4-1-2019

The Number of Fixed Points of AND-OR Networks with Chain Topology

Lauren Geiser
University of Dayton

The Number of Fixed Points of AND-OR Networks with Chain Topology

Honors Thesis

Lauren Geiser

Department: Mathematics

Advisor: Alan Veliz-Cuba, Ph.D.

April 2019

The Number of Fixed Points of AND-OR Networks with Chain Topology

Honors Thesis
Lauren Geiser
Department: Mathematics
Advisor: Alan Veliz-Cuba, Ph.D.
April 2019

Abstract

Boolean networks are sets of Boolean functions, which are functions that contain Boolean variables and the logical operators AND, OR, and NOT. In the simple case, the variables can be in one of two states—either 1 or 0, which can be interpreted in different ways such as ON or OFF, or TRUE or FALSE, depending on the application. Arranging model systems into Boolean functions, we can study steady states of these networks. This refers to the overall state of the dynamical system given an initial condition and another theoretical condition such as a subsequent point in time. Boolean networks have many applications, such as those in mathematics and computer science, and they can be used to study biological systems, especially to model gene networks.

The wide range of applications for Boolean networks brings us to two important questions: how do we compute steady states, and how do we find the number of fixed points? Computing the number of fixed points is very difficult. One way to simplify the computation is to focus on certain classes of networks. Another way to simplify our scope is by focusing on certain network topologies. We focus on AND-OR networks with chain topology.

AND-OR networks are Boolean networks where each coordinate function is either the AND or OR logical operator. We study the number of fixed points of these Boolean networks in the case that they have a wiring diagram with chain topology. We find closed formulas for subclasses of these networks and recursive formulas in the general case. Our results allow for an effective computation of the number of fixed points of AND-OR networks with chain topology. We further explore how our approach could be used in "fractal" chains.

Table of Contents

1. Introduction

Boolean networks, f : $\{0, 1\}^n \rightarrow \{0, 1\}^n$, have been used to study problems arising from areas such as mathematics, computer science, and biology [1, 2, 3, 4]. A particular problem of interest is counting the number of fixed points (x such that $f(x) = x$). To simplify this problem one can restrict the class of Boolean functions or the topology of the network [5, 6, 7, 8, 9, 4, 10, 11, 12, 13, 14], which in some cases allows us to find effective algorithms or formulas in closed form.

In this manuscript we focus on the number of fixed points of AND-OR networks (each Boolean function is either the AND or the OR operator) that have open or closed chain topology. The networks we study in this manuscript also arise by restricting Min-Max networks to a Boolean set of values {0, 1} [15]. Although one typically specifies the update order to analyze the dynamics, this is not necessary here as the fixed points would not change [16]. We first consider the case of finite open chain topology and find a recursive formula (Theorem 2.4) and sharp lower and upper bounds. We then consider the case of infinite and closed chain topology, and show how they can be reduced to the case of finite open chain topology (Theorems 3.1 and 3.2).

2. Open Chains

Let $f = (f_1, \ldots, f_n) : \{0,1\}^n \rightarrow \{0,1\}^n$ with $n \geq 2$ be an AND-OR network such that its wiring diagram is a chain, Fig 1. That is, we consider Boolean networks of the form:

$f_1 = x_2$, $\quad f_2 = x_1 \lozenge_2 x3$, $\quad f3 = x_2 \lozenge_3 x_4$, $\quad \ldots$, $\quad f_{n-1} = x_{n-2} \lozenge_{n-1} x_n$, $\quad f_n = x_{n-1}$,

where $\lozenge_i$ is the AND ($\wedge$) or the OR ($\vee$) operator.

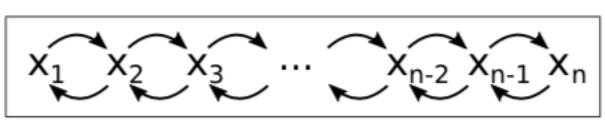

Figure 1. Wiring diagram with open chain topology.

Because this family of Boolean networks is completely determined by the sequence of logical operators $\Diamond_2, \Diamond_3, \ldots, \Diamond_{n\text{-}1}$, we can use this sequence to represent the network. Furthermore, consecutive occurrences of the same logical operator can be denoted as $\wedge^k$ or $\vee^k$.

We are interested in the number of fixed points of such Boolean networks. For simplicity we denote the elements of $\{0, 1\}^n$ as binary strings (omitting parentheses). Also, we will use the notation $\mathbf{0} = 00 \cdots 0$ and $\mathbf{1} = 11 \cdots 1$, where the length of the strings will be clear from the context. Note that $\mathbf{0}$ and $\mathbf{1}$ are fixed points of all AND-OR networks with chain topology.

Example 2.1 *Our running example will be the AND-OR network*

$$f_1 = x_2,\ f_2 = x_1 \wedge x_3,\ f_3 = x_2 \wedge x_4,\ f_4 = x_3 \vee x_5,\ f_5 = x_4 \wedge x_6,\ f_6 = x_5 \vee x_7,$$

$$f_7 = x_6 \vee x_8,\ f_8 = x_7 \vee x_9,\ f_9 = x_8 \wedge x_{10},\ f_{10} = x_9 \wedge x_{11},\ f_{11} = x_{10} \vee x_{12},\ f_{12} = x_{11}.$$

This network can be represented by the sequence of operators $\wedge \wedge \vee \wedge \vee \vee \vee \wedge \wedge \vee$. *We can further simplify this representation to* $\wedge^2 \vee \wedge \vee^3 \wedge^2 \vee$. *This AND-OR network has 13 fixed points listed in Table 1 (first column).*

Fixed points	Structure from Lemma 2.3	"Reduced" system (Proposition 2.3.1)
000000000000	000 0 0 000 00 00	00 0 0 00 00 00
000000000011	000 0 0 000 00 11	00 0 0 00 00 11
000001110000	000 0 0 111 00 00	00 0 0 11 00 00
000001111111	000 0 0 111 11 11	00 0 0 11 11 11
000001110011	000 0 0 111 00 11	00 0 0 11 00 11
000111110000	000 1 1 111 00 00	00 1 1 11 00 00
000111110011	000 1 1 111 00 11	00 1 1 11 00 11
000111111111	000 1 1 111 11 11	00 1 1 11 11 11
111100000000	111 1 0 000 00 00	11 1 0 00 00 00
111100000011	111 1 0 000 00 11	11 1 0 00 00 11
111111110000	111 1 1 111 00 00	11 1 1 11 00 00
111111110011	111 1 1 111 00 11	11 1 1 11 00 11
111111111111	111 1 1 111 11 11	11 1 1 11 11 11

Table 1. Fixed points of the AND-OR network $\wedge^2 \vee \wedge \vee^3 \wedge^2 \vee$. First column: fixed points. Second column: fixed points with the structure given by Lemma 2.3 highlighted. Third column: fixed points of reduced network, $\wedge^2 \vee \wedge \vee^2 \wedge^2 \vee$, with the structure given by Lemma 2.3 highlighted. For this example, the fixed points can be found using software [20]. We performed computations using resources from the Ohio Supercomputer Center [21].

The next lemma states that the number of fixed points depends only on the powers of the operators. Since we do not know which operator is last (∧ or ∨), we will simply use ellipses without explicitly writing the last operator.

Lemma 2.2 *The AND-OR networks* $f = \wedge^{k1} \vee^{k2} \wedge^{k3} \cdots$ *and* $g = \vee^{k1} \wedge^{k2} \vee^{k3} \cdots$ *have the same number of fixed points.*

Proof. Consider $\phi : \{0, 1\}^n \rightarrow \{0, 1\}^n$ given by $\phi(x_1, \ldots, x_n) = (\neg x1, \ldots, \neg xn)$, where ¬ is the logical operator NOT. Using the fact that $\neg(p \wedge q) = \neg p \vee \neg q$ and $\neg(p \vee q) = \neg p \wedge \neg q$, it follows that $f(\phi(x)) = \phi(g(x))$. Then, x will be a fixed point of g if and only if $\phi(x)$ is a fixed point of f. So, ϕ is a bijection between the fixed points of g and f.

Because we are interested in the number of fixed points, we will simply use $(k_1, k_2, \ldots, k_m)$ to refer to a network. For instance, the AND-OR network seen in Example 2.1 can be represented simply by (2, 1, 1, 3, 2, 1). We denote the number of fixed points by $F(k_1, k_2, \ldots, k_m)$. A similar approach was used by [17] to study non-monotonic Boolean networks.

The following lemma states that consecutive variables that have the same logical operator must be equal.

Lemma 2.3 *Consider an AND-OR network f represented by* $(k_1, k_2, \ldots, k_m)$*. Denote an element of the domain of f by* $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m)$*, where* $\mathbf{x}^1 \in \{0,1\}^{k_1+1}$*,* $\mathbf{x}^m \in \{0,1\}^{km+1}$*, and* $\mathbf{x}^i \in \{0,1\}^{k_i}$ *for* $i = 2, \ldots, m-1$*. If* $\mathbf{x}$ *is a fixed point of f, then* $\mathbf{x}^i = \mathbf{0}$ *or* $\mathbf{x}^i = \mathbf{1}$ *for* $i = 1, \ldots, m$*.*

Proof. Let $\mathbf{x}$ be a fixed point of f. We use $(\mathbf{x}^i)_j$ to denote the j-th coordinate of $\mathbf{x}^i$. Note that $(\mathbf{x}^1)_1 = (\mathbf{x}^1)_2$ and $(\mathbf{x}^m)_{k_m} = (\mathbf{x}^m)_{k_m+1}$ by definition of f (the first and last coordinate functions of f depend on single variables).

Now, the rest of the proof follows from the fact that if $q = p \wedge r$ and $r = q \wedge s$ or if $q = p \vee r$ and $r = q \vee s$, then $q = r$. This implies that consecutive variables, $(\mathbf{x}^i)_j$ and

$(\mathbf{x}^i)_{j+1}$, that have the same logical operators must be the same.

□

The next proposition states that the numbers k_i in $F(k_1, ..., k_m)$ can be assumed to be at most 2 for $2 \le i \le m$-1, and 1 for k_1 and k_m. For example, this will imply that $F(2, 1, 1, 3, 2, 1) = F(1, 1, 1, 2, 2, 1)$ and $F(2, 5, 3, 1, 4, 3) = F(1, 2, 2, 1, 2, 1)$.

Example 2.1 (cont.) We highlight the structure of the fixed points of $\wedge^2\vee\wedge\vee^3\wedge^2\vee$ in Table 1 (second column).

Proposition 2.3.1 *$F(k_1, k_2, ..., k_{m-1}, k_m) = F(1, \min\{k_2,2\}, ..., \min\{k_{m-1},2\}, 1)$ for all positive integers k_i.*

Proof. We will use the notation of Lemma 2.3.

We first show that $f = \wedge^{k_1}\vee^{k_2}\wedge^{k_3}\cdots$ and $g = \wedge\vee^{k_2}\wedge^{k_3}\cdots$ have the same number of fixed points. Let $\mathbf{x} = (\mathbf{x}^1, ..., \mathbf{x}^m)$ be a fixed point of f. Then, by Lemma 2.3 we have $\mathbf{x}^1 = \mathbf{0}$ or $\mathbf{x}^1 = \mathbf{1}$. Consider $\mathbf{y} = (\mathbf{z}, \mathbf{x}^2, ..., \mathbf{x}^m)$, where $\mathbf{z} = ((\mathbf{x}^1)_1, (\mathbf{x}^1)_2)$. It can be checked that $\mathbf{y}$ is a fixed point of g. Now, if $\mathbf{y} = (\mathbf{z}, \mathbf{x}^2, ..., \mathbf{x}^m)$ is a fixed point of g, Lemma 2.3 implies that $\mathbf{z} = \mathbf{0}$ or $\mathbf{z} = \mathbf{1}$. We define $\mathbf{x} = (\mathbf{x}^1, ..., \mathbf{x}^m)$ in the domain of f, where $\mathbf{x}^1 = \mathbf{0}$ if $\mathbf{z} = \mathbf{0}$ and $\mathbf{x}^1 = \mathbf{1}$ if $\mathbf{z} = \mathbf{1}$. Then, it can be checked that $\mathbf{x}$ is a fixed point of f. This shows that $F(k_1, k_2, ..., k_{m-1}, k_m) = F(1, k_2, ..., k_{m-1}, k_m)$, and similarly it can be shown that $F(1, k_2, ..., k_{m-1}, k_m) = \mathrm{F}(1, k_2, ..., k_{m-1}, 1)$.

We now show that for $k_2 \ge 2$, $f = \wedge^{k_1}\vee^{k_2}\wedge^{k_3}\cdots$ and $g = \wedge^{k_1}\vee^2\wedge^{k_3}\cdots$ have the same number of fixed points. The general case is analogous. Let $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^m)$ be a fixed point of f. Then, by Lemma 2.3 we have $\mathbf{x}^2 = \mathbf{0}$ or $\mathbf{x}^2 = \mathbf{1}$. Consider $\mathbf{y} = (\mathbf{x}^1, \mathbf{z}, \mathbf{x}^3, ..., \mathbf{x}^m)$, where $\mathbf{z} = ((\mathbf{x}^2)_1, (\mathbf{x}^2)_2)$. It can be checked that $\mathbf{y}$ is a fixed point of g. Now, if $\mathbf{y} = (\mathbf{x}^1, \mathbf{z}, \mathbf{x}^3, ..., \mathbf{x}^m)$ is a fixed point of g, Lemma 2.3 implies that $\mathbf{z} = \mathbf{0}$ or $\mathbf{z} = \mathbf{1}$. We define

$\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^m)$ in the domain of f, where $\mathbf{x}^1 = \mathbf{0}$ if $\mathbf{z} = \mathbf{0}$ and $\mathbf{x}^1 = \mathbf{1}$ if $\mathbf{z} = \mathbf{1}$.Then, it can be checked that $\mathbf{x}$ is a fixed point of f. This shows that $F(k_1, k_2, ..., k_{m-1}, k_m) = F(k_1, 2, k_3, ..., k_{m-1}, k_m)$ for $k_2 \geq 2$.

Example 2.1 (cont.) Proposition 2.3.1 guarantees that $\wedge^2\vee\wedge\vee^3\wedge^2\vee$ and $\wedge\vee\wedge\vee^2\wedge^2\vee$ have the same number of fixed points. We can consider the second AND-OR network as a "reduced" version of the original AND-OR network [18, 19]. This is illustrated in Table 1 (third column).

Proposition 2.3.2 *Let $r_1, \ldots, r_m$ in {1, 2}, and $m \geq 2$. Then, we have the following*

$$\mathcal{F}(1, r_1, \ldots, r_m, 1) = \begin{cases} \mathcal{F}(1, r_3, \ldots, r_m, 1) + \mathcal{F}(r_3, \ldots, r_m, 1), & \textit{for } r_1 = 1, r_2 = 1 \\ \mathcal{F}(2, r_3, \ldots, r_m, 1) + \mathcal{F}(1, r_3, \ldots, r_m, 1), & \textit{for } r_1 = 1, r_2 = 2 \\ \mathcal{F}(1, 1, r_3, \ldots, r_m, 1) + \mathcal{F}(r_3, \ldots, r_m, 1), & \textit{for } r_1 = 2, r_2 = 1 \\ \mathcal{F}(1, 2, r_3, \ldots, r_m, 1) + \mathcal{F}(1, r_3, \ldots, r_m, 1), & \textit{for } r_1 = 2, r_2 = 2 \end{cases} \quad (1)$$

Proof. We will use the notation of Lemma 2.3. If $r_1 = 1$, $r_2 = 1$, then we claim that any fixed point of $f = \wedge\vee\wedge\vee^{r_3}\wedge^{r_4}\cdots$ is of the form $\mathbf{x} = (\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^m, \mathbf{x}^{m+1})$ where either $\mathbf{x}^0 = \mathbf{0}$ and $\mathbf{z} = (\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^m, \mathbf{x}^{m+1})$ is a fixed point of $g = \wedge\vee^{r_3}\wedge^{r_4}\cdots$ or $\mathbf{x}^0 = \mathbf{x}^1 = \mathbf{1}$ and $\mathbf{z} = (\mathbf{x}^2, ..., \mathbf{x}^m, \mathbf{x}^{m+1})$ is a fixed point of $h = \vee^{r_3}\wedge^{r_4}\cdots$. Indeed, the system of Boolean equations for fixed points is

$$\begin{aligned} x_1 &= x_2 \\ x_2 &= x_1 \wedge x_3 \\ x_3 &= x_2 \vee x_4 \\ x_4 &= x_3 \wedge x_5 \\ x_5 &= x_4 \vee x_6 \\ &\vdots \\ x_n &= x_{n-1} \end{aligned}$$

We divide this system of equations in the cases $x_1 = 0$ and $x_1 = 1$. Then, using the fact that $1 = m \wedge n$ implies that $m = n = 1$, that $0 = m \vee n$ implies $m = n = 0$, it follows that we obtain the two systems

$$\begin{aligned} x_3 &= x_4 \\ x_4 &= x_3 \wedge x_5 \\ x_5 &= x_4 \vee x_6 \\ &\vdots \\ x_n &= x_{n-1} \end{aligned}$$

and

$$\begin{aligned} x_4 &= x_5 \\ x_5 &= x_4 \vee x_6 \\ &\vdots \\ x_n &= x_{n-1}, \end{aligned}$$

corresponding to the cases $x_1 = 0$ and $x_1 = 1$, respectively (see Fig. 2). This means that the number of fixed points of f is equal to the number of solutions of these two systems. Since the solutions of the first system are the fixed points of $g = \wedge\vee^{r_3}\wedge^{r_4}\cdots$ and the solutions of the second system are the fixed points of $h = \vee^{r_3}\wedge^{r_4}\cdots$, we obtain $F(1, 1, 1, r_3 \ldots, r_m, 1) = F(1, r_3, \ldots, r_m, 1) + F(r_3, \ldots, r_m, 1)$.

The proof for the other three cases is similar.

□

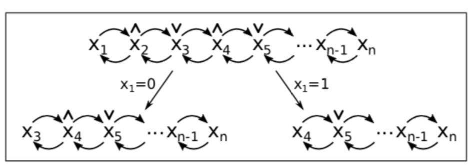

Figure 2. Idea behind the proof of Proposition 2.3.2 (logical operators are included for clarity). Considering the cases $x_1 = 0$ and $x_1 = 1$ yields systems of equations that correspond to smaller AND-OR networks.

By convention, we denote the AND-OR network $f(x_1, x_2) = (x_2, x_1)$ by an empty sequence, ( ). We also use the convention $F(0, k_1, ..., k_m, 0) = F(k_1, ..., k_m, 0) = F(0, k_1, ..., k_m) = F(k_1, ..., k_m)$ which will simplify the formulation of upcoming results.

Theorem 2.4 *With the convention above, we have that for $m \geq 3$ and $k_i \geq 1$*

$$F(k_1, ..., k_m) = F(k_2\text{-}1, k_3, ..., k_m) + F(k_3\text{-}1, k_4, ..., k_m)$$

and

$$F(k_1, ..., k_m) = F(k_1, ..., k_{m\text{-}2}, k_{m\text{-}1}\text{-}1) + F(k_1, ..., k_{m\text{-}3}, k_{m\text{-}2}\text{-}1).$$

Also,

$$F(k_1,k_2) = 3,\ F(k) = 2 \text{ for } k \geq 0.$$

Proof. For $m \geq 4$ the result follows directly from Propositions 2.3.1 and 2.3.2. For $m = 3$ the results follows from $F(1, 2, 1) = 5$, $F(1, 1, 1) = 4$, $F(1, 1) = 3$, $F(1) = 2$, and $F(0) = 2$ which can be easily checked by complete enumeration.

□

Example 2.1 (cont.) We now use Theorem 2.4 to find the number of fixed points of $\wedge^2\vee\wedge\vee^3\wedge^2\vee$:

$F(2, 1, 1, 3, 2, 1) = F(1, 1, 1, 2, 2, 1)$

$= F(1\text{-}1, 1, 2, 2, 1) + F(1\text{-}1, 2, 2, 1)$

$= F(1, 2, 2, 1) + F(2, 2, 1)$

$= F(2\text{-}1, 2, 1) + F(2\text{-}1, 1) + F(2\text{-}1, 1) + F(1\text{-}1)$

$= F(1, 2, 1) + F(1, 1) + F(1, 1) + F(0)$

$= F(2\text{-}1,1) + F(1\text{-}1) + F(1, 1) + F(1, 1) + F(0)$

$= F(1, 1) + F(0) + F(1, 1) + F(1, 1) + F(0)$

$= 3+2+3+3+2$

$= 13$

or

$F(2, 1, 1, 3, 2, 1) = F(1, 1, 1, 2, 2, 1)$

$= F(1, 1, 1, 2, 2\text{-}1) + F(1, 1, 1, 2\text{-}1)$

$= F(1, 1, 1, 2, 1) + F(1, 1, 1, 1)$

$= F(1, 1, 1, 2\text{-}1) + F(1, 1, 1\text{-}1) + F(1, 1, 1\text{-}1) + F(1, 1\text{-}1)$

$= F(1, 1, 1, 1) + F(1, 1) + F(1, 1) + F(1)$

$= F(1, 1, 1\text{-}1) + F(1, 1\text{-}1) + F(1, 1) + F(1, 1) + F(1)$

$= F(1, 1) + F(1) + F(1, 1) + F(1, 1) + F(1)$

$= 3+2+3+3+2$

$= 13$

In this way, Theorem 2.4 provides a recursive formula to compute the number of fixed points of AND-OR networks with chain topology without the need of exhaustive enumeration. We now study the 2 especial cases of $F(1, 1, \ldots , 1, 1)$ and F(2, 2, ..., 2, 2).

Define $A_n = (1, \underbrace{1, 1, ..., 1, 1}_{n \text{ times}}, 1)$ and $B_n = (2, \underbrace{2, 2, ..., 2, 2}_{n \text{ times}}, 2)$. Also define the sequences $a_0 = 1$, $a_1 = 1$, $a_2 = 1$, and $a_n = a_{n-2} + a_{n-3}$ for $n \geq 3$ and $b_0 = 1$, $b_1 = 1$, and $b_n = b_{n-1} + b_{n-2}$ for $n \geq 2$. Note that (a_n) is the Padovan sequence and (b_n) is the Fibonacci sequence.

Corollary 2.4.1 *With the definitions above we have $F(A_n) = a_{n+5}$ and $F(B_n) = b_{n+3}$ for $n \geq 0$, and the sharp bounds $F(A_n) \leq F(1, r_1, r_2, ..., r_n, 1) \leq F(B_n)$ for all $r_i \geq 1$.*

Proof. It follows from Theorem 2.4 or Proposition 2.3.2 using induction.

$\square$

3. Infinite and Closed Chains

In this section we study the cases of AND-OR networks with infinitely many variables and when the topology is a closed chain.

When the AND-OR network has infinitely many variables we have a infinite collection of Boolean functions $f = (..., f_2, f_1, f_0, f_1, f_2,...)$ such that $f_i = x_{i-1} \wedge x_{i+1}$ or $f_i = x_{i-1} \vee x_{i+1}$. We can use the same notation of Section 2 and denote consecutive logical operators as $\wedge^k$ or $\vee^k$, where k could also be ∞. Also, we can simply use the exponents to represent the AND-OR network. For example, $(\infty, 1, 2, \infty)$ and $\wedge^\infty\vee\wedge^2\vee^\infty$ represent the AND-OR network . . . ∧∧∧∨∧∧∨∨∨… . Similarly, (. . . , 1, 1, 2, 1, 1, 2, 1, 1, 2, . . .) and $\ldots\wedge\vee\wedge^2\vee\wedge\vee^2\wedge\vee\wedge^2 \ldots$ represent the AND-OR network . . . ∧∨∧∧∨∧∨∨∧∨∧∧… .

The following theorem allows us to use the results from Section 2 to study AND-OR networks with infinitely many variables.

Theorem 3.1 *With the notation above and* $k_i \geq 1$ *we have the following.*

$$F(\infty) = 2$$

$$F(\infty, k_1, k_2, ..., k_{m-1}, k_m, \infty) = F(1, k_1, k_2, ..., k_{m-1}, k_m, 1)$$

$$F(\infty, k_1, k_2, k_3, ...) = \infty$$

$$F(..., k_3, k_2, k_1, \infty) = \infty$$

$$F(..., k_{-3}, k_{-2}, k_{-1}, k_0, k_1, k_2, k_3, ...) = \infty$$

Proof. To prove the first equality we consider the AND-OR network where all logical operators are ∧. If one of the variables is 0, it follows that all the other variables are also 0. Similarly, if one of the variables is 1, all the other variables are also 1. Thus, the only fixed points of this AND-OR network are **0** and **1**.

The second equality follows the same approach seen in Proposition 2.3.1.

To prove the third equality we first observe that $F(\infty, k_1, k_2, k_3, \ldots) = F(1, k_1, k_2, k_3, \ldots)$. Now, we will show that any fixed point of the AND-OR network $F(1, k_1, k_2, k_3, \ldots, k_r)$ defines a fixed point of $F(1, k_1, k_2, k_3, \ldots)$. Indeed, using the notation of Lemma 2.3, a fixed point of the AND-OR network $F(1, k_1, \ldots, k_r)$ has the form $\mathbf{x} = (\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^r)$. Then, denoting $\mathbf{z} = (1, 1, \ldots)$ if $\mathbf{x}^r = \mathbf{1}$ and $\mathbf{z} = (0, 0, \ldots)$ if $\mathbf{x}^r = \mathbf{0}$, it follows that $(\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^r, \mathbf{z})$ is a fixed point of $F(1, k_1, k_2, k_3, \ldots)$. Since r is arbitrary, $F(1, k_1, \ldots, k_r)$ is not bounded (see Corollary 2.4.1) and then number of fixed points of $F(1, k_1, \ldots)$ is ∞. The last two equalities are similar.

□

When the topology of the network is a closed chain, we have the network

$$f_1 = x_n \Diamond_1 x_2, \quad f_2 = x_1 \Diamond_2 x_3, \quad f_3 = x_2 \Diamond_3 x_4, \ \ldots\ , f_{n-1} = x_{n-2} \Diamond_{n-1} x_n, \quad f_n = x_{n-1} \Diamond_n x_1.$$

We denote this network as $[k_1, k_2, \ldots, k_r]$ or any cyclic permutation that groups consecutive logical operators. Thus, the AND-OR network

$$f_1 = x_n \wedge x_2, \quad f_2 = x_1 \vee x_3, \quad f_3 = x_2 \wedge x_4, \quad f_4 = x_3 \vee x_5, \quad f_5 = x_4 \vee x_6, \quad f_6 = x_5 \wedge x_1,$$

will not be denoted by [1, 1, 1, 2, 1] ("splitting" the first and last ∧'s), but by [1, 1, 2, 2], [1, 2, 2, 1], [2, 2, 1, 1], or [2, 1, 1, 2] (combining the first and last ∧'s). This means that r in $[k_1, k_2, \ldots, k_r]$ will always be an even number or equal to 1. The number of fixed points will be denoted by $F[k_1, k_2, \ldots, k_r]$. The following propositions and theorem allow us to use the results from Section 2 to study AND-OR networks with closed chain topology.

<u>**Proposition 3.1.1**</u> *With the notation above, we have that for $k_i \geq 1$*

$$F[k_1, k_2, \ldots, k_r] = F[\min\{2, k_1\}, \min\{2, k_2\}, \ldots, \min\{2, k_r\}].$$

It is analogous to the proof of Proposition 2.3.1.

<u>**Proposition 3.1.2**</u> *Consider $k_i \geq 1$, $m \geq 6$, and $l \geq 8$. Then,*

$$F[2, k_2, ..., k_m] = F(k_2\text{-}1, k_3, ..., k_{m\text{-}1}, k_m\text{-}1) + F(k_3\text{-}1, k_4, ..., k_{m\text{-}2}, k_{m\text{-}1}\text{-}1),$$

$$F[1, k_2, ..., k_l] = F(k_3\text{-}1, k_4, ..., k_{l\text{-}1}\text{-}1) + F(k_4\text{-}1, k_5, ..., k_{l\text{-}1}, k_l\text{-}1) + F(k_2\text{-}1, k_3, ..., k_{l\text{-}3}, k_{l\text{-}2}\text{-}1) + F(k_4\text{-}1, k_5, ..., k_{l\text{-}3}, k_{l\text{-}2}\text{-}1).$$

Proof. The first equality is analogous to Proposition 2.3.2. To prove the second equality we use the notation of Lemma 2.3.

We have several cases to consider for $k_{l\text{-}2}$, $k_{l\text{-}1}$, k_l, k_2, k_3, and k_4. We focus on the case $k_{l\text{-}2} = k_{l\text{-}1} = k_l = k_2 = k_3 = k_4 = 1$ since the other cases are analogous. Note that we want to prove

$$F[1, 1, 1, 1, k_5, ..., k_{l\text{-}3}, 1, 1, 1] = F(1, k_5, ..., k_{l\text{-}3}, 1) + F(k_5, ..., k_{l\text{-}3}, 1, 1) + F(1, 1, k_5, ..., k_{l\text{-}3}) - F(k_5, ..., k_{l\text{-}3}).$$

The fixed points of the AND-OR network are the solutions of

$$\begin{aligned}
x_1 &= x_n \wedge x_2,\\
x_2 &= x_1 \vee x_3,\\
x_3 &= x_2 \wedge x_4,\\
x_4 &= x_3 \vee x_5,\\
x_5 &= x_4 \wedge x_6,\\
&\vdots\\
x_{n-3} &= x_{n-4} \wedge x_{n-2},\\
x_{n-2} &= x_{n-3} \vee x_{n-1},\\
x_{n-1} &= x_{n-2} \wedge x_n,\\
x_n &= x_{n-1} \vee x_1.
\end{aligned}$$

We now consider the cases $x_1 = 1$ and $x_1 = 0$ (see Fig. 3). The case $x_1 = 1$ yields the system of equations

$$\begin{aligned}
x_3 &= x_4,\\
x_4 &= x_3 \vee x_5,\\
x_5 &= x_4 \wedge x_6,\\
&\vdots\\
x_{n-3} &= x_{n-4} \wedge x_{n-2},\\
x_{n-2} &= x_{n-3} \vee x_{n-1},\\
x_{n-1} &= x_{n-2},
\end{aligned}$$

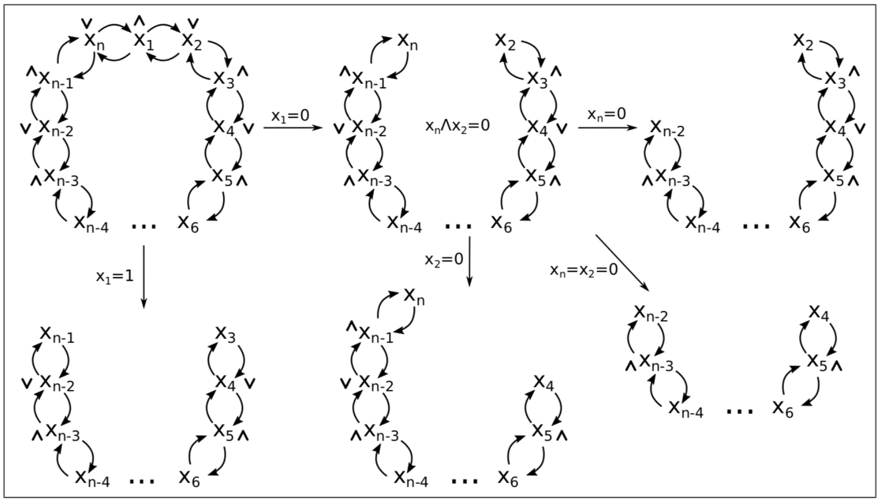

Figure 3. Idea behind the proof of Proposition 3.1.2 (logical operators are included for clarity). Considering the case $x_1 = 1$ yields a system of equations that corresponds to a smaller AND-OR network. Considering the case $x_1 = 0$ yields a system of equation that does not correspond to an AND-OR network (due to the equation $x_n \wedge x_2 = 0$). However, the subcases $x_n = 0$ and $x_2 = 0$ yield systems of equations that do correspond to smaller AND-OR networks. These two systems have overlapping solutions, so we must also take into consideration the common case $x_n = x_2 = 0$ when counting the number of fixed points.

which has $F(1, k_5, \ldots, k_{l\text{-}3}, 1)$ solutions. On the other hand, when we consider $x_1 = 0$ the first equation becomes $x_n \wedge x_2 = 0$. We now have 2 subcases: $x_n = 0$ and $x_2 = 0$. The subcase $x_n = 0$ yields

$$\begin{aligned}
x_2 &= x_3, \\
x_3 &= x_2 \wedge x_4, \\
x_4 &= x_3 \vee x_5, \\
x_5 &= x_4 \wedge x_6, \\
&\vdots \\
x_{n-3} &= x_{n-4} \wedge x_{n-2}, \\
x_{n-2} &= x_{n-3},
\end{aligned}$$

which has $F(1, 1, k_5, \ldots, k_{l\text{-}3})$ solutions. The subcase $x_2 = 0$ yields

$$\begin{aligned}
x_4 &= x_5, \\
x_5 &= x_4 \wedge x_6, \\
&\vdots \\
x_{n-3} &= x_{n-4} \wedge x_{n-2}, \\
x_{n-2} &= x_{n-3} \vee x_{n-1}, \\
x_{n-1} &= x_{n-2} \wedge x_n, \\
x_n &= x_{n-1},
\end{aligned}$$

which has $F(k_5, ..., k_{l-3}, 1, 1)$ solutions. Thus, adding up these 3 numbers we obtain $F(1, k_5, ..., k_{l-3}, 1) + F(k_5, \ldots, k_{l-3}, 1, 1) + F(1, 1, k_5, ..., k_{l-3})$. However, this is not $F[1,1, 1, 1, k_5, ..., k_{l-3}, 1, 1, 1]$, since the subcases $x_n = 0$ and $x_2 = 0$ overlap. We need to subtract the number of solutions of the system

$$
\begin{aligned}
x_4 &= x_5, \\
x_5 &= x_4 \wedge x_6, \\
&\vdots \\
x_{n-3} &= x_{n-4} \wedge x_{n-2}, \\
x_{n-2} &= x_{n-3},
\end{aligned}
$$

which has $F(k_5, \ldots, k_{l-3})$ solutions. Then, the result follows.

□

We now declare some conventions to write Proposition 3.1.2 more compactly. We define $F(-1) = 1$, $(k_s\text{-}1, ..., k_{s-1}\text{-}1) = (k_s\text{-}2)$, and $(k_s\text{-}1, ..., k_t\text{-}1) = (\text{-}1)$ for $s > t$.

<u>Theorem 3.2</u> *With the conventions above, we have that for $m \geq 4$ and $k_i \geq 1$*

$F[2, k_2, ..., k_r] = F(k_2\text{-}1, k_3, ..., k_{r\text{-}1}, k_r\text{-}1) + F(k_3\text{-}1, k_4, ..., k_{r\text{-}2}, k_{r\text{-}1}\text{-}1)$,

$F[1, k_2, ..., k_r] = F(k_3\text{-}1, k_4, ..., k_{r\text{-}1}\text{-}1) + F(k_4\text{-}1, k_5, ..., k_{r\text{-}1}, k_r\text{-}1) + F(k_2\text{-}1, k_3, ..., k_{r\text{-}3}, k_{r\text{-}2}\text{-}1)$
$\quad - F(k_4\text{-}1, k_5, ..., k_{r\text{-}3}, k_{r\text{-}2}\text{-}1)$.

Also,

$F[k] = 2$ *for* $k \geq 3$,

$F[k, 1] = 2$ *for* $k \geq 2$,

$F[k_1, k_2] = 3$ *for* $k_1, k_2 \geq 2$,

Proof. The first two equalities follow directly from Proposition 3.1.1 and 3.1.2 using the convention declared above. The last 3 equalities follow from Proposition 3.1.1 and $F[3] = F[2, 1] = 2$ and $F[2, 2] = 3$, which can be verified by complete enumeration.

□

As in Section 2, we now consider the cases $A_n = (1, \underbrace{1, 1, ..., 1, 1,}_{n \text{ times}} 1)$ and $B_n = (2, \underbrace{2, 2, ..., 2, 2,}_{n \text{ times}} 2)$. We denote the number of fixed points of the corresponding AND-OR networks with closed chain topology as $F[A_n]$ and $F[B_n]$, respectively.

Corollary 3.2.1 *With the notation above we have* $F[A_n] = 3a_n - a_{n-2}$ *and* $F[B_n] = b_{n+2} + b_n$ *for* $n \geq 2$*, and the sharp bounds* $F[A_n] \leq F[k_0, k_1, ..., k_n, k_{n+1}] \leq F[B_n]$ *for all* $r_i \geq 1$.

Proof. The proof follows from first using Theorem 3.2 and then Corollary 2.4.1.

□

Example 3.3 *We consider*

$f_1 = x_{12} \wedge x_2,\quad f_2 = x_1 \wedge x_3,\quad f_3 = x_2 \wedge x_4,\quad f4 = x_3 \vee x_5,\quad f_5 = x_4 \wedge x_6,\quad f_6 = x_5 \vee x_7,$

$f_7 = x_6 \vee x_8,\quad f_8 = x_7 \vee x_9,\quad f_9 = x_8 \wedge x_{10},\quad f_{10} = x_9 \wedge x_{11},\quad f_{11} = x_{10} \vee x_{12},\quad f_{12} = x_{11} \vee x_1.$

We will use Theorems 2.4 and 3.2 for the representations [3, 1, 1, 3, 2, 2] *and* [1, 3, 2, 2, 3,1] *of f.*

$$F[3, 1, 1, 3, 2, 2] = F[2, 1, 1, 2, 2, 2]$$

$$= F(1\text{-}1, 1, 2, 2, 2\text{-}1) + F(1\text{-}1, 2, 2\text{-}1)$$

$$= F(1, 2, 2, 1) + F(2, 1)$$

$$= F(2\text{-}1, 2, 1) + F(2\text{-}1, 1) + F(2, 1)$$

$$= F(1, 2, 1) + F(1, 1) + F(2, 1)$$

$$= F(2\text{-}1, 1) + F(1\text{-}1) + F(1, 1) + F(2, 1)$$

$$= F(1, 1) + F(0) + F(1, 1) + F(2, 1)$$

$= 3+2+3+3$

$=11$

$F[1, 3, 2, 2, 3, 1] = F[1, 2, 2, 2, 2, 1]$

$= F(2\text{-}1, 2, 2\text{-}1) + F(2\text{-}1, 2, 1\text{-}1) + F(2\text{-}1, 2, 2\text{-}1) - F(2\text{-}2)$

$= F(1, 2, 1) + F(1, 2) + F(1, 2, 1) - F(0)$

$= F(1, 1) + F(0) + F(1, 2) + F(1, 1) + F(0) - F(0)$

$= 3+2+3+3+2-2$

$=11$

4. Final Remarks: Coupled Chains

Although the results in this manuscript are for chain topology, we now show how our techniques could also be used for coupled chains. These couplings could be considered as "fractal" versions of the 1-dimensional chains that we covered in previous sections. However, due to the complex couplings that could be attain and the different cases that appear (e.g. the proof of Proposition 3.1.1 has 2^6 subcases per case), a single proposition that covers all cases would be unfeasible. Thus, we will consider two examples featuring different couplings of chains: a coupling of three open chains, and a coupling of an open and a closed chain.

First, we will prove two lemmas that will allow us to handle intersection of chains. To make the notation simpler, a pair of edges between two vertices will be simply denoted by a single undirected edge (Fig. 5). When it is not required to label vertices, we will use an even simpler representation of the wiring diagram (top insets in Fig. 5). Also, we will use $F[f]$ to denote the number of fixed points of an AND-OR network f.

Consider an AND-OR network, f: $\{0, 1\}^n \rightarrow \{0, 1\}^n$, and $S \subseteq \{1, 2, ..., n\}$. We define a new network g : $\{0,1\}^{n-|S|} \rightarrow \{0,1\}^{n-|S|}$ in the variables $\{x_i : i \notin S\}$, denoted by $g = f \setminus S$, as follows.

1. Remove the vertices in $\{x_i : i \in S\}$ from the wiring diagram of f. Note that this also means that we remove the edges of the form $x_i \rightarrow x_j$ and $x_k \rightarrow x_i$ where $i \in S$.
2. For each variable x_k in the new wiring diagram, g_k will be the same logical operator as in the wiring diagram of f, but may possibly depend on fewer variables. Note that the operators ∨ and ∧ on a single variable are simply the identity function.

Example 4.1 *Consider the AND-OR network with wiring diagram given by Fig. 4a and let* $S = \{7, 8\}$. *The new network* $g = f \setminus S$ *has wiring diagram shown in Fig. 5a. Note that g depends on 20 variables, and variables* x_6 , x_9 , *and* x_{10} *depend on a single variable only (e.g. the Boolean function corresponding to* x_6 *is* $g_6 = x_5$*).*

We now state and prove the lemmas.

Lemma 4.2 *Consider an AND-OR network f consisting of coupled chains such that* x_n *depends on 2 or more variables as shown in Fig. 6a. Denote with* $R = \{1, 2, \ldots, r\}$. *Then, the number of steady states of f is equal to*

$$\mathcal{F}[f] = \mathcal{F}[f \setminus (\{n\} \cup \{i_s : s \in R\})] + \sum_{\emptyset \neq S \subseteq R} (-1)^{|S|+1} \mathcal{F}[f \setminus (\{n\} \cup \{i_s : s \in S\} \cup \{j_s : s \in S\})].$$

Proof. We proceed by cases as in the proof of Proposition 3.1.2.

If $x_n = 1$, then any fixed point of f will satisfy $x_{i_s} = x_{j_s} \vee 1 = 1$ and $x_{j_s} = x_{k_s} \wedge x_{i_S} = x_{k_s} \wedge 1 = x_{k_s}$. Thus, x is a fixed point of f if and only if $y = (x_s)_{s \in \{1, ..., n\} \setminus \{n, i_1, ..., i_r\}}$ is a fixed point of the AND-OR network with wiring diagram given in Fig. 6b. This smaller network is precisely $f \setminus (\{n\} \cup \{i_s : s \in R\})$

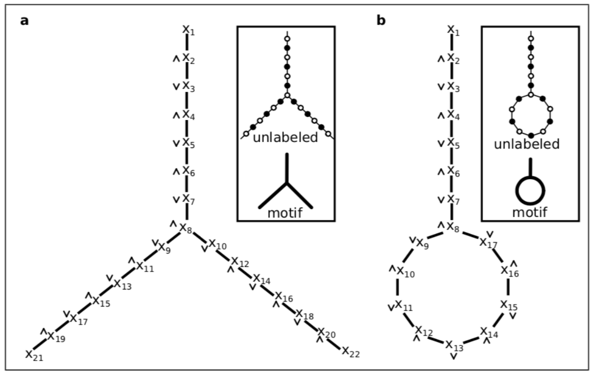

Figure 4. Coupling chains. (**a**) Wiring diagram of an AND-OR network consisting of the coupling of 3 open chains. Each undirected edge represents two edges as shown in Fig. 1. For example, $f_1 = x_2$, $f_2 = x_1 \wedge x_3$, and $f_8 = x_7 \wedge x_9 \wedge x_{10}$. Vertex x_1 could also be assigned the AND or OR operator. (**b**) Wiring diagram of an AND-OR network consisting of the coupling of a closed and open chain. In both panels, the insets show the simplified representation of the wiring diagram where the labels of variables are omitted. Open circles indicate the AND operator, whereas filled circles indicate the OR operator. The vertex corresponding to x_1 is left blank, but could also be assigned a filled or open circle. The insets also show an undirected graph highlighting the coupling motif of the AND-OR network. In previous sections the coupling motif would simply be a (finite or infinite) line or a circle.

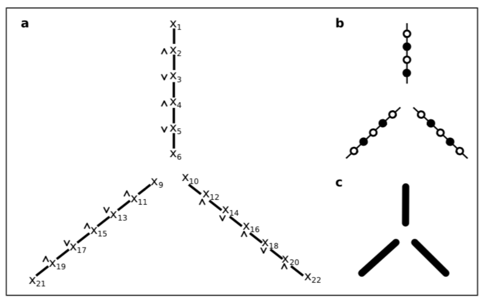

Figure 5. Example of $f \backslash S$. (**a**) Wiring diagram of the AND-OR network $f \backslash S$, where f is given in Fig. 4a and $S = \{7, 8\}$. (**b**) Simplified representation of the wiring diagram with labels omitted. (**c**) Undirected graph highlighting the coupling motif of $f \backslash S$. Note that this graph now has three connected components all of them being open chains.

If $x_n = 0$, then any fixed point of f will satisfy $x_{i_1} \wedge ... \wedge x_{i_r} = 0$ (Fig. 6c). This does not correspond to a system of equations of an AND-OR network, so we consider the subcases $x_{i_s} = 0$ for each $s \in R = \{1, 2, ..., r\}$.

If $x_{i_s} = 0$, then equation $x_{i_1} \wedge ... \wedge x_{i_r} = 0$ is satisfied. Also, $x_{j_s} = x_{k_s} \wedge 0 = 0$ and then x_{k_s} will depend on a single variable only (see Fig. 6d for the cases $s = 1$ and $s = r$). The resulting AND-OR network is $f \backslash (\{n, i_s, j_s\})$. Note that these subcases overlap, so we use the inclusion-exclusion principle to properly account for this. For the case $x_n = 0$, the inclusion-exclusion principle implies that the number of fixed points is

$$\left|\{\text{fixed points of the form } x_n = 0\}\right| = \sum_{\emptyset \neq S \subseteq R} (-1)^{|S|+1} \left|\{\text{fixed points of the form } x_{i_s} = 0 \text{ for } s \in S\}\right|.$$

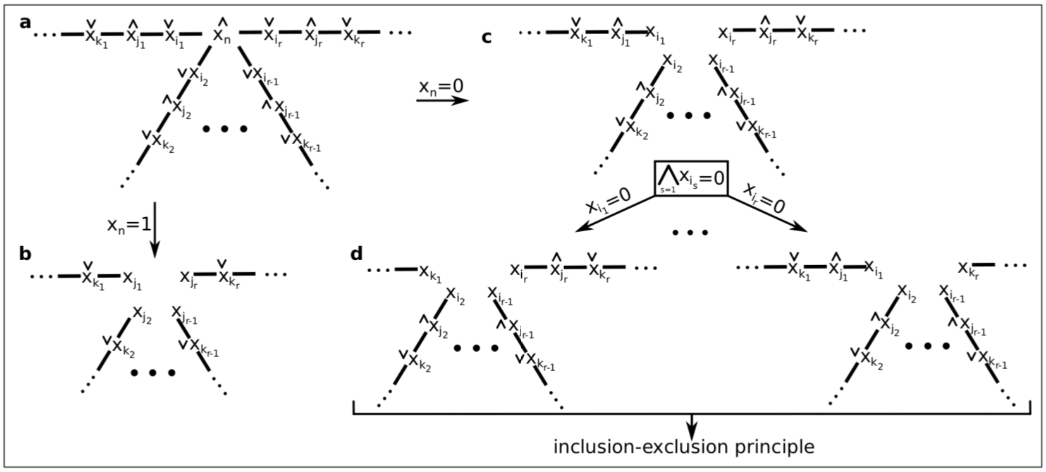

Figure 6. Idea of the proof in Lemma 4.2. (**a**) Wiring diagram of AND-OR network f that has a vertex that depends on 2 or more variables. The three dots represent other variables in the r chains that could potentially intersect as in Fig. 4a. We consider 2 cases, $x_n = 1$ and $x_n = 0$ in the system of equations $f(x) = x$. (**b**) In the case $x_n = 1$, we obtain a smaller system of equations that corresponds to a smaller wiring diagram of an AND-OR network. (**c**) In the case $x_n = 0$, the system of equations does not correspond to an AND-OR network due to the condition $\bigwedge_{s=1}^{r} x_{i_s} = 0$. (**d**) To obtain AND-OR networks we consider the subcases $x_{i_1} = 0$, $x_{i_2} = 0, \ldots$, and $x_{i_r} = 0$. However, there is overlap of fixed points between the different subcases, so we use the inclusion-exclusion principle.

We now claim that $|\{\text{fixed points of the form } x_{i_s} = 0 \text{ for } s \in S\}| = F[f\backslash(\{n\} \cup \{i_s : s \in S\} \cup \{j_s : s \in S\})]$. Indeed, if $x_{i_s} = 0$ for $s \in S$, it follows that $x_{j_s} = 0$ and that x_{k_s} depends on a single variable only for $s \in S$. The AND-OR network corresponding to this is precisely $f\backslash(\{n\}\cup\{i_s : s \in S\} \cup \{j_s : s \in S\})$.

The proof then follows by adding the total number of fixed points from cases $x_n = 1$ and $x_n = 0$.

□

<u>**Lemma 4.3**</u> *Suppose a Boolean network f is the Cartesian product of h and g; that is, up to a relabeling of variables,* $f(x, y) = (g(x), h(y))$ *(also denoted by* $f = g\times h$*). Then,*

$$\mathrm{F}[f] = F[\mathrm{g}]F[h]$$

Proof. Follows from the fact that (x, y) is a steady state of f if and only if x is a steady state of g and y is a steady state of h.

□

With these lemmas we can now find the number of fixed points of the AND-OR

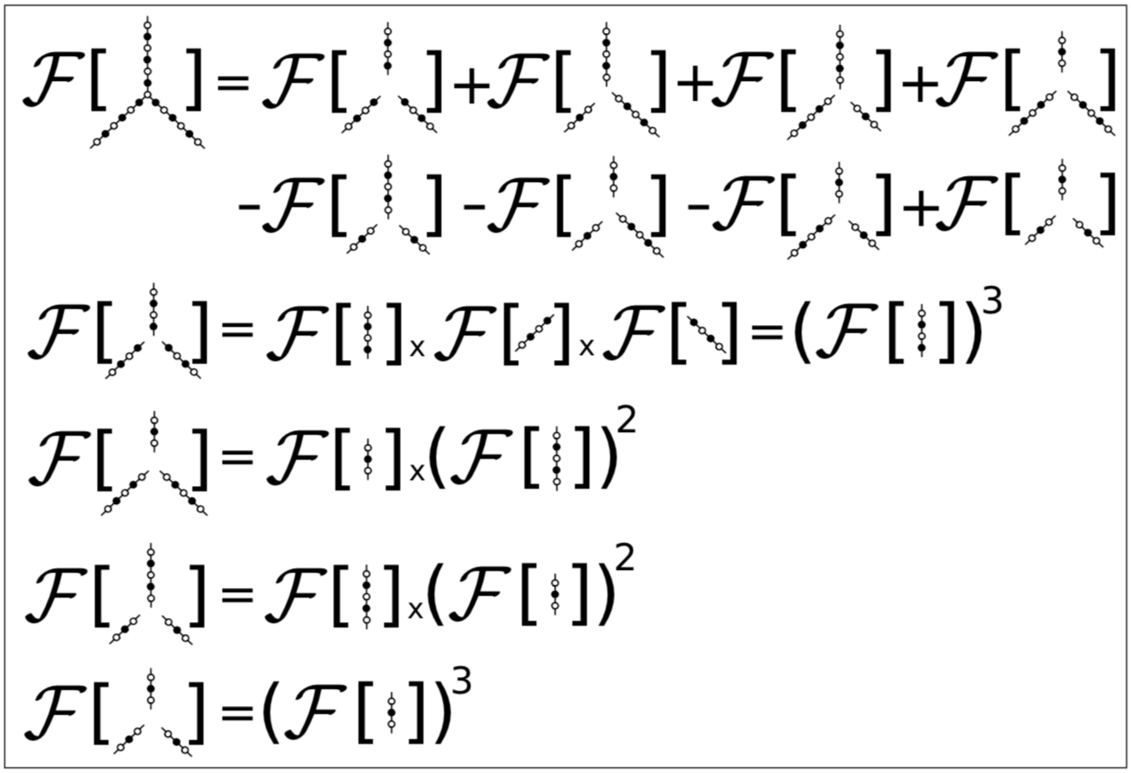

Figure 7. Using our results to find the number of fixed points of a coupling of 3 open chains.

networks given in Fig. 4. For notational purposes, we apply the lemmas using the unlabeled representation of the wiring diagrams.

Example 4.4 *Consider the AND-OR network with wiring diagram given by Fig. 4a. We use Lemma 4.2 to split the wiring diagram at x_8. The process is shown in Fig. 7. Using this lemma, we find that the number of fixed points can be written as a sum/difference of the number of fixed points of disjoint chains. Then, we use Lemma 4.3 to express the number of fixed points as an algebraic combination of the number of fixed points of single chains. Once we have single chains, we can use the results from previous sections. Thus, the number of fixed points is*

$$F[f] = (F(1,1,1,1))^3 + 3\,F(1,1,1)\,(F(1,1,1,1,1))^2 - 3\,F(1,1,1,1,1)\,(F(1,1,1))^2 + (F(1,1,1))^3$$

$$= (5)^3 + 3(4)(7)^2 - 3(7)(4)^2 + (4)^3$$

$$= 441.$$

<u>Example 4.5</u> *Consider the AND-OR network with wiring diagram given by Fig. 4b. We use Lemma 4.2 to split the wiring diagram at x_8 and then we use Lemma 4.3. Analogous to the previous example, we obtain*

$$F[f] = F(1,1,1,1)\, F(1,1,1,1,1) + F(1,1,1)\, F(1,1,1,1,1,1,1) + 2(F(1,1,1,1,1))^2 -$$
$$3\, F(1,1,1)\, F(1,1,1,1,1) + (F(1,1,1))^2$$

$$= (5)(7) + (4)(12) + 2(7)^2\ 3(4)(7) + (4)^2$$

$$= 113.$$

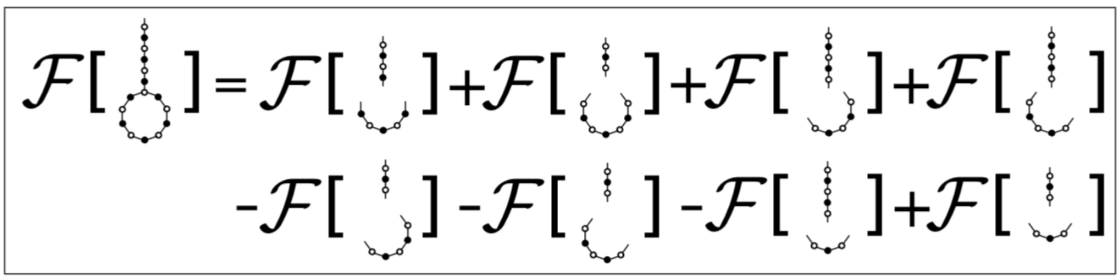

Figure 8. Using our results to find the number of fixed points of a coupling of an open chain and a closed chain.

5. Conclusion

Our results provide recursive formulas and sharp bounds for the number of fixed points of AND-OR networks with chain topology. Other work regarding the number of fixed points has focused on bounds with respect to the number of nodes [6]. Our results, on the other hand, focus on formulas and bounds with respect to the pattern of logical operators. Thus, our findings complement previous results. Our approach can potentially be extended to cases where an AND-OR network has a topology that can be seen as the "combination" of open chains. Then, the number of fixed points of the original AND-OR network will be given by the inclusion-exclusion principle in terms of the number of fixed points of the AND-OR networks with open chain topology. Indeed, Section 4 shows how our approach can be used in such cases.

6. References

[1] T. Akutsu, S. Kuhara, O. Maruyama, S. Miyano, A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions, Genome Inform. 9 (1998) 151–160.

[2] R. Albert, H. Othmer, The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*, J. Theor. Biol. 223 (2003) 1–18.

[3] L. Mendoza, I. Xenarios, A method for the generation of standardized qualitative dynamical systems of regulatory networks, Theoretical Biology and Medical Modelling 3 (1) (2006) 13. doi:10.1186/1742-4682-3-13.

[4] A. Jarrah, R. Laubenbacher, A. Veliz-Cuba, The dynamics of conjunctive and disjunctive Boolean network models, Bull. Math. Bio. 72 (6) (2010) 1425–1447.

[5] Z. Agur, A. Fraenkel, S. Klein, The number of fixed points of the majority rule, Discrete Math. 70(3)(1988)295–302.

[6] J. Aracena, J. Demongeot, E. Goles, Fixed points and maximal independent sets in AND-OR networks, Discrete Appl. Math. 138(3)(2004) 277 –288.

[7] A. Jarrah, B. Raposa, R. Laubenbacher, Nested canalyzing, unate cascade, and polynomial functions, Physica D: Nonlinear Phenomena 233 (2) (2007) 167 – 174.

[8] J. Aracena, Maximum number of fixed points in regulatory Boolean networks, Bulletin of Mathematical Biology 70(5)(2008)1398–1409.

[9] D. Murrugarra, R. Laubenbacher, Regulatory patterns in molecular interaction networks, Journal of Theoretical Biology 288(0)(2011)66–72.

[10] D.Bollman,O.Colo ́n-Reyes, V. A. Ocasio, E. Orozco, A control theory for Boolean monomial dynamical systems, Discrete Event Dynamic Systems 20 (1) (2010) 19–35.

[11] A. Veliz-Cuba, A. Kumar, K. Josić, Piecewise lineareand Boolean models of chemical reaction networks, Bulletin of Mathematical Biology 76 (12) (2014) 2945–2984.

[12] A.Veliz-Cuba, D. Murrugarra, R. Laubenbacher, Structure and dynamics of acyclic networks, Discrete Event Dynamic Systems 24(4)(2014) 647–658.

[13] E. Dimitrova, O. Yordanov, M. Matache, Difference equation for tracking perturbations in systems of Boolean nested canalyzing functions, Phys. Rev. E 91 (2015) 062812.

[14] E. Weiss, M. Margaliot, A polynomial-time algorithm for solving the minimal observability problem in conjunctive Boolean networks, arXiv preprint arXiv:1706.04072.

[15] E.Goles,M.Matamala,P.A.Este ́vez,Dynamicalpropertiesofmin-maxnetworks,InternationalJournalofNeuralSystems10(06)(2000) 467 –473, pMID: 11307860.

[16] A. Hansson, H. Mortveit, C. Reidys, On asynchronous cellular automata, Advances in Complex Systems 08 (04) (2005) 521–538. doi:10.1142/S0219525905000555.

[17] A. Alcolei, K. Perrot, S. Sen, On the flora of asynchronous locally non-monotonic Boolean automata networks, Electronic Notes in Theoretical Computer Science 326 (Supplement C) (2016) 3 – 25, the 6th International Workshop on Static Analysis and Systems Biology, SASB 2015.

[18] A. Veliz-Cuba, Reduction of Boolean network models, Journal of Theoretical Biology 289(2011)167–172.

[19] M. T. Matache, V. Matache, Logical reduction of biological networks to their most determinative components, Bulletin of Mathematical Biology 78 (7) (2016) 1520 – 1545.

[20] S. H. Elmeligy Abdelhamid, C. J. Kuhlman, M. V. Marathe, H. S. Mortveit, S. S. Ravi, Gdscalc: A web-based application for evaluating discrete graph dynamical systems, PLOS ONE 10 (8) (2015) 1–24. doi:10.1371/journal.pone.0133660.

[21] O. S. Center, Ohio supercomputer center, http://osc.edu/ark:/19495/f5s1ph73(1987).