a *forcing input* of gate i. That is, if the value of j_{im} is u at time t, then the value of i is guaranteed to be v at time $t+1$.

Note that all four 1-input boolean functions are trivially canalyzing, and all of the 2-input boolean functions except equivalence and exclusive or are canalyzing.

The next definition is borrowed partly from Kauffman [5] and Łuczak and Cohen [6].

Definition 2.2. Again $\langle D, b\rangle$ is a boolean cellular automaton. Using induction on t, we define what it means for gate i to be *forced to a value v in t steps*.

If b_i is the constant function v, then i is forced to v in 0 steps.

If all inputs $j_{i1}, \ldots, j_{ik}$ of i are forced to $u_1, \ldots, u_k$ respectively in t steps then i is forced to $b_i(u_1, \ldots, u_k)$ in $t+1$ steps.

If b_i is a canalyzing function with forcing argument m, forcing value u, and forced value v, and j_{im} is forced to u in t steps, then i is forced to v in $t+1$ steps.

By induction on t it can be seen that if i is forced in t steps, then it stabilizes for all initial states x in t steps. That is, for all $t' \geq t$, $b_i^{t'}(x) = b_i^t(x)$. However, the converse is not true. It is easy to construct boolean cellular automata without any forced gates but with gates that stabilize for all initial states (see fig. 1).

The next definition is due to Łuczak and Cohen [6].

Definition 2.3. For any gate i in $\langle D, b\rangle$, let

$$N_0^-(i) = \{i\}\,,$$

$$N_{d+1}^-(i) = \bigcup \{N_d^-(j) \colon j \text{ is an input to } i\}\,.$$

Then

$$S_d^-(i) = \bigcup_{c \leq d} N_c^-(i)\,.$$

Note that whether i is forced in d steps is completely determined by the restriction of D and b to $S_d^-(i)$.

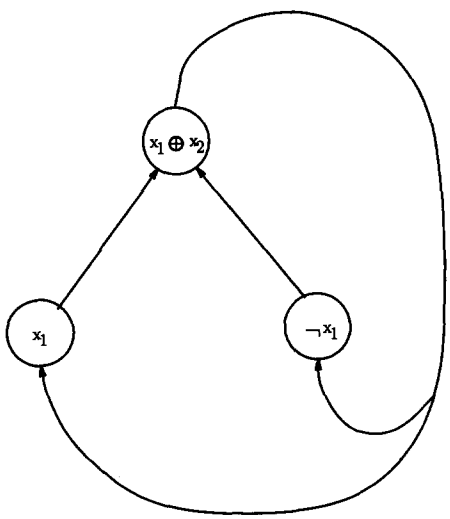

Fig. 1. An example of a boolean cellular automaton without any forced gates, all of whose gates become stable. The labels in the circles denote the boolean functions assigned to the gates: $\oplus$ is exclusive or, and $\neg$ is negation.

The class of *random* boolean cellular automata studied in this paper is the same as Kauffman's except that the probabilities of assigning boolean functions to gates are slightly biased in favor of the six functions that depend on at most one argument:

$$f(x_1, x_2) = x_1\,,$$
$$f(x_1, x_2) = \neg x_1\,,$$
$$f(x_1, x_2) = x_2\,,$$
$$f(x_1, x_2) = \neg x_2\,,$$
$$f(x_1, x_2) = 0\,,$$
$$f(x_1, x_2) = 1\,.$$

Let $\varepsilon(n)$ be a function on the natural numbers such that $0 \leq \varepsilon(n)$ for all n and $\lim_{n\to\infty} \varepsilon(n) = 0$. (We will impose further conditions on ε later.) A directed graph D with n gates is generated as in Kauffman's model, and the sequence b of boolean functions is generated using the distribution

probability that $b_i = f$ is

$$\begin{cases} \frac{1}{16} - \frac{1}{16}\varepsilon(n) \text{ if } f \text{ depends on both arguments}\,, \\ \frac{1}{16} + \frac{1}{16}\varepsilon(n) \text{ if } f \text{ depends on one argument}\,, \\ \frac{1}{16} + \frac{3}{16}\varepsilon(n) \text{ if } f \text{ is a constant function}\,. \end{cases}$$

The particular coefficients of $\varepsilon(n)$ in the above definition were chosen to simplify the notation in our proofs. Our theorems apply to any class of random boolean cellular automata where the probability of assigning f to gate i is $\frac{1}{16} - a_f\varepsilon(n)$ for some positive constant a_f when f depends on both arguments, and the probability is $\frac{1}{16} + a_f\varepsilon(n)$ otherwise. The proofs for the general case are more cumbersome, but they do not involve any important additional ideas. We will use $\tilde{B} = \langle \tilde{D}, \tilde{b}\rangle$ to denote a random boolean cellular automaton generated as above. For any properties $\mathcal{P}$ and $\mathcal{Q}$ pertaining to boolean cellular automata, we put $\mathrm{pr}(\mathcal{P}, n)$ for the probability that the random boolean cellular automaton on n

gates has property $\mathcal{P}$ and $\mathrm{pr}(\mathcal{P} \mid \mathcal{Q}, n)$ for the conditional probability that $\mathcal{P}$ holds, given that $\mathcal{Q}$ holds. Usually, we will omit the n in these expressions since it will be understood.

The properties of $\langle D, b\rangle$ that we are really interested in depend only on the function of $\{0,1\}^n$ computed by $\langle D, b\rangle$. What this means is that if j_1 and j_2 are the two inputs of a gate i and $b_i = f$, where f depends on only one argument, say $f(x_1, x_2) = x_1$, then we can delete the edge (j_2, i) from D and replace f by the one argument function $g(x_1) = x_1$. In this case we say that the input of i is j_1 and $b_i = g$. If f is a constant function, we could delete both input edges, but for our purposes it is simpler just to delete the second input edge and regard f as a constant function of x_1. This will make our proofs easier, and is the motivation for defining a slightly different class of random boolean cellular automata, which is essentially equivalent to the class just defined.

We take ε and n as before. For $i = 1, \ldots, n$ independently, gate i will have two inputs with probability $1 - \varepsilon(n)$ and it will have one input with probability $\varepsilon(n)$. If it has two inputs then the inputs and the boolean function assigned to it are chosen as in Kauffman's model. If it has one input then the input is chosen with uniform probability from among the n gates, and its boolean function is chosen from the four possibilities with equal probability. We will use $\tilde{B}' = \langle \tilde{D}', \tilde{b}'\rangle$ to denote a random boolean cellular automaton generated this way, and pr′ for its associated probability function. The following proposition shows that these two models of random boolean cellular automata have the same probability distribution with respect to the functions that they compute.

Proposition 2.4. For any natural number n, let $\tilde{B} = \langle \tilde{D}, \tilde{b}\rangle$ and $\tilde{B}' = \langle \tilde{D}', \tilde{b}'\rangle$ be the two random boolean cellular automata with n gates defined above. Then for any function $F\colon \{0,1\}^n \to \{0,1\}^n$, $\mathrm{pr}(\tilde{B} = F) = \mathrm{pr}'(\tilde{B}' = F)$.

Proof. We need only show the following.

(1) For every gate i, every $1 \le j_1 < j_2 \le n$, and every boolean function f that depends on both arguments,

$$\mathrm{pr}'(\text{inputs of } i \text{ are } j_1,\ j_2 \text{ and } b'_i = f) = \mathrm{pr}(\text{inputs of } i \text{ are } j_1,\ j_2 \text{ and } b_i = f)\,.$$

(2) For every gate i every $1 \le j \le n$, and every boolean function g of one argument,

$$\mathrm{pr}'(\text{input of } i \text{ is } j \text{ and } b'_i = g) = \mathrm{pr}(\text{input of } i \text{ is } j \text{ and } b_i = g)\,.$$

(3) For every gate i and constant boolean function g,

$$\mathrm{pr}'(b'_i = g) = \mathrm{pr}(b_i = g)\,.$$

To prove (1),

$$\begin{aligned}\mathrm{pr}'&(\text{inputs of } i \text{ are } j_1,\ j_2 \text{ and } b'_i = f)\\ &= [1 - \varepsilon(n)] \times \frac{1}{\binom{n}{2}} \times \frac{1}{16}\\ &= \frac{1}{\binom{n}{2}} \times \left(\frac{1}{16} - \frac{\varepsilon(n)}{16}\right)\\ &= \mathrm{pr}(\text{inputs of } i \text{ are } j_1,\ j_2 \text{ and } b_i = f)\,.\end{aligned}$$

To prove (2),

$$\begin{aligned}\mathrm{pr}'&(\text{input of } i \text{ is } j \text{ and } b'_i = g)\\ &= [1 - \varepsilon(n)] \times \frac{n-1}{\binom{n}{2}} \times \frac{1}{16} + \varepsilon(n) \times \frac{1}{n} \times \frac{1}{4}\\ &= \frac{2}{n} \times \left(\frac{1}{16} + \frac{\varepsilon(n)}{16}\right)\\ &= \mathrm{pr}(\text{input of } i \text{ is } j \text{ and } b_i = g)\,.\end{aligned}$$

To prove (3)

$$\begin{aligned}\mathrm{pr}'(b'_i = g) &= (1 - \varepsilon(n)) \times \tfrac{1}{16} + \varepsilon(n) \times \tfrac{1}{4}\\ &= \frac{1}{16} + \frac{3\varepsilon(n)}{16}\\ &= \mathrm{pr}(b_i = g)\,.\end{aligned}$$

□

Thus we make no distinction between $\langle \tilde{D}, \tilde{b}\rangle$ and $\langle \tilde{D}', \tilde{b}'\rangle$.

The following lemma is essentially a generalization of Lemma 1 in [6].

Lemma 2.5. For sufficiently large n, any natural

number d, and any gate i in the random boolean cellular automaton $\langle \tilde{D}, \tilde{b} \rangle$ with n gates,

$$\begin{aligned}&\mathrm{pr}(i \text{ is not forced in } d \text{ steps} \mid \\ &\quad S_d^-(i) \text{ induces an acyclic subgraph of } \tilde{D}) \\ &\quad \le \frac{16}{d}\,.\end{aligned}$$

Proof. Let p_d be the conditional probability in question. We first show that it satisfies the following recurrence:

$$p_0 = \tfrac{7}{8} - \tfrac{3}{8}\varepsilon(n)\,,$$
$$p_{d+1} = [1 - \tfrac{1}{2}\varepsilon(n)]p_d - \tfrac{1}{8}[1 - \varepsilon(n)]p_d^2\,.$$

The base case for p_0 follows from the fact that two out of the 16 2-input boolean functions are constant while two out of the four of the 1-input boolean functions are constant. Thus $p_0 = (1 - \varepsilon(n)) \cdot \frac{7}{8} + \varepsilon(n) \cdot \frac{1}{2}$.

To prove the induction step, first take the case when i has two inputs. Then i is not forced in $d+1$ steps if and only if neither of its inputs is forced in d steps and b_i is not constant, or exactly one of its inputs is forced but its forced value is not forcing for i. The first possibility has probability $p_d^2 \times \frac{7}{8}$, while the second has probability $2p_d(1-p_d) \times \frac{1}{2}$. If i has one input, then it is not forced if and only if the input is not forced and b_i is not constant. This probability is $p_d \times \frac{1}{2}$. Altogether we have

$$\begin{aligned}p_{d+1} &= [1 - \varepsilon(n)][\tfrac{7}{8}p_d^2 + p_d(1-p_d)] + \tfrac{1}{2}\varepsilon(n)p_d \\ &= [1 - \tfrac{1}{2}\varepsilon(n)]p_d - \tfrac{1}{8}[1 - \varepsilon(n)]p_d^2\,.\end{aligned}$$

Let $q_d = 1/p_d$. We will show by induction on d that $q_d \ge \frac{1}{16}d$ for sufficiently large n, from which the lemma follows. Clearly $q_0 \ge 0$.

Assuming $q_d \ge \frac{1}{16}d$, we use our recurrence for p_{d+1}, getting

$$1/q_{d+1} = [1 - \tfrac{1}{2}\varepsilon(n)]/q_d - [1 - \varepsilon(n)]/8q_d^2\,.$$

Rearranging,

$$\begin{aligned}q_{d+1} &= \frac{q_d}{[1 - \frac{1}{2}\varepsilon(n)] - [1 - \varepsilon(n)]/8q_d} \\ &\ge \frac{q_d}{1 - [1 - \varepsilon(n)]/8q_d} \\ &\ge q_d + \tfrac{1}{8}[1 - \varepsilon(n)] \\ &\ge q_d + \tfrac{1}{16} \text{ for sufficiently large } n \text{ since} \\ &\quad \varepsilon(n) \to 0\,.\end{aligned}$$

Thus $q_{d+1} \ge \frac{1}{16}(d+1)$, and the proof is complete. □

Our final basic idea concerns chains of gates that are not likely to stabilize. An *unforced path* is a sequence of distinct gates $P = (i_1, \ldots, i_p)$ such that i_r is an input of i_{r+1} for $1 \le r < p$ and none of the gates are forced in $256/\varepsilon(n)$ steps. An *unforced cycle* is the same except $i_1 = i_p$.

3. Theorems and proofs

In this section, we will prove our upper bounds on the sizes of the tails and state cycles of the random boolean cellular automaton $\langle \tilde{D}, \tilde{b} \rangle$ as a function of ε. First, we prove some upper bounds on the sizes of unforced structures in $\langle \tilde{D}, \tilde{b} \rangle$. Throughout the paper, log will mean $\log_2$.

Lemma 3.1. If $\varepsilon(n) \gg 1/\log n$ then

$$\begin{aligned}&\mathrm{pr}(\langle \tilde{D}, \tilde{b} \rangle \text{ has an unforced path longer than} \\ &\quad 4 \log n/\varepsilon(n)) = o(1)\,.\end{aligned}$$

Proof. Let $l = \lceil 4 \log n/\varepsilon(n) \rceil$. The gates in a path of length l can be chosen in $n(n-1)\ldots(n-l+1) \le n^l$ ways. Having chosen the l gates $i_1, \ldots, i_l$, for $r = 1, \ldots, l$ let $\mathscr{P}_r$ be the event that $i_1, \ldots, i_r$ form an unforced path. Then the probability that $i_1, \ldots, i_l$ actually form an unforced path is bounded above by

$$\prod_{r=2}^{l} \{[1 - \varepsilon(n)]\alpha_r + \varepsilon(n)\beta_r\}\,,$$

where

$\alpha_r = \mathrm{pr}(i_{r-1}$ is an input to i_r and
i_r is not forced in $256/\varepsilon(n)$ steps|
indegree$(i_r) = 2$ and $\mathcal{P}_{r-1})$,

$\beta_r = \mathrm{pr}(i_{r-1}$ is the input to i_r and
i_r is not forced in $256/\varepsilon(n)$ steps|
indegree$(i_r) = 1$ and $\mathcal{P}_{r-1})$.

Clearly $\beta_r = 1/2n$.

Assuming i_r has two inputs and $\mathcal{P}_{r-1}$ holds, let $j_r \neq i_{r-1}$ be the other input of i_r. Let $\mathcal{Q}_r$ be the event that $S^-_{256/\varepsilon(n)-1}(j_r)$ is a tree and $S^-_{256/\varepsilon(n)-1}(j_r) \cap \cup \{S^-_{256/\varepsilon(n)}(i_s) : s < r\} = \emptyset$. Then

$$\alpha_r \leq \left(\frac{(n-1)}{\binom{n}{2}}\right) \times (\gamma_r + \delta_r + \zeta_r),$$

where

$\gamma_r = \mathrm{pr}(\text{not } \mathcal{Q}_r)$,

$\delta_r = \mathrm{pr}(j_r$ not forced in $256/\varepsilon(n) - 1$ steps $|\, \mathcal{Q}_r)$,

$\zeta_r = \mathrm{pr}(i_r$ not forced in $256/\varepsilon(n)$ steps|
$\mathcal{Q}_r$ and j_r is forced in $256/\varepsilon(n) - 1$ steps).

We now get upper bounds on γ_r, δ_r, and ζ_r.

First, $S^-_{256/\varepsilon(n)-1}(j_r)$ is not a tree only if there exist two paths of length at most $256/\varepsilon(n) - 1$ beginning at some common gate and ending at j_r. This probability is bounded above by

$$\sum_{p=0}^{256/\varepsilon(n)-1} n^p \left(\frac{2}{n}\right)^p (p+1) \sum_{q=1}^{256/\varepsilon(n)-1} n^{q-1} \left(\frac{2}{n}\right)^q$$
$$\leq \left(\frac{256}{\varepsilon(n)}\right)^3 2^{512/\varepsilon(n)} n^{-1}$$
$$= o(\varepsilon(n)) \text{ since } 1/\varepsilon(n) \ll \log n.$$

The probability that $S^-_{256/\varepsilon(n)-1}(j_r) \cap \cup \{S^-_{256/\varepsilon(n)}(i_s) : s < r\} \neq \emptyset$ is bounded above by

$$l \times 2^{256/\varepsilon(n)} \times \sum_{p=0}^{256/\varepsilon(n)-1} n^{p-1} \left(\frac{2}{n}\right)^p = o(\varepsilon(n)),$$

so

$$\gamma_r = o(\varepsilon(n)).$$

Assuming Q_r holds, the event that j_r is not forced in $256/\varepsilon(n) - 1$ steps is independent of the events that i_s is forced in $256/\varepsilon(n)$ steps, $s < r$. Therefore by Lemma 2.5,

$$\delta_r \leq \tfrac{16}{255}\varepsilon(n).$$

Finally, $\zeta_r = \frac{1}{2}$, so $\alpha_r \leq n^{-1}[1 + \frac{32}{255}\varepsilon(n) + o(\varepsilon(n))]$. Therefore the probability that $i_1, \ldots, i_l$ form an unforced path is at most

$$(n^{-1}\{[1 - \varepsilon(n)][1 + \tfrac{32}{255}\varepsilon(n) + o(\varepsilon(n))] + \tfrac{1}{2}\varepsilon(n)\})^{l-1}$$
$$\leq n^{-l+1}[1 - \tfrac{1}{4}\varepsilon(n)]^{l-1},$$

and the probability that there exists such $i_1, \ldots, i_l$ is at most $n(1 - \frac{1}{4}\varepsilon(n))^{l-1}$. Since $l = \lceil 4 \log n/\varepsilon(n) \rceil$, this is asymptotic to

$$n\, \mathrm{e}^{-\log n} \to 0. \qquad \square$$

Lemma 3.2. If $\varepsilon(n) \gg 1/\log n$ then

$\mathrm{pr}(\langle \tilde{D}, \tilde{b} \rangle$ has an unforced cycle larger than
$8 \log\log n/\varepsilon(n))$
$= o(1)$.

Proof. By Lemma 3.1, we need consider only cycles of length at most $4 \log n/\varepsilon(n)$. Summing over all cycle sizes from $8 \log\log n/\varepsilon(n)$ to $4 \log n/\varepsilon(n)$, and using the same estimates as in the proof of Lemma 3.1, the probability is bounded above by

$$4 \log n/\varepsilon(n) \times [1 - \tfrac{1}{4}\varepsilon(n)]^{8 \log\log n/\varepsilon(n)}$$
$$\leq 4(\log n)^2 \times \mathrm{e}^{-2 \log\log n}$$
$$\to 0. \qquad \square$$

Lemma 3.3. If $\varepsilon(n) \gg 1/\log n$ then

$\mathrm{pr}(\langle \tilde{D}, \tilde{b} \rangle$ has unforced cycles connected
by an unforced path)
$= o(1)$.

Proof. By Lemmas 3.1 and 3.2, we need consider only cycles of size at most $8 \log\log n/\varepsilon(n)$ and paths of length at most $4 \log n/\varepsilon(n)$. The endpoints of the path can be chosen in at most $[8 \log\log n/\varepsilon(n)]^2$ ways. Summing over all cycle sizes up to $8 \log\log n/\varepsilon(n)$, all possible choices of endpoints of the path, and all paths of length up

to $4 \log n/\varepsilon(n)$, the probability is bounded above by

$$[8 \log \log n/\varepsilon(n)]^4 \times [4 \log n/\varepsilon(n)] \times n^{-1} \to 0 .$$

□

Theorem 3.4. if $\varepsilon(n) \gg 1/\log n$ then

$$\mathrm{pr}(\langle \tilde{D}, \tilde{b} \rangle \text{ has a tail longer than } 9 \log n/\varepsilon(n)) = o(1)$$

Proof. After $256/\varepsilon(n)$ steps, the only gates that are not yet stable are those in unforced paths and cycles. We may assume that all the cycles and paths are disjoint except possibly at the endpoints of the paths. By Lemma 3.3, with probability $1 - o(1)$, no path begins and ends at a cycle (see fig. 2a).

Let l be the length of the longest unforced path in $\langle \tilde{D}, \tilde{b} \rangle$ and m be the size of its largest unforced cycle. After l more steps, the only gates that are not yet stable are those in unforced cycles and paths beginning at an unforced cycle (see fig. 2b).

Now consider the state of the gates in these cycles, i.e., the projection of the state of $\langle \tilde{D}, \tilde{b} \rangle$, where we look only at the values of the gates in the unforced cycles. This state will reach its state cycle in at most $2m$ steps. Then $\langle \tilde{D}, \tilde{b} \rangle$ will reach its state cycle in at most l more steps. The theorem then follows from Lemmas 3.1 and 3.2.

□

Corollary 3.5. If $\varepsilon(n) \gg 1/\log n$ then

$$\lim_{n \to \infty} \mathrm{pr}(\text{all tail lengths of } \langle \tilde{D}, \tilde{b} \rangle \text{ are } o((\log n)^2)) = 1 .$$

Theorem 3.6. If $\epsilon(n) \gg 1/\log n$ then

$$\mathrm{pr}(\langle \tilde{D}, \tilde{b} \rangle \text{ has a state cycle larger than } 2^{12 \log \log n/\varepsilon(n)}) = o(1) .$$

Proof. Any path (or cycle) of unstable gates is also an unforced path (or cycle). By Lemma 3.3, no unforced cycle is connected by an unforced path to an unforced cycle. Therefore no cycle of unstable gates is connected by a path of unstable gates to a cycle of unstable gates. Then the size of any state cycle of $\langle \tilde{D}, \tilde{b} \rangle$ is at most the least common multiple of the periods of all the cycles of unstable gates. A cycle consisting of s unstable gates has period t or $2t$ for some factor t of s. Let m be the size of the largest cycle of unstable gates. Then the size of any state cycle of $\langle \tilde{D}, \tilde{b} \rangle$ is bounded by twice the least common multiple of all the natural numbers less than or equal to m.

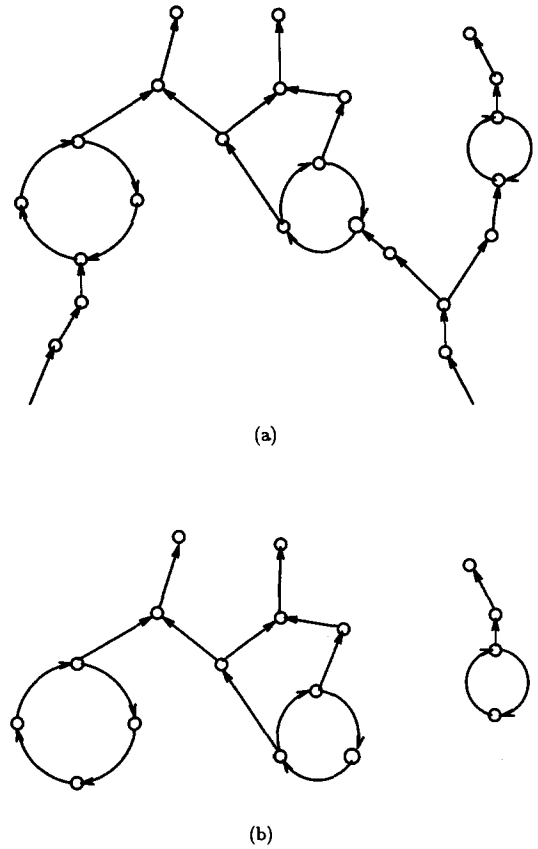

Fig. 2. Typical structure of the unstable gates of $\langle \tilde{D}, \tilde{b} \rangle$. (a) After $256/\varepsilon(n)$ steps. (b) After l more steps, where l is the maximum length of unforced paths that do not begin at an unforced cycle.

By the prime number theorem (see [7]), the number of primes less than or equal to m is asymptotic to $m \log e/\log m$. Then the least common multiple of all the numbers less than or

equal to m is bounded by $m^{1.45m/\log m}$, and the size of any state cycle of $\langle \tilde{D}, \tilde{b} \rangle$ is at most

$$2m^{1.45m/\log m} \leq 2^{12 \log \log m/\varepsilon(n)}$$

by Lemma 3.2. □

Corollary 3.7. Assume $\varepsilon(n) \geq a \log \log n / \log n$ for some constant a. Then there is a constant c such that

$$\mathrm{pr}(\langle \tilde{D}, \tilde{b} \rangle \text{ has a tail longer than } c(\log n)^2/\log \log n) = o(1),$$

$$\mathrm{pr}(\langle \tilde{D}, \tilde{b} \rangle \text{ has a state cycle larger than } n^c) = o(1).$$

In particular, if $a > 12$, then we can take $c < 1$.

4. Discussion

There are a number of problems suggested by our results. An immediate question is whether small upper bounds on tail length and state cycle size can be proven for Kauffman's model. More generally, the effect of using other distributions on the boolean functions should be investigated. In our model, as the distribution approaches the uniform distribution, where each boolean function has probability $\frac{1}{16}$, the upper bounds get larger. While this does not prove anything about a transition to chaos, it raises the possibility that the uniform distribution is a threshold between chaotic and antichaotic behavior. It may be meaningful to group the functions into several categories, such as constant functions, functions depending on one argument, canalyzing functions depending on two arguments, and non-canalyzing functions.

Consequences of using functions with more than two arguments should also be studied. Compared to the two argument functions, the three argument functions have a much smaller proportion of canalyzing functions, and there may be a threshold involving the indegree of the gates. A network with a significant number of three input gates is likely to have more non-canalyzing functions, and these would be less likely to be stable.

Acknowledgement

The author thanks Dr. Alan Woods of the University of Western Australia for ideas and stimulating conversations that improved this article.

References

[1] F. Fogelman-Soulié, Parallel and sequential computation on boolean networks, Theor. Comp. Sci. 40 (1985) 275–300.

[2] B. Harris, Probability distributions related to random mappings, Ann. Math. Stat. 31 (1960) 1045–1062.

[3] S.A. Kauffman, Behaviour of randomly constructed genetic nets: binary element nets, in: Towards a Theoretical Biology, ed. C.H. Waddington (Aldine, Chicago, 1970) pp. 18–37.

[4] S.A. Kauffman, Emergent properties in random complex automata, Physica D 10 (1984) 145–156.

[5] S.A. Kauffman, Requirements for evolvability in complex systems: orderly dynamics and frozen components, Physica D 42 (1990) 135–152.

[6] T. Łuczak and J.E. Cohen, Stability of vertices in random boolean cellular automata, Random Structures Algorithms 2 (1991) 327–334.

[7] E.C. Titchmarsh, The Theory of the Riemann Zeta Function (Oxford Univ. Press. Oxford, 1951).