Exact behavior of the critical Kauffman model with connectivity one

T. M. A. Fink
London Institute for Mathematical Sciences, Royal Institution, 21 Albemarle St, London W1S 4BS, United Kingdom

(Received 31 March 2023; accepted 18 June 2024; published xxxxxxxxxx)

The critical Kauffman model with connectivity one is the simplest class of critical Boolean networks. Nevertheless, it exhibits intricate behavior at the boundary of order and chaos. We show that the model is equivalent to a deceptively simple algebraic system of polynomials which count the number and length of cycles. The polynomial for multiple loops is the product of the polynomials for individual loops. Using this perspective, we prove that the number of cycles scales as 2^m, where m is the number of nodes in loops—as fast as possible and faster than previously believed.

DOI: 10.1103/PhysRevResearch.00.003000

I. INTRODUCTION

Exactly solvable models play a special role in physics, for several reasons. First, they are archetypal, in that they capture the important behavior of a broad class of systems. Second, stripped of extraneous detail, they are amenable to analytic solutions, allowing us to peer under the hood and see what's driving the behavior. Third, they suggest lines of attack for more realistic models that cannot be solved exactly. Fourth, they are a gift that keeps on giving: new approaches to solving them reveal additional structure and insights.

One such model is the critical Kauffman model with connectivity one [1,2]. It is the simplest class of Boolean networks at the boundary of order and chaos, at which many biological systems seem poised. But before we describe it, let's summarize the general Kauffman model.

Introduced as a simple model of genetic computation [3,4], the Kauffman model is a Boolean network in which there are N nodes and each node has exactly K inputs, randomly chosen from the N nodes. Each node is permanently assigned one of the 2^{2^K} possible Boolean functions on K inputs. Then, starting from some initial configuration of 0 s and 1s, at each time step the state of the network is simultaneously updated according to the Boolean functions at the nodes. Because the number of configurations is finite, the network eventually enters into a repeating set of states, or a cycle.

Depending on the choice of K and the Boolean functions, the behavior of a Kauffman model falls into two regimes. In the frozen regime, perturbations die out, and the cycle lengths do not grow with system size. In the chaotic regime, perturbations grow exponentially, and the cycle lengths grow with system size. These regimes are separated by a critical boundary, in which a perturbation to one node propagates to, on average, one other node. This boundary is of particular interest because of the celebrated and controversial hypothesis that life operates at the edge of order and chaos [5,6]. If the Boolean functions are uniformly drawn from those that are possible, then $K = 2$ alone gives criticality; lower K leads to freezing and higher K to chaos.

In a series of advances, researchers honed in on how the number of attractors in the critical regime grows with network size N. The growth rate was first thought to be $\sqrt{N}$ [4], then linear in N [7], faster than linear [8], a stretched exponential [9,10], and faster than any power law [11]. But a definitive answer has remained elusive.

Reducing $K = 2$ to $K = 1$ drastically simplifies the Kauffman model, so much so that the model might seem trivial. The network is composed of loops and trees branching off of loops, as shown in Fig. 1 top. Because the nodes in the trees are enslaved by the loops, they do not contribute to the number or length of cycles, which are set by the m nodes in loops. Each node can have one of four Boolean functions: on, off, copy, and invert. But the critical version of the model requires that the functions be copy or invert, because just one on or off in a loop freezes it, rendering it irrelevant [2,12].

Despite its simplicity, the $K = 1$ critical Kauffman model exhibits startlingly rich and subtle behavior. An exact solution was first laid out in an incisive paper by Flyvbjerg and Kjaer [1]. Later, Drossel, Mihaljev and Greil [2] obtained a more complete, if tersely presented, understanding of the critical behavior by generating networks through a growth process. In this paper we take a new tack, translating the problem into a purely algebraic system. Doing so reveals new quantitative insights and offers a fresh perspective for further research.

This paper is organized as follows. In part 2, we introduce an expression for the number and length of cycles for a single loop, which we call a primitive cycle polynomial. In part 3, we show that the cycle polynomial for multiple loops can be obtained by taking the appropriate product of primitive cycle polynomials. We deduce some useful properties of cycle polynomials in part 4, and in part 5 we calculate the number of cycles for multiple loops by summing the cycle polynomial coefficients. In part 6, we prove that the number of cycles scales as 2^m to first order in m. This is the first proof that

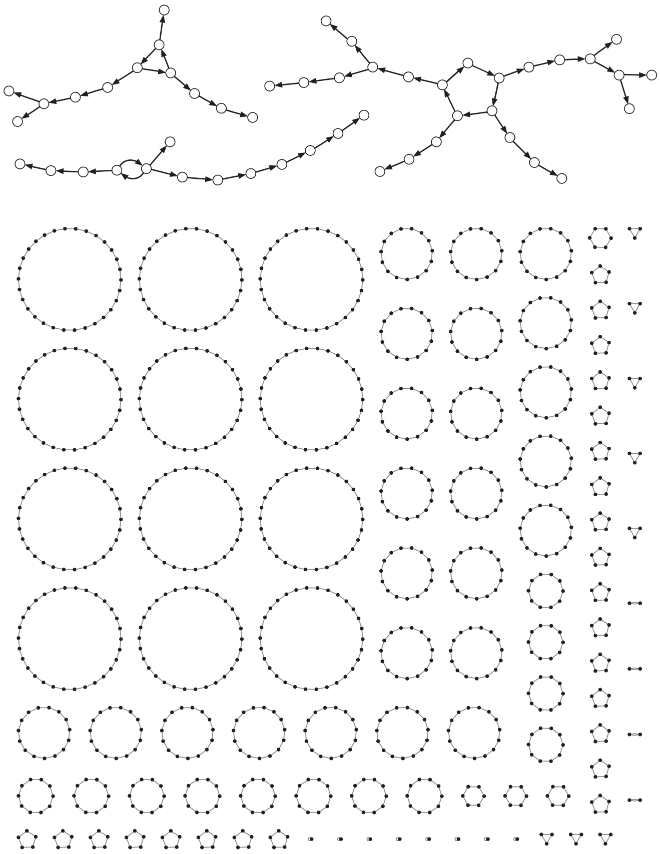

FIG. 1. A Kauffman network and its cycles. (Top) This typical Kauffman network with connectivity one has $N = 50$ nodes and $m = 10$ nodes in loops: a 2-loop, a 3-loop, and a 5-loop. Only the nodes in loops contribute to the cycles. (Bottom) When all of the Boolean functions are copy, the 2^{10} states form 8 cycles of length 1, 4 cycles of length 2, and so on, up to 12 cycles of length 30. We denote this by the cycle polynomial: $D_2D_3D_5 = 8x + 4x^2 + 8x^3 + 24x^5 + 4x^6 + 12x^{10} + 24x^{15} + 12x^{30}$, where D_2, D_3, and D_5 are given in Table I.

the number of cycles grows as fast as possible with m. We discuss our two main take-home messages in part 7, where we also compare our scaling result with the best known bounds of $2^{0.5m}$ and $2^{0.47m}$ [1,2].

While this paper was under review, we were able to translate the growth rate of the number of cycles from a function of loop nodes m to network nodes N. This gives a long-sought answer to how the number of cycles depends on network size, and was reported in late 2023 [13].

II. SINGLE LOOP

We start with a single loop of length l. There are 2^l ways of assigning copy and invert to the l nodes, but these lead to just two behaviors [2,12]. If the number of inverts is even, the number and length of cycles is identical to all of them being copy; this is called an even loop. If the number of inverts is odd, the number and length of cycles is identical to all of them being copy apart from one invert; this is called an odd loop. To be clear, even and odd refers to the parity of the number of inverts, and not the loop size itself.

We can express the number and length of cycles of even and odd loops in terms of the well-known sequences

$$a(k) = \frac{1}{k}\sum_{j|k}\mu(j)2^{k/j} \quad\text{and}\quad b(k) = \frac{1}{2k}\sum_{\text{odd } j|k}\mu(j)2^{k/j}, \qquad (1)$$

where μ is the Möbius function and the first sum is over all j that divide k and the second is over all odd j that divide k. These are described in OEIS A001037 and A000048 [14]. The $a(k)$ are the number of binary Lyndon words of length k, that is, the number of circular binary strings inequivalent up to rotation and not having a period smaller than k. For example, the six Lyndon words of length five are 00001, 00011, 00101, 00111, 01011, and 01111. The $b(k)$ are the number of such Lyndon words with an odd number of ones—or, equivalently, when 0 and 1 can be interchanged.

An even l-loop, indicated by $\{l\}$, has cycles of length k if and only if k divides l; there are $a(k)$ of them. An odd l-loop, indicated by $\{\bar{l}\}$, has cycles of length $2k$ if and only if k divides l and l/k is odd; there are $b(k)$ of them. Let Ax^ν denote A cycles of length ν. Then we can represent the number and length of cycles in a given loop by the cycle polynomials, which we introduce here:

$$D_l(x) = \sum_{k|l} a(k)x^k \quad\text{and}\quad D_{\bar{l}}(x) = \sum_{k|l,\ l/k\ \text{odd}} b(k)x^{2k}$$

(we dropped the braces around l and $\bar{l}$ in D_l and $D_{\bar{l}}$ for convenience). The first eight D_l and $D_{\bar{l}}$ are shown in the left of Table I. We call these primitive cycle polynomials, because, as we shall see, all other cycle polynomials are built out of them, like how the composite numbers are built out of the primes.

We note in passing that, since all 2^l states of the loop belong to cycles,

$$\sum_{k|l} ka(k) = 2^l \quad\text{and}\quad \sum_{k|l,\ l/k\ \text{odd}} 2kb(k) = 2^l.$$

For example, in D_6, $1\cdot 2 + 2\cdot 1 + 3\cdot 2 + 6\cdot 9 = 2^6$.

We can get some intuition for Eq. (1) by observing that it relies on an inclusion-exclusion argument. Consider an even loop of size 6. The number of cycles of length 6 is $(2^6 - 2^3 - 2^2 + 2)/6 = 9$. The first term is the number of binary strings of length 6, while the second and third subtract off strings with period 3 and 2. The last term adds back on strings with period 1 that had been doubly subtracted. Finally, the total is divided by 6 to account for rotations around the loop. The Möbius function μ tracks how many times a divisor has been counted: $\mu(j) = 0$ if j has any repeated prime factors and $\mu(j) = (-1)^i$ if j is the product of i distinct primes.

III. MULTIPLE LOOPS

Multiple loops can give rise to more complex behavior, where the cycle lengths of the set of loops are the least common multiples of the cycle lengths of individual loops. As an example, Fig. 1 shows the cycles resulting from a network containing a 2-loop, a 3-loop, and a 5-loop.

The cycle polynomial for multiple loops can be deduced from the cycle polynomials for individual loops by defining an appropriate product—not the familiar one—between the polynomials. The key observation is that, given A cycles of

TABLE I. Cycle polynomials for single loops and loops of the same size. The primitive cycle polynomial $D_l(x)$ indicates the number and length of cycles in a loop of size l. For example, $D_3(x) = 2x + 2x^3$ reads as two cycles of length one and two cycles of length three. An even parity loop of size l has $a(k)$ cycles of length k if k divides l. An odd parity loop of size l has $b(k)$ cycles of length $2k$ if k divides l and l/k is odd. The cycle polynomial for two loops is given by the product of the individual primitive cycle polynomials, where the product is defined by Eq. (2). Note in particular that $D_l(x)D_{\bar{l}}(x) = D_{\bar{l}}(x)D_{\bar{l}}(x)$.

Even loop	Odd loop	Two even loops	Two odd loops
$D_1 = 2x$	$D_{\bar{1}} = x^2$	$D_1D_1 = 4x$	$D_{\bar{1}}D_{\bar{1}} = 2x^2$
$D_2 = 2x + x^2$	$D_{\bar{2}} = x^4$	$D_2D_2 = 4x + 6x^2$	$D_{\bar{2}}D_{\bar{2}} = 4x^4$
$D_3 = 2x + 2x^3$	$D_{\bar{3}} = x^2 + x^6$	$D_3D_3 = 4x + 20x^3$	$D_{\bar{3}}D_{\bar{3}} = 2x^2 + 10x^6$
$D_4 = 2x + x^2 + 3x^4$	$D_{\bar{4}} = 2x^8$	$D_4D_4 = 4x + 6x^2 + 60x^4$	$D_{\bar{4}}D_{\bar{4}} = 32x^8$
$D_5 = 2x + 6x^5$	$D_{\bar{5}} = x^2 + 3x^{10}$	$D_5D_5 = 4x + 204x^5$	$D_{\bar{5}}D_{\bar{5}} = 2x^2 + 102x^{10}$
$D_6 = 2x + x^2 + 2x^3 + 9x^6$	$D_{\bar{6}} = x^4 + 5x^{12}$	$D_6D_6 = 4x + 6x^2 + 20x^3 + 670x^6$	$D_{\bar{6}}D_{\bar{6}} = 4x^4 + 340x^{12}$
$D_7 = 2x + 18x^7$	$D_{\bar{7}} = x^2 + 9x^{14}$	$D_7D_7 = 4x + 2340x^7$	$D_{\bar{7}}D_{\bar{7}} = 2x^2 + 1170x^{14}$
$D_8 = 2x + x^2 + 3x^4 + 30x^8$	$D_{\bar{8}} = 16x^{16}$	$D_8D_8 = 4x + 6x^2 + 60x^4 + 8160x^8$	$D_{\bar{8}}D_{\bar{8}} = 4096x^{16}$

length ν and B cycles of length ξ, their product is

$$Ax^{\nu} \cdot Bx^{\xi} = AB \gcd(\nu, \xi) x^{\mathrm{lcm}(\nu,\xi)}.$$

Then the product between two cycle polynomials is

$$\sum_i A_i x^{\nu_i} \cdot \sum_j B_j x^{\xi_j} = \sum_{i,j} A_i B_j \gcd(\nu_i, \xi_j) x^{\mathrm{lcm}(\nu_i,\xi_j)}. \qquad (2)$$

For example, the cycle polynomial for two odd 3-loops is

$$\begin{aligned} D^2_{\bar{3}}(x) &= (x^2 + x^6)(x^2 + x^6) \\ &= 2x^2 + 2x^6 + 2x^6 + 6x^6 \\ &= 2x^2 + 10x^6. \end{aligned}$$

The cycle polynomial in Fig. 1 is

$$\begin{aligned} &D_2(x)D_3(x)D_5(x) = (2x + x^2)(2x + 2x^3)(2x + 6x^5) \\ &= 8x + 4x^2 + 8x^3 + 24x^5 + 4x^6 + 12x^{10} + 24x^{15} + 12x^{30}. \end{aligned}$$

More examples of cycle polynomials for multiple loops are given in Table I and Table II.

IV. PROPERTIES OF CYCLE POLYNOMIALS

The cycle polynomials satisfy two properties that will prove useful. Both involve n loops of the same size l, which we call a cluster of l-loops.

The first property is that, in a cluster, odd parity loops are contagious. Consider a cluster with $p \geqslant 0$ even parity loops and $q = n - p \geqslant 0$ odd parity loops, which we denote by $\{l^p, \bar{l}^q\}$. If all the loops in the cluster are even, we call it an even cluster. If one or more loops is odd, then the cycle polynomial is the same as if all loops were odd, and we call it an odd cluster. Specifically, for $q \geqslant 1$,

$$D^p_l(x) D^q_{\bar{l}}(x) = D^{p+q}_{\bar{l}}(x). \qquad (3)$$

For example, with $p = q = 1$,

$$\begin{aligned} D_4D_{\bar{4}} &= (2x + x^2 + 3x^4)(2x^8) \qquad & D_{\bar{4}}D_{\bar{4}} &= (2x^8)(2x^8) \\ &= 4x^8 + 4x^8 + 24x^8 & &= 32x^8 \\ &= 32x^8. \end{aligned}$$

This contagion property can be seen as follows. Consider the n loops as concentric circles, and let u^i_j be the value of the jth node in loop i. Let α_j be the values of a radial cut through the circles: $\alpha_j = u^1_j, \ldots, u^n_j$. Assume all n loops are even, with no inverts. Then α_j is just copied around the loops of the same size. Now assume all loops are even except the first, with a single invert. A first pass around the loops maps all the α_j to $\bar{u}^1_j, \ldots, u^n_j$ (here $\bar{u}$ means not u), which on a second pass is mapped back to α_j. Similar arguments apply for any combination of the loops in which one or more of the loops are odd.

The second property of cycle polynomials is that there is a shortcut for computing the cycle polynomial for a cluster of n loops of the same size. Instead of multiplying out the n polynomials explicitly, we can write the cycle polynomial for

TABLE II. Cycle polynomials for loops of different sizes. Here we show the cycle polynomials for multiple loops of different sizes. They are computed using the product formula in Eq. (2). The cycle polynomials for individual loops, which we call primitive cycle polynomials, are given in Table I. Note that $D_{l_1}D_{\bar{l}_2}$ and $D_{\bar{l}_1}D_{l_2}$ are not in general the same.

Two even loops	Even and odd loops	Odd and even loops	Two odd loops
$D_1D_2 = 4x + 2x^2$	$D_1D_{\bar{2}} = 2x^4$	$D_{\bar{1}}D_2 = 4x^2$	$D_{\bar{1}}D_{\bar{2}} = 2x^4$
$D_1D_3 = 4x + 4x^3$	$D_1D_{\bar{3}} = 2x^2 + 2x^6$	$D_{\bar{1}}D_3 = 2x^2 + 2x^6$	$D_{\bar{1}}D_{\bar{3}} = 2x^2 + 2x^6$
$D_2D_3 = 4x + 2x^2 + 4x^3 + 2x^6$	$D_2D_{\bar{3}} = 4x^2 + 4x^6$	$D_{\bar{2}}D_3 = 2x^4 + 2x^{12}$	$D_{\bar{2}}D_{\bar{3}} = 2x^4 + 2x^{12}$
$D_1D_4 = 4x + 2x^2 + 6x^4$	$D_1D_{\bar{4}} = 4x^8$	$D_{\bar{1}}D_4 = 4x^2 + 6x^4$	$D_{\bar{1}}D_{\bar{4}} = 4x^8$
$D_2D_4 = 4x + 6x^2 + 12x^4$	$D_2D_{\bar{4}} = 8x^8$	$D_{\bar{2}}D_4 = 16x^4$	$D_{\bar{2}}D_{\bar{4}} = 8x^8$
$D_3D_4 = 4x + 2x^2 + 4x^3 + 6x^4 + 2x^6 + 6x^{12}$	$D_3D_{\bar{4}} = 4x^8 + 4x^{24}$	$D_{\bar{3}}D_4 = 4x^2 + 6x^4 + 4x^6 + 6x^{12}$	$D_{\bar{3}}D_{\bar{4}} = 4x^8 + 4x^{24}$

even and odd clusters as

$$D_l^n(x) = \sum_{k|l} a_n(k)x^k \quad \text{and} \quad D_{\bar{l}}^n(x) = \sum_{k|l,\ l/k \text{ odd}} b_n(k)x^{2k},$$

where

$$a_n(k) = \frac{1}{k}\sum_{j|k}\mu(j)2^{nk/j} \text{ and } b_n(k) = \frac{1}{2k}\sum_{\text{odd } j|k}\mu(j)2^{nk/j}.$$

Examples are given in the right half of Table I.

This property can be seen as follows. Again consider the n loops as concentric circles. Since α_j is in one of 2^n states, we can think of the cluster of loops as a single loop in which each node can take 2^n states. The $a_n(k)$ are the number of 2^n-ary Lyndon words of length k, and the $b_n(k)$ are the number of such words when each color can be interchanged with a unique other color. The sequences a_2 and a_3 are described in OEIS A027377 and A027380 [14].

V. NUMBER OF CYCLES

The cycle polynomial for a set of loops contains the number and length of cycles generated by the loops. In particular, we can extract the number of cycles by evaluating the polynomial at $x = 1$, which just sums the coefficients. Consider a collection of loops in which there are s loop sizes and therefore s clusters. For a given cluster, there are n_i loops of size l_i, of which p_i are even and $q_i = n_i - p_i$ are odd. The number of cycles c is

$$c\left(l_1^{p_1}, \bar{l}_1^{q_1} \dots l_s^{p_s}, \bar{l}_s^{q_s}\right) = \left(D_{l_1}^{p_1} D_{\bar{l}_1}^{q_1} \dots D_{l_s}^{p_s} D_{\bar{l}_s}^{q_s}\right)\Big|_{x=1}.$$

Consider two cycle polynomials

$$E = \sum_i A_i x^{\nu_i} \quad \text{and} \quad F = \sum_j B_j x^{\xi_j}.$$

From our product formula in Eq. (2),

$$\begin{aligned}(EF)|_{x=1} &= \sum_{i,j} A_i B_j \gcd(\nu_i, \xi_j)\\ &\geqslant \sum_{i,j} A_i B_j\\ &= E|_{x=1} F|_{x=1}.\end{aligned}$$

Thus we see that the number of cycles is superadditive:

$$c\left(l_1^{p_1}, \bar{l}_1^{q_1} \dots l_s^{p_s}, \bar{l}_s^{q_s}\right) \geqslant c(l_1^{p_1})c\left(\bar{l}_1^{q_1}\right)\dots c\left(l_s^{p_s}\right)c\left(\bar{l}_s^{q_s}\right). \tag{4}$$

For odd clusters, all cycle lengths are even. Since $\gcd(ij, ik) = i\gcd(j, k)$,

$$c\left(\bar{l}_1^{q_1} \dots \bar{l}_s^{q_s}\right) \geqslant 2^{s-1} c\left(\bar{l}_1^{q_1}\right)\dots c\left(\bar{l}_s^{q_s}\right). \tag{5}$$

Focusing on a single loop,

$$\begin{aligned}c(l) &= \sum_{k|l} a(k)\\ &= \sum_{k|l}\frac{1}{k}\sum_{j|k}\mu(j)2^{k/j}\\ &= \frac{1}{l}\sum_{k|l}\phi(k)2^{l/k},\end{aligned}$$

where $\phi(k)$ is the Euler totient function: $\phi(k)$ counts the numbers up to k that are relatively prime to k. The last step makes use of the standard Dirichlet convolution identity, $\phi(k) = \sum_{j|k} j\,\mu(k/j)$.

Using similar arguments, we can write down the number of cycles for even and odd clusters:

$$c(l^n) = \frac{1}{l}\sum_{k|l}\phi(k)2^{nl/k} \quad \text{and} \quad c(\bar{l}^n) = \frac{1}{2l}\sum_{\text{odd } k|l}\phi(k)2^{nl/k}.$$

For $n = 1$, these are described in OEIS A000031 and A000016 [14]. Taking just the $k = 1$ term gives the following good bounds, which we will use later:

$$c(l^n) > 2^{nl}/l \quad \text{and} \quad c(\bar{l}^n) > 2^{nl}/(2l). \tag{6}$$

VI. MINIMUM NUMBER OF CYCLES

Equipped with the above results, we can now calculate the minimum number of cycles for m nodes in a set of loops L. We divide the s clusters into two categories: those in which one or more of the loops is odd—of which there are some number r—and those in which they are all even:

$$c(L) = c\left(l_1^{p_1}, \bar{l}_1^{q_1}, \dots, l_r^{p_r}, \bar{l}_r^{q_r}, l_{r+1}^{n_{r+1}}, \dots, l_s^{n_s}\right).$$

Assume at least one of the clusters in L is odd (we will deal with the alternative case below). By the contagion property in Eq. (3),

$$c(L) = c\left(\bar{l}_1^{n_1}, \dots, \bar{l}_r^{n_r}, l_{r+1}^{n_{r+1}}, \dots, l_s^{n_s}\right).$$

Applying the inequalities in Eqs. (4) and (5),

$$c(L) \geqslant 2^{r-1} c\left(\bar{l}_1^{n_1}\right)\dots c\left(\bar{l}_r^{n_r}\right)c\left(l_{r+1}^{n_{r+1}}\right)\dots c\left(l_s^{n_s}\right).$$

Using the bounds in Eq. (6),

$$c(L) > \frac{1}{2}\prod_{i=1}^{s}\frac{2^{n_i l_i}}{l_i}.$$

Since $\sum_{i=1}^{s} n_i l_i = m$,

$$c(L) > 2^{m-1}\prod_{i=1}^{s}\frac{1}{l_i}. \tag{7}$$

If no cluster in L is odd, the bound is twice this.

To minimize the right side of Eq. (7), we want to maximize the product of the l_i, which occurs when the n_i are all 1, that is, $\sum_{i=1}^{s} l_i = m$. We want the distinct l_i as small as possible but greater than 1. For $m = \frac{(s+1)(s+2)}{2} - 1$, the optimal choice of the l_i is $2, 3, \dots, s+1$. As m increases, this sequence progresses by incrementing one element at a time, from right to left. The process restarts after the leftmost element is incremented. For example, for $s = 3$ and $m = 9$, the progression is: 2,3,4; 2,3,5; 2,4,5; 3,4,5; 3,4,6; and so on. When m reaches $\frac{(s+2)(s+3)}{2} - 1$, the number of l_i increases from s to $s+1$. Thus $\prod l_i$ is at most $\prod_{i=2}^{s+1} i$ for $\frac{s(s+1)}{2} - 1 < m \leqslant \frac{(s+1)(s+2)}{2} - 1$.

Returning to Eq. (7), with the l_i set to $2, 3, \dots, s+1$,

$$c(L) > 2^{m-1}/(s+1)!.$$

Since $m > \frac{s(s+1)}{2} - 1$, $s < \frac{\sqrt{8m+9}-1}{2} < \sqrt{2m}$ (the latter for $m > 1$). Since $(s+1)! > 2s^s$, we find the bound on the number of cycles for a set of loops L no longer depends on the individual loops sizes, but rather their sum $m = \sum_{i=1} n_i l_i$. For $m \geqslant 1$, and writing $c(L|\sum_{i=1} n_i l_i = m)$ as $c(m)$, we have

$$c(m) > 2^{m-2-\sqrt{2m}\log_2\sqrt{2m}}, \tag{8}$$

which to first order in the exponent is 2^m. This is the first proof that the number of cycles grows as fast as possible with m. We compare our result with the best known bounds in the Discussion.

VII. DISCUSSION

This paper contains two take-home messages about the critical Kauffman model with connectivity one. The first is a new interpretation: the model is equivalent to a system of primitive cycle polynomials and their products. The second is a new result about the number of cycles: to first order in m, the number of cycles grows as 2^m, which is as fast as possible.

Let's start with the first take-home message. The mathematical structure of the $K = 1$ critical Kauffman model is entirely described by a deceptively simple algebraic system, namely, products of the primitive cycle polynomials D_l and $D_{\bar{l}}$ in Table I. These polynomials do not satisfy ordinary polynomial multiplication, but rather have a product defined by Eq. (2). Combining loops or sets of loops in a network is equivalent to multiplying out the relevant cycle polynomials.

One special property of the primitive cycle polynomials is the contagion of odd parity loops. In particular, for loops of the same size, $D_l D_{\bar{l}} = D_{\bar{l}} D_{\bar{l}}$. This is important, because it means that clusters of equal-sized loops behave in just one of two ways: as if all of the Boolean functions are copy, or as if each loop has one invert. One open question is whether the contagion of odd loops extends beyond loops of the same size. As Table I suggests, in many instances $D_{l_1} D_{\bar{l}_2}$ and $D_{\bar{l}_1} D_{\bar{l}_2}$ are identical, and it seems that factors of 2 in the loop sizes play a key role in determining this.

The two quantities of interest in a critical Kauffman model are the number of cycles c and the mean attractor length $\bar{A}$. Both can be readily obtained from the cycle polynomial. Let $E(x)$ be the product of primitive cycle polynomials: $E = D_{l_1}^{p_1} D_{\bar{l}_1}^{q_1} \dots D_{l_s}^{p_s} D_{\bar{l}_s}^{q_s}$. The number of cycles and the mean attractor length for the set of loops $\{l_1^{p_1}, \bar{l}_1^{q_1} \dots l_s^{p_s}, \bar{l}_s^{q_s}\}$ are

$$c = E|_{x=1} \quad \text{and} \quad \bar{A} = E'/E|_{x=1}, \tag{9}$$

where $E'(x) = d/dxE(x)$. Note that $E'|_{x=1} = 2^m$.

We conjecture that a finite fraction of the 2^m states of the m nodes in loops belong to cycles of the largest length. Specifically, we conjecture that the exponent times the coefficient in the last term of E divided by 2^m is at least $\prod(2^p - 2)/2^p = 0.346$, where the product is over all primes p. For example, for the network in Fig. 1, the fraction of states in cycles of length 30 is $30 \cdot 12/2^{10} = 0.352$.

Now we turn to the second take-home message. To first order in m, the number of cycles scales as 2^m. This is considerably faster than the lower bounds of $2^{0.47m}$ derived by Drossel et al. in [2] and, using a more detailed calculation, $2^{0.5m}$ derived by Flyvbjerg and Kjaer in [1]. Ours is the first proof that the number of cycles grows as fast as possible with m.

We can re-express this result in terms of the number of nodes in the network N, whereby m becomes a random variable: choose uniformly from the distribution of single input networks and see what m is. In the large N limit, the mean number of loops of length l is $\exp(-l^2/(2N))/l$. Summing over this, the mean number of nodes in loops $\overline{m}$ is asymptotically $\sqrt{\frac{\pi}{2}N}$. Since this is convex, by Jensen's inequality we can replace m with its mean, giving

$$c(N) > 2^{1.25\sqrt{N}},$$

compared to the best known bounds on the growth rate, $2^{0.63\sqrt{N}}$ [1] and $2^{0.59\sqrt{N}}$ [2].

While this paper was under review, we were able to use some of the results in it to more carefully translate $c(m)$ to $c(N)$, by averaging $c(m)$ over the distribution of m given N. Our result—that the number of cycles grows as $(2/\sqrt{e})^N$—recently appeared in Ref. [13]. This current paper can be seen as a precursor to [13], and one which opens the door to further insights by introducing a new analytic technique for studying the critical Kauffman model.

But what about the mean attractor length $\bar{A}$? Note that $\bar{A}(m)$ can always be 1, even for large m, by choosing all the loops to be of size 1. So our Jensen's inequality approach used above is no use here. Calculating $\bar{A}(N)$ requires a detailed understanding of the distribution of loop sizes given a uniform distribution over single input networks. What makes this particularly difficult is that the probabilities of finding loops of different sizes are not independent.

One topic we do not consider here is a network's resilience to a perturbation, such as a change of the state of a node from 0 to 1, or the Boolean function of a node from copy to invert [7,15]. Our technique of taking the product of cycle polynomials could help to understand how an error in one loop infects the number and length of cycles in a set of loops.

ACKNOWLEDGMENT

T.F. thanks A. Fedosyeyev and F. Sheldon for useful advice.

[1] H. Flyvbjerg and N. Kjaer, Exact solution of Kauffman's model with connectivity one, J. Phys. A: Math. Gen. **21**, 1695 (1988).

[2] B. Drossel, T. Mihaljev, and F. Greil, Number and length of attractors in a critical Kauffman model with connectivity one, Phys. Rev. Lett. **94**, 088701 (2005).

[3] S. A. Kauffman, Homeostasis and differentiation in random genetic control networks, Nature (London) **224**, 177 (1969).

[4] S. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, J. Theor. Biol. **22**, 437 (1969).

[5] M. Muñoz, Criticality and dynamical scaling in living systems, Rev. Mod. Phys. **90**, 031001 (2018).

[6] B. Daniels *et al.*, Criticality distinguishes the ensemble of biological regulatory networks, Phys. Rev. Lett. **121**, 138102 (2018).

[7] S. Bilke and F. Sjunnesson, Stability of the Kauffman model, Phys. Rev. E **65**, 016129 (2001).

[8] J. Socolar and S. Kauffman, Scaling in ordered and critical random Boolean networks, Phys. Rev. Lett. **90**, 068702 (2003).

[9] U. Bastolla and G. Parisi, The modular structure of Kauffman networks, Physica D **115**, 219 (1998).

[10] U. Bastolla and G. Parisi, Relevant elements, magnetization and dynamic properties in Kauffman networks: A numerical study, Physica D **115**, 203 (1998).

[11] B. Samuelsson and C. Troein, Superpolynomial growth in the number of attractors in Kauffman networks, Phys. Rev. Lett. **90**, 098701 (2003).

[12] V. Kaufman and B. Drossel, On the properties of cycles of simple Boolean networks, Eur. Phys. J. B **43**, 115 (2005).

[13] T. Fink and F. Sheldon, Number of attractors in the critical Kauffman model is exponential, Phys. Rev. Lett. **131**, 267402 (2023).

[14] N. J. A. Sloane (ed.), The On-Line Encyclopedia of Integer Sequences (2023), https://oeis.org.

[15] T. P. Peixoto, Redundancy and error resilience in Boolean networks, Phys. Rev. Lett. **104**, 048701 (2010).