T U M

I N S T I T U T F Ü R I N F O R M A T I K

Dichotomy Results for Fixed Point Counting in Boolean Dynamical Systems

Sven Kosub Christopher M. Homan

TUM-I0706
Januar 07

T E C H N I S C H E U N I V E R S I T Ä T M Ü N C H E N

TUM-INFO-01-I0706-0/1.-FI

Druck: Institut für Informatik der
Technischen Universität München

Dichotomy Results for Fixed Point Counting in Boolean Dynamical Systems

Sven Kosub
Fakultät für Informatik
Technische Universität München,
D-85748 Garching, Germany
`kosub@in.tum.de`

Christopher M. Homan
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623, USA
`cmh@cs.rit.edu`

Abstract

We present dichotomy theorems regarding the computational complexity of counting fixed points in boolean (discrete) dynamical systems, i.e., finite discrete dynamical systems over the domain $\{0,1\}$. For a class $\mathcal{F}$ of boolean functions and a class $\mathcal{G}$ of graphs, an $(\mathcal{F},\mathcal{G})$-system is a boolean dynamical system with local transitions functions lying in $\mathcal{F}$ and graph in $\mathcal{G}$. We show that, if local transition functions are given by lookup tables, then the following complexity classification holds: Let $\mathcal{F}$ be a class of boolean functions closed under superposition and let $\mathcal{G}$ be a graph class closed under taking minors. If $\mathcal{F}$ contains all min-functions, all max-functions, or all self-dual and monotone functions, and $\mathcal{G}$ contains all planar graphs, then it is #P-complete to compute the number of fixed points in an $(\mathcal{F},\mathcal{G})$-system; otherwise it is computable in polynomial time. The theorem relies on an evident conjecture for an open case. In contrast, we prove a dichotomy theorem for the case that local transition functions are given by formulas (over logical bases). A corresponding theorem for boolean circuits coincides with the theorem for formulas.

1 Introduction

Efforts to understand the behavior of complex systems have led to various models for finite discrete dynamical systems, including (finite) cellular automata (see, e.g., [32, 33]), discrete recurrent Hopfield networks (see, e.g., [17, 1]), and concurrent and communicating finite state machines (see, e.g., [20, 23]). A fairly general class of systems was introduced in [6]. There, a finite discrete dynamical system (over a finite domain $\mathcal{D}$) is defined as: (a) a finite undirected graph, where vertices correspond to variables and edges correspond to an interdependence between the two connected variables, (b) for each vertex v, a local transition function that maps tuples of values (belonging to $\mathcal{D}$) of v and v's neighbors to values of v, and (c) an update schedule that governs which variables are allowed to update their values in which time steps. Formal definitions can be found in Sect. 2.

A central goal in the study of dynamical systems is to classify them according to how easy it is to predict their behavior. In a finite, discrete setting, a certain behavioral pattern is considered predictable if it can be decided in polynomial time whether a given system will show the pattern [10]. Although the pattern reachability problem is, in general, an intractable problem, i.e., at least NP-hard (see, e.g., [15, 26, 4]), many tractable classes of patterns and systems have been identified. However, there is still a serious demand for exhaustive characterizations of *islands of predictability*.

A fundamental behavioral pattern is the *fixed point* (a.k.a., homogeneous state, or equilibrium). A value assignment to the variables of a system is a fixed point if the values assigned to the variables are left unchanged after the system updates them. Note that fixed points are invariant under changes of the update regime. In this sense, they can be seen as a particularly robust behavior. A series of recent papers has been devoted to the identification of finite systems with tractable/intractable fixed-point analyses [5, 29, 27, 28, 18]. Precise boundaries are known for which systems finding fixed points can be done in polynomial time. For the fixed-point counting problem this is far less so.

Contributions of the paper. We prove dichotomy theorems on the computational complexity of counting fixed points in boolean (discrete) dynamical systems, i.e., finite discrete dynamical systems over the domain $\{0, 1\}$. For a class $\mathcal{F}$ of boolean functions and a class $\mathcal{G}$ of graphs, an $(\mathcal{F}, \mathcal{G})$-system is a boolean dynamical system with local transition functions lying in $\mathcal{F}$ and a graph lying in $\mathcal{G}$. Following [18], Post classes (a.k.a., clones) and forbidden-minor classes are used to classify $(\mathcal{F}, \mathcal{G})$-systems. In Sect. 4 we state the following theorem (Theorem 8): Let $\mathcal{F}$ be a class of boolean function closed under superposition and let $\mathcal{G}$ be a minor-closed graph class. If $\mathcal{F}$ contains all min-functions, all max-functions, or all self-dual and monotone functions, and $\mathcal{G}$ contains all planar graphs, then it is #P-complete to compute the number of the fixed points in an $(\mathcal{F}, \mathcal{G})$-system; otherwise it is computable in polynomial time. Here, the local transition functions are supposed to be given by lookup tables. In fact, we prove a conditional version of the theorem requiring an evident conjecture for the only open case. In addition, we prove a dichotomy theorem (Theorem 18) for the case that local transition functions are given by formulas (over logical bases). Moreover, the corresponding theorem for boolean circuits coincides with the theorem for formulas. The theorem has a significantly more complicated structure than for lookup tables.

Related work. There is a series of work regarding the complexity of certain computational problems for finite discrete dynamical systems (see, e.g., [15, 26, 5, 2, 3, 29, 27, 4] and the references therein). The problem of counting fixed points of boolean dynamical systems has been studied in [29, 27, 28]. To summarize: counting the number of fixed points is in general #P-complete. So is counting the number of fixed points for boolean dynamical systems with monotone local transition functions over planar bipartite graphs or over uniformly sparse graphs. We note that all system classes considered here are based on formula or circuit representations. That is, if they fit into our scheme at all, then the intractability results fall into the scope of Theorem 18 (and are covered there). Detailed studies of computational problems related to fixed-point existence have been reported in [5, 18]. In [18], a complete classification of the fixed-point existence problem with respect to the analysis framework we use in this paper was shown.

2 The Dynamical Systems Framework

In this section we present a formal framework for dynamical systems. A fairly general approach is motivated by the theoretical study of simulations. The following is based on [8, 6, 7, 18].

The underlying network structure of a dynamical system is given by an undirected graph $G = (V, E)$ without multi-edges and loops. We suppose that the set V of vertices is ordered. So, without loss of generality, we assume $V = \{1, 2, \ldots, n\}$. For any vertex set $U \subseteq V$, let $N_G(U)$ denote the neighbors of U in G, i.e.,

$$N_G(U) =_{\text{def}} \{\ j \mid j \notin U \text{ and there is an } i \in U \text{ such that } \{i, j\} \in E\ \}.$$

If $U = \{i\}$ for some vertex i, then we use $N_G(i)$ as a shorthand for $N_G(\{i\})$. The degree d_i of a vertex i is the number of its neighbors, i.e., $d_i =_{\text{def}} \|N_G(i)\|$.

A *dynamical system S over a domain $\mathcal{D}$* is a pair (G, F) where $G = (V, E)$ is an undirected graph (the *network*) and $F = \{f_i \mid i \in V\}$ is a set of *local transition functions* $f_i : \mathcal{D}^{d_i+1} \to \mathcal{D}$. The intuition of the definition is that each vertex i corresponds to an active element (entity, agent, actor etc.) which is always in some state x_i and which is capable to change its state, if necessary. The domain of S formalizes the set of possible states of all vertices of the network, i.e., for all $i \in V$, it always holds that $x_i \in \mathcal{D}$. A vector $\vec{x} = (x_i)_{i \in V}$ such that $x_i \in \mathcal{D}$ for all $i \in V$ is called a *configuration of* S. The local transition function f_i for some vertex i describes how i changes its state depending on the states of its neighbors $N_G(i)$ in the network and its own state.

We are particularly interested in dynamical system operating on a discrete time-scale. A *discrete dynamical system* $\mathcal{S} = (S, \alpha)$ consists of a dynamical system S and a mapping $\alpha : \{1, \ldots, T\} \to \mathcal{P}(V)$, where V is a set of vertices of the network of S and $T \in \mathbb{N}$. The mapping α is called the *update schedule* and specifies which states updates are realized at certain time-steps: for $t \in \{1, \ldots, T\}$, $\alpha(t)$ specifies those vertices that simultaneously update their states in step t.

A discrete dynamical system $\mathcal{S} = (S, \alpha)$ over domain $\mathcal{D}$ induces a global map $\mathbf{F}_{\mathcal{S}} : \mathcal{D}^n \to \mathcal{D}^n$ where n is the number of vertices of S. For each vertex $i \in V$, define an *activity function* φ_i for a set $U \subseteq V$ and $\vec{x} = (x_1, \ldots, x_n) \in \mathcal{D}^n$ by

$$\varphi_i[U](\vec{x}) =_{\text{def}} \begin{cases} f_i(x_{i_1}, \ldots, x_{i_{d_i+1}}) & \text{if } i \in U \\ x_i & \text{if } i \notin U \end{cases}$$

where $\{i_1, i_2, \ldots, i_{d_i+1}\} = \{i\} \cup N_G(i)$. For a set $U \subseteq V$, define the *global transition function* $\mathbf{F}_S[U] : \mathcal{D}^n \to \mathcal{D}^n$ for all $\vec{x} \in \mathcal{D}^n$ by

$$\mathbf{F}_S[U](\vec{x}) =_{\text{def}} (\varphi_1[U](\vec{x}), \ldots, \varphi_n[U](\vec{x})).$$

Note that the global transition function does not refer to the update schedule, i.e., it only depends on the dynamical system S and not on $\mathcal{S}$. The function $F_{\mathcal{S}} : \mathcal{D}^n \to \mathcal{D}^n$ computed by the discrete dynamical system $\mathcal{S}$, the *global map* of $\mathcal{S}$, is defined by

$$\mathbf{F}_{\mathcal{S}} =_{\text{def}} \prod_{k=1}^{T} \mathbf{F}_S[\alpha(k)].$$

The central notion for our study of dynamical systems is the concept of a fixed point, i.e., a configuration which does not change under any global behavior of the system. Let $S = (G, \{f_i \mid i \in V\})$ be a dynamical system over domain $\mathcal{D}$. A configuration $\vec{x} \in \mathcal{D}^n$ is said to be a *local fixed point of S for $U \subseteq V$* if and only if $\mathbf{F}_S[U](\vec{x}) = \vec{x}$. A configuration $\vec{x} \in \mathcal{D}^n$ is said to be a *fixed point of* S if and only if $\vec{x}$ is a local fixed point of S for V. Note that a fixed point does not depend on a concrete update schedule: a configuration $\vec{x} \in \mathcal{D}^n$ is a fixed point of S if and only if for all update schedules $\alpha : \{1, \ldots, T\} \to \mathcal{P}(V)$, it holds that $\mathbf{F}_{(S,\alpha)}(\vec{x}) = \vec{x}$.

3 The Analysis Framework

In this section we specify our analysis framework for $(\mathcal{F}, \mathcal{G})$-systems. Following [18], local transition functions are classified by Post classes, i.e., superpositionally closed classes of boolean functions, and graphs are classified using the theory of graph minors as a tool. In the following we gather relevant notation.

3.1 Transition Classes

We adopt notation from [9]. An n-ary boolean function f is a mapping $f : \{0,1\}^n \to \{0,1\}$. Let BF denote the class of all boolean functions. There are two 0-ary boolean functions: $c_0 =_{\text{def}} 0$ and $c_1 =_{\text{def}} 1$ (which are denoted in formulas by the symbols 0 and 1). There are two 1-ary boolean functions: $\text{id}(x) =_{\text{def}} x$ and $\text{not}(x) =_{\text{def}} 1 - x$ (which are denoted in formulas by x for $\text{id}(x)$ and $\overline{x}$ for $\text{not}(x)$).

We say that a class $\mathcal{F}$ is *Post* if and only if $\mathcal{F}$ contains the function id and $\mathcal{F}$ is closed under the introduction of fictive variables, permutations of variables, identification of variables, and substitution (see, e.g., [9] for definitions). It is a famous theorem by Post [21] that the family of all Post classes is a countable lattice with respect to set inclusion. In particular, each Post class is the intersection of a finite set of meet-irreducible classes, which are the following:

- *The classes* R_0 *and* R_1*.* For $b \in \{0,1\}$, a boolean function f is said to be *b-reproducing* if and only if $f(b,\ldots,b) = b$. Let R_b denote the class of all b-reproducing functions.

- *The class* M. For binary n-tuples $\vec{a} = (a_1,\ldots,a_n)$ and $\vec{b} = (b_1,\ldots,b_n)$, we say that $(a_1,\ldots,a_n) \leq (b_1,\ldots,b_n)$ if and only if for all $i \in \{1,\ldots,n\}$, it holds that $a_i \leq b_i$. An n-ary boolean function f is said to be *monotone* if and only if for all $\vec{x},\vec{y} \in \{0,1\}^n$, $\vec{x} \leq \vec{y}$ implies $f(\vec{x}) \leq f(\vec{y})$. Let M denote the class of all monotone boolean functions.

- *The class* D. An n-ary boolean function f is said to be *self-dual* if and only if for all $(x_1,\ldots,x_n) \in \{0,1\}^n$, it holds that $f(x_1,\ldots,x_n) = \text{not}(f(\text{not}(x_1),\ldots,\text{not}(x_n)))$. Let D denote the class of all self-dual functions.

- *The class* L*.* A boolean function f is linear if and only if there exists constants $a_1,\ldots,a_n \in \{0,1\}$ such that $f(x_1,\ldots,x_n) = a_0 \oplus a_1x_1 \oplus \cdots \oplus a_nx_n$. Note that $\oplus$ is understood as addition modulo 2 and xy is understood as multiplication modulo 2. Let L denote the class of all linear functions. The logical basis of L is $\{\oplus, 0, 1\}$.

- *The classes* S_b *and* S_b^k*.* For $b \in \{0,1\}$, a tuple set $T \subseteq \{0,1\}^n$ is said to be b-separating if and only if there is an $i \in \{1,\ldots,n\}$ such that for $(t_1,\ldots,t_n) \in T$ holds $t_i = b$. A boolean function f is *b-separating* if and only if $f^{-1}(b)$ is b-separating. A function f is called *b-separating of level k* if and only if every $T \subseteq f^{-1}(b)$ such that $\|T\| = k$ is b-separating. Let S_b denote the class of b-separating functions and let S_b^k denote the class of all functions which are b-separating of level k.

- *The classes* E *and* V*.* We denote by E the class of all AND functions, i.e., the class of all functions f, the arity of which is n, such that for some set $J \subseteq \{1,\ldots,n\}$, the equality $f(x_1,\ldots,x_n) = \min_{i \in J} x_i$ is satisfied for all $x_1,\ldots,x_n \in \{0,1\}$. The logical basis over E is $\{\wedge, 0, 1\}$. Dually, we denote by V the class of all OR functions, i.e., the class of all functions f, the arity of which is n, such that for some set $J \subseteq \{1, \ldots, n\}$, the equality $f(x_1, \ldots, x_n) = \max_{i \in J} x_i$ is satisfied for all $x_1, \ldots, x_n \in \{0, 1\}$. The logical basis of V is $\{\vee, 0, 1\}$.

- *The class* N. An n-ary boolean function f is a projection if and only if there is an $i \in \{1, \ldots, n\}$ such that for all $x_1, \ldots, x_n \in \{0, 1\}$, it holds that $f(x_1, \ldots, x_n) = x_i$. A boolean function f is the negation of a projection if and only if there is an $i \in \{1, \ldots, n\}$ such that for all $x_1, \ldots, x_n \in \{0, 1\}$, it holds that $f(x_1, \ldots, x_n) = \mathrm{not}(x_i)$. A boolean function f is constant if and only if there exists a $b \in \{0, 1\}$ such that for all $x_1, \ldots, x_n \in \{0, 1\}$, it holds that $f(x_1, \ldots, x_n) = b$. Let N denote the class of boolean functions which are projections, negations of projections, or constant functions.

Note that the classes possess the following inclusion structure (see, e.g., [9]):

- $\mathrm{S}_0 \subseteq \cdots \subseteq \mathrm{S}_0^k \subseteq \mathrm{S}_0^{k-1} \subseteq \cdots \subseteq \mathrm{S}_0^2 \subseteq \mathrm{R}_1$
- $\mathrm{S}_1 \subseteq \cdots \subseteq \mathrm{S}_1^k \subseteq \mathrm{S}_1^{k-1} \subseteq \cdots \subseteq \mathrm{S}_1^2 \subseteq \mathrm{R}_0$
- $\mathrm{E} \subseteq \mathrm{M}$ and $\mathrm{V} \subseteq \mathrm{M}$
- $\mathrm{N} \subseteq \mathrm{L}$

No other inclusions hold among these classes. Moreover, all Post classes have a finite logical basis. Particular relevance for our studies have the following classes:

$$
\begin{array}{rcll}
\mathrm{D}_2 & =_{\mathrm{def}} & \mathrm{D} \cap \mathrm{M} & \text{with logical basis } \{(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)\} \\
\mathrm{D}_1 & =_{\mathrm{def}} & \mathrm{D} \cap \mathrm{R}_0 \cap \mathrm{R}_1 & \text{with logical basis } \{(x \wedge y) \vee (x \wedge \overline{z}) \vee (y \wedge \overline{z})\} \\
\mathrm{S}_{00} & =_{\mathrm{def}} & \mathrm{S}_0 \cap \mathrm{M} \cap \mathrm{R}_0 & \text{with logical basis } \{x \vee (y \wedge z)\} \\
\mathrm{S}_{10} & =_{\mathrm{def}} & \mathrm{S}_1 \cap \mathrm{M} \cap \mathrm{R}_1 & \text{with logical basis } \{x \wedge (y \vee z)\} \\
\mathrm{E}_2 & =_{\mathrm{def}} & \mathrm{E} \cap \mathrm{S}_{10} & \text{with logical basis } \{\wedge\} \\
\mathrm{V}_2 & =_{\mathrm{def}} & \mathrm{V} \cap \mathrm{S}_{00} & \text{with logical basis } \{\vee\}
\end{array}
$$

3.2 Network Classes

We adopt notation from [13]. Let X and Y be two undirected graphs. We say that X is minor of Y if and only if there is a subgraph Y' of Y such that X can be obtained by contracting edges of Y'. Let $\preceq$ be the relation on graphs defined by $X \preceq Y$ if and only if X is a minor of Y. A class $\mathcal{G}$ of graphs is said to be *closed under taking minors* if and only if for all graphs G and G', if $G \in \mathcal{G}$ and $G' \preceq G$, then $G' \in \mathcal{G}$. Let $\mathcal{X}$ be any set of graphs. $\mathrm{Forb}_{\preceq}(\mathcal{X})$ denotes the class of all graphs without a minor in $\mathcal{X}$ (and which is closed under isomorphisms). More specifically, $\mathrm{Forb}_{\preceq}(\mathcal{X}) =_{\mathrm{def}} \{G \mid G \not\succeq X \text{ for all } X \in \mathcal{X}\}$. The set $\mathcal{X}$ is called the set of *forbidden minors*. Note that $\mathrm{Forb}_{\preceq}(\emptyset)$ is the class of all graphs. As usual, we write $\mathrm{Forb}_{\preceq}(X_1, \ldots, X_n)$ instead of $\mathrm{Forb}_{\preceq}(\{X_1, \ldots, X_n\})$. Forbidden-minor classes are monotone with respect to $\preceq$, i.e., $X \preceq Y$ implies $\mathrm{Forb}_{\preceq}(X) \subseteq \mathrm{Forb}_{\preceq}(Y)$. The celebrated Graph Minor Theorem, due to Robertson and Seymour [25], shows that there are only countably many network classes closed under taking minors: A class $\mathcal{G}$ of graphs is closed under taking minors if and only if there is a finite set $\mathcal{X}$ such that $\mathcal{G} = \mathrm{Forb}_{\preceq}(\mathcal{X})$.

Two graph classes are particularly relevant to our study: planar graphs and graphs having a vertex cover of size one. Let K^n denote the complete graphs on n vertices and let $K_{n,m}$ denote the complete bipartite graph having n vertices in one component and m vertices in the other component. The well-known Kuratowski-Wagner theorem (see, e.g., [13]) states that a graph G is planar if and only if G belongs to $\mathrm{Forb}_{\preceq}(K_{3,3}, K^5)$. Moreover, a graph X is planar if and only if $\mathrm{Forb}_{\preceq}(X)$ has bounded treewidth [24]. As we use the treewidth of a graph only in a black-box fashion, we refer to, e.g., [13] for a definition. A class $\mathcal{G}$ of graphs is said to have *bounded treewidth* if and only if there is a $k \in \mathbb{N}$ such that all graphs in the class have treewidth at most k. Let $G = (V, E)$ be a graph. We say that a subset $U \subseteq V$ is a *vertex cover* of G if and only if for all edges $\{u, v\} \in E$, it holds that $\{u, v\} \cap U \neq \emptyset$. It is known that the class of graphs having a vertex cover of size at most k is closed under taking minors [11]. Moreover, G has a vertex cover of size one if and only if G belongs to $\mathrm{Forb}_{\preceq}(K^3 \oplus K^2)$ [11], where for graphs G and G', $G \oplus G'$ denotes the graph obtained by the disjoint union of G and G'. A class of graphs is said to have *bounded degree* if and only if there is a $k \in \mathbb{N}$ such that all graphs in the class have a maximum vertex-degree of at most k. It is known that a graph X has a vertex cover of size one if and only if $\mathrm{Forb}_{\preceq}(X)$ has bounded degree (cf., e.g., [18]).

4 Islands of Tractability for Fixed Point Counting

In this section we are interested in the computational complexity of the following counting problem. Let $\mathcal{F}$ be a class of boolean functions and let $\mathcal{G}$ be a class of graphs.

Problem:	#FIXEDPOINTS$(\mathcal{F}, \mathcal{G})$
Input:	An $(\mathcal{F}, \mathcal{G})$-system S, i.e., a boolean dynamical system $S = (G, \{f_1, \dots, f_n\})$ such that $G \in \mathcal{G}$ and for all $i \in \{1, \dots, n\}$, $f_i \in \mathcal{F}$
Output:	The number of fixed points of S

The complexity of the problem depends on how transition functions are represented. We consider the cases of lookup table, formula, and circuit representations. The corresponding problems are denoted by #FIXEDPOINTS$_\mathrm{T}$, #FIXEDPOINTS$_\mathrm{F}$, and #FIXEDPOINTS$_\mathrm{C}$. It is obvious that all problem versions belong to #P. We say that a problem is intractable if it is #P-hard (with respect to Turing reductions, as described in, e.g., [16]), and it is tractable if it is solvable in polynomial time.

4.1 The Case of Local Transition Functions Given By Lookup Tables

We start by identifying tractable counting problems.

Lemma 1. #FIXEDPOINTS$_\mathrm{T}(\mathrm{L}, \mathrm{Forb}_{\preceq}(\emptyset))$ *is solvable in polynomial time.*

Proof. Notice that for a linear function $f(x_1, \dots, x_n) = a_0 \oplus a_1 x_2 \oplus a_2 x_2 \oplus \dots \oplus a_n x_n$, the proposition $x_i \leftrightarrow [a_0 \oplus a_1 x_2 \oplus a_2 x_2 \oplus \dots \oplus a_n x_n]$ is true if and only if $a_0 \oplus a_1 x_2 \oplus a_2 x_2 \oplus \dots \oplus a_n x_n \oplus x_i \oplus 1$ is satisfiable. So, each dynamical system with linear, boolean local transition functions constitutes a system of linear equations over Z_2, for which the number of solutions can be computed in polynomial time using Gaussian elimination (cf. [12]). □

In [18], it has been shown that the decision version of #$\text{FIXEDPOINTS}_\text{T}(\text{BF}, \text{Forb}_\preceq(X))$ for planar graphs X can be solved in polynomial time. This result is obtained by a reduction to a certain type of constraint satisfaction problems. Actually, the reduction establishes injections between the fixed points of a dynamical system and the satisfying assignments of the corresponding constraint satisfaction problem. Consequently, the numbers of fixed points and the numbers of satisfying assignments are equal.

Lemma 2. *Let X be a planar graph. Then,* #$\text{FIXEDPOINTS}(\text{BF}, \text{Forb}_\preceq(X))$ *is solvable in polynomial time.*

Proof. By inspection of [18] and noting that counting satisfying assignments for constraint satisfaction problems having constraint graphs of bounded treewidth can be done in polynomial time (cf. [14]). □

We turn to the intractable fixed-point counting problems. Let H be a 2CNF such that each clause consists of exactly one positive and one negative literal. H is called a Horn-2CNF formula. Moreover, suppose H has a planar graph representation, i.e., the graph $\Gamma(H) = (V, E)$ with vertex set $V = \{x_1, \dots, x_n, C_1, \dots, C_m\}$, where the x_i's are the variables and the C_i's are the clauses of H, and edge set $E = \{\{x_i, C_j\} \mid x_i \text{ is a variable in } C_j\}$ is planar. Then, H is called a planar Horn 2-CNF formula. #PLANAR HORN-2SAT is the problem of counting all satisfying assignments of a given planar Horn-2CNF formula.

Proposition 3. #PLANAR HORN-2SAT *is* #P*-complete even if each variable is allowed to occur in four clauses only.*

Proof. In [30], it has been shown that the following problem is #P-complete: #4Δ-PLANAR BIPARTITE INDEPENDENT SET, i.e., compute, on a given bipartite graph $G = (V, E)$ with maximum vertex-degree at most four, the number of independent sets $U \subseteq V$. Let $G = (V, E)$ be a bipartite graph, $V = V_1 \cup V_2$ and $E \subseteq V_1 \times V_2$. Define H to be the 2CNF given by clauses $(x_u \vee \overline{x_v})$ for all $u \in V_1$, $v \in V_2$ such that $\{u, v\} \in E$. Clearly, H is a Horn-2CNF formula. Moreover, if G is planar and the maximum degree is at most four, the graph representation of H is planar and each variable occurs at most four times in H. Finally, it is easily seen that there is a bijection between the independent sets of G and the satisfying assignments for H (cf., e.g., [22, 19]). Hence, #4Δ-PLANARBIPARTITE INDEPENDENT SET reduces to #PLANAR HORN-2SAT with each variable occuring in at most four clauses. □

Lemma 4. #$\text{FIXEDPOINTS}_\text{T}(\text{E}_2, \text{Forb}_\preceq(K_{3,3}, K^5))$ *is* #P*-complete.*

Proof. We reduce from #PLANAR HORN-2SAT assuming that each variable occurs only four times in the formula. Let $H = C_1 \wedge \cdots \wedge C_m$ be a planar Horn-2CNF formula. Define a dynamical system $S = (G, F)$ as follows. $G = (V, E)$ is given by $V =_{\text{def}} \{1, \dots, n\}$ and $E =_{\text{def}} \{\ \{i, j\} \mid (\overline{x_i} \vee x_j) = C_r \text{ for some } r \in \{1, \dots, m\}\ \}$. Since H has a planar graph representation, G is planar, i.e., $G \in \text{Forb}_\preceq(K_{3,3}, K^5)$. The local transition functions are specified in the following way. For a vertex $i_0 \in V$ let $\{i_1, \dots, i_r\}$ be the set of all vertices such that $(\overline{x_{i_j}} \vee x_{i_0})$ is a clause in H. Then, f_{i_0} is the function given by the formula $H_{i_0} = x_{i_0} \wedge x_{i_1} \wedge \cdots \wedge x_{i_r}$. Notice that all local transition functions belong to E_2 and also notice that the maximum degree of a vertex in G is four. Thus, we can compute the lookup tables in polynomial time depending on the size of H. Moreover, it is easily seen that $(x_{i_0} \leftrightarrow \bigwedge_{j=1}^{r} x_{i_j}) \equiv \bigwedge_{j=1}^{r} (\overline{x_{i_j}} \vee x_{i_0})$. Hence, the number of satisfying assignments of H is equal to the number of fixed-point configurations of S_H. This shows that #PLANAR HORN-2SAT reduces to $\#\text{FIXEDPOINTS}_\text{T}(\text{E}_2, \text{Forb}_\preceq(K_{3,3}, K^5))$. □

Lemma 5. $\#\text{FIXEDPOINTS}_\text{T}(\text{V}_2, \text{Forb}_\preceq(K_{3,3}, K^5))$ *is* #P*-complete.*

Proof. Again we reduce from #PLANAR HORN-2SAT assuming that each variable occurs only four times in the formula. Let $H = C_1 \wedge \cdots \wedge C_m$ be a planar Horn-2CNF formula. We construct the same network as in the proof of Lemma 4 on a given planar Horn-2CNF formula H having the variables $x_1, \dots, x_n$. However, the local transition functions are specified as follows. For a vertex $i_0 \in V$, let $\{i_1, \dots, i_r\}$ be the set of all vertices such that $(\overline{x_{i_0}} \vee x_{i_j})$ is a clause in H. Then, f_{i_0} is the function given by the formula $H_{i_0} = x_{i_0} \vee x_{i_1} \vee \cdots \vee x_{i_r}$ which clearly belongs to V_2. It remains to verify the number of satisfying assignments of H equals the number of fixed-point configuration of S_H. This follows from $(x_{i_0} \leftrightarrow \bigvee_{j=1}^{r} x_{i_j}) \equiv \bigwedge_{j=1}^{r}(\overline{x_{i_0}} \vee x_{i_j})$. Hence, #PLANAR HORN-2SAT reduces to $\#\text{FIXEDPOINTS}_\text{T}(\text{V}_2, \text{Forb}_\preceq(K_{3,3}, K^5))$. □

Proposition 6. $\#\text{FIXEDPOINTS}_\text{T}(\text{D}_1, \text{Forb}_\preceq(K_{3,3}, K^5))$ *is* #P*-complete.*

Proof. We reduce from the #PLANAR HORN-2SAT version where each variable occurs only up to four times. Let $H = C_1 \wedge \cdots \wedge C_m$ be a planar Horn 2CNF formula having variables $x_1, \dots, x_n$. We define a dynamical system $S_H = (G, \{f_1, \dots, f_n\})$ as follows. The network $G = (V, E)$ is given by

$$\begin{aligned} V &=_{\text{def}} \{1, \dots, n, n+1, \dots n+m\} \\ E &=_{\text{def}} \{\, \{i, j\,\} \mid (i \leq n) \text{ and } (j > n) \text{ and } (x_i \text{ is a variable in clause } C_{j-n})\}. \end{aligned}$$

That is, G is isomorphic to the graph representation of H. Hence, G is a planar graph. Moreover, the maximum degree of G is at most four. First, assume that G is connected. The local transition functions are specified as follows. For each $i \in \{1, \dots, n\}$, let $\{j_1, \dots, j_r\} = N_G(i)$. Define

$$f_i(x_i, x_{j_1}, \dots, x_{j_r}) =_{\text{def}} \begin{cases} x_i & \text{if } x_{j_1} = \cdots = x_{j_r} \\ \text{not}(x_i) & \text{otherwise} \end{cases}$$

It is easily seen that for each $i \in \{1, \dots, n\}$, $f_i \in \text{D}_1$. Note that, since G is connected and because each x_{j_ℓ} is an argument to exactly two of these functions, the above functions require that all variables x_j for $j \in \{n+1, \dots, n+m\}$, i.e., all clauses, be equal in order to have a fixed point. For each $i \in \{n+1, \dots, n+m\}$, let (without loss of generality) $C_i = (\overline{x_{i_1}} \vee x_{i_2})$. Define a formula

$$H_i(x_i, x_{i_1}, x_{i_2}) =_{\text{def}} \begin{cases} \overline{x_{i_1}} \wedge x_{i_2} & \text{if } x_i = 0 \\ \overline{x_{i_1}} \vee x_{i_2} & \text{if } x_i = 1 \end{cases}$$

Let f_i be the function represented by H_i. Truth-table inspection shows that for each $i \in \{n+1, \dots, n+m\}$, $f_i \in \text{D}_1$. Note that the lookup tables for the local transition functions have at most 2^5 entries. Also, it is easy to check that for every satisfying assignment of H there are exactly two fixed points in S_H, i.e, the number of fixed-point configurations of S_H is equal to $2 \cdot \#_+(H)$, where $\#_+(H)$ is the number of satisfying assignments of H. To see this, note that a configuration $\vec{x}$ is a fixed point exactly when $x_{n+1} = \cdots = x_{n+m} = 1$ and the values $x_1, \dots, x_n$ are a satisfying assignment to H, or when $x_{n+1} = \cdots = x_{n+m} = 0$ and $x_1, \dots, x_n$ are a satisfying assignment to H', the formula that is the same as H except that each literal is negated (so, e.g., literals with negations in H no longer have negations in H'). Now, assume that G is not connected. Then, the arguments apply independently to each connected component of G, so the total number of fixed points becomes $2^s \cdot \#_+(H)$, where s is the number of connected components G has. All in all, this shows that #PLANAR HORN-2SAT reduces to $\#\text{FIXEDPOINTS}_{\text{T}}(\text{D}_1, \text{Forb}_{\preceq}(K_{3,3}, K^5))$. $\square$

The remaining case of D_2 functions is special. We conjecture that the fixed-point counting problem for dynamical systems with local transition functions in D_2 and planar networks is intractable. This is based on the following weaker proposition.

Proposition 7. $\#\text{FIXEDPOINTS}_{\text{T}}(\text{D}_2, \text{Forb}_{\preceq}(\emptyset))$ *is* #P-*complete.*

Proof. We reduce from the #PLANAR HORN-2SAT version where each variable occurs only up to four times. Let $H = C_1 \wedge \cdots \wedge C_m$ be a planar Horn 2CNF formula having variables $x_1, \ldots, x_n$. We define a dynamical system $S_H = (G, \{f_1, \ldots, f_n\})$ as follows. The network $G = (V, E)$ is given by

$$\begin{aligned} V &=_{\text{def}} \{1, \ldots, n, n+1, \ldots, 2n\} \\ E &=_{\text{def}} \{\ \{i, j\ \} \mid (\overline{x_i} \vee x_j) \text{ or } (x_i \vee \overline{x_j}) \text{ is a clause of } H\ \} \\ &\qquad\qquad \cup\ \{\ \{i, i+n\} \mid i \in \{1, \ldots, n\}\ \}\ \cup\ \{\ \{j, j+1\} \mid j \in \{n, \ldots, 2n-1\} \end{aligned}$$

The local transition functions are specified as follows. Set $f_{2n}(x_n, x_{2n-1}, x_{2n}) =_{\text{def}} x_{2n}$, $f_{n+1}(x_1, x_{n+1}, x_{n+2}) =_{\text{def}} x_{n+2}$, and for $n+1 < i < 2n$, define $f_i(x_{i-n}, x_{i-1}, x_i, x_{i+1}) =_{\text{def}} x_{i+1}$. For $i \in U$, let $\{0, i_1, \ldots, i_r\} = N_G(i) \cup \{i\}$ and let J_i denote the set of all vertices $j \in N_G(i) \cap \{1, \ldots, n\}$ such that $(\overline{x_i} \vee x_j)$ is a clause in H. We define a formula

$$H_i(x_{i_1}, \ldots, x_{i_r}, x_{i+n}) =_{\text{def}} \begin{cases} x_i \wedge \bigwedge_{j \in J_i} x_j & \text{if } x_{i+n} = 0 \\ x_i \vee \bigvee_{j \in J_i} x_j & \text{if } x_{i+n} = 1 \end{cases}$$

Let f_i be the boolean function represented by H_i. It is easily seen that for all $i \in V$, we have $f_i \in \text{D}_2$. Note that the lookup tables for the local transition functions have at most 2^6 entries. Thus, S_H is computable in time polynomial in the size of H. Moreover, a simple analysis shows that the number of fixed-point configurations of S_H is equal to $2 \cdot \#_+(H)$, where $\#_+(H)$ is the number of satisfying assignments of H. For this to obtain, observe that any fixed-configuration $\vec{x}$ satisfies $x_{n+1} = \cdots = x_n$. Thus, #PLANAR HORN-2SAT reduces to $\#\text{FIXEDPOINTS}_{\text{T}}(\text{D}_2, \text{Forb}_{\preceq}(\emptyset))$. $\square$

Finally, we combine the results to obtain the following conditional dichotomy theorem.

Theorem 8. *Let $\mathcal{F}$ be a Post class of boolean functions and let $\mathcal{G}$ be a graph class closed under taking minors. Under the assumption that $\#\text{FIXEDPOINTS}(\text{D}_2, \text{Forb}_{\preceq}(K_{3,3}, K^5))$ is an intractable problem, the following holds: If ($\mathcal{F} \supseteq \text{V}_2$ or $\mathcal{F} \supseteq \text{E}_2$ or $\mathcal{F} \supseteq \text{D}_2$) and $\mathcal{G} \supseteq$ $\text{Forb}_{\preceq}(K_{3,3}, K^5)$, then $\#\text{FIXEDPOINTS}_{\text{T}}(\mathcal{F}, \mathcal{G})$ is intractable, otherwise $\#\text{FIXEDPOINTS}_{\text{T}}(\mathcal{F}, \mathcal{G})$ is tractable.*

Proof. If ($\mathcal{F} \supseteq \text{V}_2$ or $\mathcal{F} \supseteq \text{E}_2$ or $\mathcal{F} \supseteq \text{D}_2$) and $\mathcal{G} \supseteq \text{Forb}_{\preceq}(K_{3,3}, K^5)$, then $\#\text{FIXEDPOINTS}_{\text{T}}$ $(\mathcal{F}, \mathcal{G})$ is #P-complete by Lemma 4, Lemma 5, and by the assumption made for D_2. Suppose the premise is not satisfied. First, assume that $\mathcal{F} \not\subseteq \mathrm{V}_2$, $\mathcal{F} \not\subseteq \mathrm{E}_2$, and $\mathcal{F} \not\subseteq \mathrm{D}_2$. The maximal Post class having this property is L. By Lemma 1, $\#\textsc{FixedPoints}_\mathrm{T}(\mathrm{L}, \mathrm{Forb}_\preceq(\emptyset))$ is tractable. It remains to consider the case $\mathcal{G} \not\supseteq \mathrm{Forb}_\preceq(K_{3,3}, K^5)$. That is $\mathcal{G} \subseteq \mathrm{Forb}_\preceq(X)$ for some planar graph X. Lemma 2 shows that $\#\textsc{FixedPoints}_\mathrm{T}(\mathrm{BF}, \mathcal{G})$ is solvable in polynomial time. □

4.2 Succinctly Represented Local Transition Functions

In this section we prove a dichotomy theorem for the fixed-point counting problem when transition are given by formulas or circuits. As usual, the size of formula is the number of symbols from the basis used to encode the formula, the size of a circuit is the number of gates (from the basis) it consists of (including the input gates). Both succinct representations of functions lead to the same result. We only prove special results for the case of formula representations. The corresponding results for circuit representations follow easily.

Again we start with gathering the tractable cases.

Lemma 9. $\#\textsc{FixedPoints}_\mathrm{F}(\mathrm{L}, \mathrm{Forb}_\preceq(\emptyset))$ *is solvable in polynomial time.*

Proof. Similar to the proof of Lemma 1 by noting that each boolean circuit C over the base $\{\oplus, 1, 0\}$ can be easily transformed (in polynomial time in the number of gates of C) into the described system of linear equations over Z_2. □

Lemma 10. *Let X be a planar graph. Then,* $\#\textsc{FixedPoints}_\mathrm{F}(\mathrm{E}, \mathrm{Forb}_\preceq(X))$ *is solvable in polynomial time.*

Proof. Since X is planar, there exists a $k \in \mathbb{N}$ such that for all $G \in \mathrm{Forb}_\preceq(X)$, the treewidth of G is at most k. Let $S = (G, \{f_1, \ldots, f_n\})$ be a dynamical system such that $G = (V, E) \in \mathrm{Forb}_\preceq(X)$ and for all $i \in V$, the local transition function f_i is one of the constant functions c_0 or c_1, or is represented by a formula $H_i = \bigwedge_{j \in J_i} x_j$, where $J_i \subseteq N_G(i) \cup \{i\}$. Without loss of generality, we may assume that there is no $i \in V$ such that $f_i \equiv c_1$ or $f_i \equiv c_0$. (Otherwise, an obvious procedure exists to eliminate such vertices.) We define the directed graph $A(S)$ to consist of S's vertex set V and the edge set $E' =_{\mathrm{def}} \{\ (i,j) \mid i,j \in V, i \in J_j\ \}$. Note that $A(S)$ is allowed to have loops. Observe that for all vertices $i, j \in V$ and all fixed-point configurations $\vec{x}$ it holds that if $x_i = 0$ then $x_j = 0$. An easy consequence is that if $C = \{i_1, \ldots, i_r\}$ is a strongly connected component of $A(S)$ and $\vec{x}$ is a fixed-point configuration, then $x_{i_1} = \cdots = x_{i_r}$. Let $\{C_1, \ldots, C_\ell\}$ be the set of all strongly connected components of $A(S)$. Then, the number of fixed-point configurations of S is equal to the number of satisfying assignments of the constraint satisfaction problem $\mathrm{CSP}(S) = (W, \mathcal{D}, \mathcal{C})$ defined by $W =_{\mathrm{def}} \{x_1, \ldots, x_\ell\}$, $\mathcal{D} =_{\mathrm{def}} \{0, 1\}$, and $\mathcal{C} =_{\mathrm{def}} \{\ Ex_ix_j \mid$ there are $u \in C_i$ and $v \in C_j$ such that $(u, v) \in E'\ \}$ where for all i, j such that $Ex_ix_j \in \mathcal{C}$, $E_{ij} =_{\mathrm{def}} \{\ (0,0), (1,0), (1,1)\ \}$[1]. Note that the constraint graph of $\mathrm{CSP}(S)$ (up to being oriented) is a minor of the network of S. It follows that the constraint graph has treewidth at most k. Hence, using the algorithms in [14], the number of fixed-point configurations can be

[1]A constraint satisfaction problem (CSP) consists of triples $(X, \mathcal{D}, \mathcal{C})$, where $X = \{x_1, \ldots, x_n\}$ is the set of variables, $\mathcal{D}$ is the domain of the variables, $\mathcal{C}$ is a set of constraints $Rx_{i_1}, \ldots, x_{i_k}$ having associated corresponding relations $R_{i_1,\ldots,i_k}$. The set $\mathcal{C}$ of constraints is listed by pairs $\langle Rx_{i_1}, \ldots, x_{i_k}, R_{i_1,\ldots,i_k}\rangle$. A solution for $(X, \mathcal{D}, \mathcal{C})$ is an assignment $I : X \to \mathcal{D}$ such that $(I(x_{i_1}), \ldots, I(x_{i_k})) \in R_{i_1,\ldots,i_k}$ for all constraints $Rx_{i_1}, \ldots, x_{i_k} \in \mathcal{C}$. The (primal) constraint graph for $(X, \mathcal{D}, \mathcal{C})$ consists of the vertex set X and the edge set $\{\{x_i, x_j\} \mid x_i$ and x_j occur in the same constraint of $\mathcal{C}\}$.

computed in polynomial time. Consequently, #FIXEDPOINTS$_F$(E, Forb$_{\preceq}(X)$) can be solved in polynomial time. □

Lemma 11. *Let X be a planar graph. Then,* #FIXEDPOINTS$_F$(V, Forb$_{\preceq}(X)$) *is solvable in polynomial time.*

Proof. The case of V is dual to the case of E. Indeed, suppose we have a dynamical system $S = (G, \{f_1, \ldots, f_n\})$ such that $G = (V, E) \in \mathrm{Forb}_{\preceq}(X)$ and for all $i \in V$, f_i is constant or represented by a formula $H_i = \bigvee_{j \in J_i} x_j$ where $J_i \subseteq N_G(i) \cup \{i\}$. Replace each $\vee$ by $\wedge$, 0 by 1, and 1 by 0. Obviously, this gives a dynamical system having the same number of fixed-point configurations as S. Thus, #FIXEDPOINTS$_F$(V, Forb$_{\preceq}(X)$) reduces to #FIXEDPOINTS$_F$(E, Forb$_{\preceq}(X)$). Hence, by Lemma 10, #FIXEDPOINTS$_F$(V, Forb$_{\preceq}(X)$) can be solved in polynomial time. □

Lemma 12. *Let X be a graph with a vertex cover of size one. Then,* #FIXEDPOINTS$_F$ (BF, Forb$_{\preceq}(X)$) *is solvable in polynomial time.*

Proof. Let X have a vertex cover of size one, i.e., Forb$_{\preceq}(X)$ has bounded degree. So, it is easily seen that for all classes $\mathcal{F}$ of boolean functions, #FIXEDPOINTS$_F$($\mathcal{F}$, Forb$_{\preceq}(X)$) reduces to #FIXEDPOINTS$_T$($\mathcal{F}$, Forb$_{\preceq}(X)$). As X is also a planar graph (note that $K^3 \oplus K^2 \preceq K_{3,3}$ and $K^3 \oplus K^2 \preceq K^5$), #FIXEDPOINTS$_F$(BF, Forb$_{\preceq}(X)$) is solvable in polynomial time using Lemma 2. □

We turn to the #P-complete cases.

Lemma 13. #FIXEDPOINTS$_F$(E_2, Forb$_{\preceq}(K_{3,3}, K^5)$) *is* #P*-complete.*

Proof. An inspection of the proof of Lemma 4 shows that the local transition functions specified there are in fact, represented by formulas. Thus, the proposition follows from the proof of Lemma 4. □

Lemma 14. #FIXEDPOINTS$_F$(V_2, Forb$_{\preceq}(K_{3,3}, K^5)$) *is* #P*-complete.*

Proof. Similar to Lemma 13 by inspecting the proof of Lemma 5. □

Let H be a 2CNF formula such that each clause consists of positive literals only. H is called a positive 2CNF. It is well known that the counting problem #POS 2SAT , i.e., counting the satisfying assignments of positive 2CNF, is #P-complete [31].

Lemma 15. #FIXEDPOINTS$_F$(S_{10}, Forb$_{\preceq}(K^3 \oplus K^2)$) *is* #P*-complete.*

Proof. We reduce from #POS 2SAT. Let $H = C_1 \wedge \cdots \wedge C_m$ be a positive 2CNF formula having variables $x_1, \ldots, x_n$. Let $\#_+(H)$ denote the number of satisfying assignments of H. Let $S_{10}(x, y, z) =_{\mathrm{def}} (x \wedge (y \vee z))$ denote the only element in the logical basis of S_{10}. Define S_H to be the dynamical system consisting of the network $G = (V, E)$, where $V =_{\mathrm{def}} \{1, \ldots, n, n+1\}$ and $E =_{\mathrm{def}} \{ \{i, n+1\} \mid i \in \{1, \ldots, n\} \}$, and the local transition functions are specified as follows. For $i \in \{1, \ldots, n\}$ set $P_i(x_i, x_{n+1}) =_{\mathrm{def}} S_{10}(x_i, x_i, x_i)$ and let f_i be represented by P_i. For $i = n+1$, we first define auxiliary formulas A_j for $j \in \{1, \ldots, m\}$ by $A_1(x_1, \ldots, x_{n+1}) =_{\mathrm{def}} S_{10}(x_{n+1}, x_{11}, x_{12})$ and for $k > 1$ by $A_k(x_1, \ldots, x_{n+1}) =_{\mathrm{def}} S_{10}(A_{k-1}(x_1, \ldots, x_{n+1}), x_{k1}, x_{k2})$ where $C_k = (x_{k1} \vee x_{k2})$. Finally, set $P_{n+1}(x_1, \ldots, x_{n+1}) =_{\mathrm{def}} A_m(x_1, \ldots, x_{n+1})$ and let f_{n+1} be represented by P_{n+1}. Certainly, S_H is an (S_{10}, Forb$_{\preceq}(K^3 \oplus K^2)$)-system computable in time polynomial in the size of H. Moreover, note that $P_{n+1}(x_1, \ldots, x_{n+1}) \equiv x_{n+1} \wedge \bigwedge_{j=1}^{m} C_j$. It follows that the number of fixed-point configurations of S_H is $\#_+(H) + 2^n$. Hence, #Pos 2SAT reduces to $\#\text{FixedPoints}_\text{F}(\text{S}_{10}, \text{Forb}_{\preceq}(K^3 \oplus K^2))$. □

Lemma 16. $\#\text{FixedPoints}_\text{F}(\text{S}_{00}, \text{Forb}_{\preceq}(K^3 \oplus K^2))$ *is* #P-*complete.*

Proof. Again we reduce from #Pos 2SAT. Let $H = C_1 \wedge \cdots \wedge C_m$ be a positive 2CNF formula having variables $x_1, \ldots, x_n$. Let $\#_+(H)$ denote the number of satisfying assignments of H. Let $S_{00}(x, y, z) =_{\text{def}} (x \vee (y \wedge z))$ denote the only element in the logical basis of S_{00}. We define S_H to be the dynamical system consisting of the network $G = (V, E)$, where $V =_{\text{def}} \{0, 1, \ldots, n, n+1\}$ and $E =_{\text{def}} \{\ \{i, n+1\} \mid i \in \{0, \ldots, n\}\ \}$, and the set of local transition functions specified as follows: for $i \in \{0, \ldots, n\}$, set $P_i(x_i, x_{n+1}) =_{\text{def}} S_{00}(x_i, x_i, x_i)$ and let f_i be the function represented by P_i. For $i = n+1$, i.e., the center of the star G, we first introduce auxiliary formulas $A_{j_1,\ldots,j_k}(x_0, x_1, \ldots, x_n)$ for $k \in \mathbb{N}_+$ and $j_1 < \cdots < j_k$ inductively defined by $A_i(x_0, x_1, \ldots, x_n) =_{\text{def}} S_{00}(x_{i_1}, x_{i_2}, x_{i_2})$, such that $C_i = (x_{i_1} \vee x_{i_2})$, and

$$A_{j_1,\ldots,j_k}(x_0, \ldots, x_n) =_{\text{def}} S_{00}(x_0, A_{j_1,\ldots,j_{\lfloor k/2 \rfloor}}(x_0, \ldots, x_n), A_{j_{\lfloor k/2 \rfloor+1},\ldots,j_k}(x_0, \ldots, x_n)).$$

We finally define $P_{n+1}(x_0, \ldots, x_{n+1}) =_{\text{def}} S_{00}(x_0, A_{1,\ldots,m}(x_0, \ldots, x_n), x_{n+1})$. Clearly, S_H is an $(S_{00}, \text{Forb}_{\preceq}(K^3 \oplus K^2))$-system computable in time polynomial in the size of H. Moreover, as is easily seen by induction it holds that

$$A_{j_1,\ldots,j_k}(c_0, x_1, \ldots, x_n) \equiv \bigwedge_{\ell=1}^{k} C_\ell \quad \text{and} \quad A_{j_1,\ldots,j_k}(c_1, x_1, \ldots, x_n) \equiv c_1.$$

This leads to the following numbers of fixed-point configurations of S_H:

- there are 2^n fixed-point configurations $\vec{x}$ such that $x_0 = 0$ and $x_{n+1} = 0$,
- there are $\#_+(H)$ fixed-point configurations $\vec{x}$ such that $x_0 = 0$ and $x_{n+1} = 1$,
- there is no fixed-point configuration $\vec{x}$ such that $x_0 = 1$ and $x_{n+1} = 0$, and
- there are 2^n fixed-point configurations $\vec{x}$ such that $x_0 = 1$ and $x_{n+1} = 1$.

Hence, the number of fixed-point configurations of S_H is just $\#_+(H) + 2^{n+1}$. Consequently, #Pos 2SAT reduces to $\#\text{FixedPoints}_\text{F}(\text{S}_{00}, \text{Forb}_{\preceq}(K^3 \oplus K^2))$. □

Lemma 17. $\#\text{FixedPoints}_\text{F}(\text{D}_2, \text{Forb}_{\preceq}(K^3 \oplus K^2))$ *is* #P-*complete.*

Proof. We reduce from #Pos 2SAT. We construct the same network as for the case S_{00} in the proof Lemma 16 on a given positive 2CNF formula $H = C_1 \wedge \cdots \wedge C_m$ having variables $x_1, \ldots, x_n$. Let $\#_+(H)$ denote the number of satisfying assignments of H. The local transition functions are specified as follows. Let $D_2(x, y, z) =_{\text{def}} (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$ denote the only element in the logical basis of D_2. For $i \in \{0, \ldots, n\}$ set $P_i(x_i, x_{n+1}) =_{\text{def}} D_2(x_i, x_i, x_i)$ and let f_i be represented by P_i. For $i = n+1$, we again introduce auxiliary $A_{j_1,\ldots,j_k}(x_0, x_1, \ldots, x_n, x_{n+1})$ for $k \in \mathbb{N}_+$ and $j_1 < \cdots < j_k$ inductively defined by $A_i(x_0, \ldots, x_{n+1}) =_{\text{def}} D_2(x_{i_1}, x_{i_2}, x_{n+1})$, such that $C_i = (x_{i_1} \vee x_{i_2})$, and

$$A_{j_1,\ldots,j_k}(x_0, \ldots, x_{n+1}) =_{\text{def}} D_2(A_{j_1,\ldots,j_{\lfloor k/2 \rfloor}}(x_0, \ldots, x_{n+1}), A_{j_{\lfloor k/2 \rfloor+1},\ldots,j_k}(x_0, \ldots, x_{n+1}), x_0).$$

We finally define $P_{n+1} =_{\mathrm{def}} A_{1,\ldots,m}$. Evidently, S_H is a $(\mathrm{D}_2, \mathrm{Forb}_{\preceq}(K^3 \oplus K^2))$-system and can be computed in time polynomial in the size of H. Moreover, by induction over the formula structure of P_{n+1} we easily obtain the following equivalences:

$$\begin{array}{rcl} P_{n+1}(0, x_1, \ldots, x_n, 0) & \equiv & \bigwedge_{i=1}^{n} x_i \\ P_{n+1}(0, x_1, \ldots, x_n, 1) & \equiv & \bigwedge_{i=1}^{m} (x_{i1} \vee x_{i2}) \\ P_{n+1}(1, x_1, \ldots, x_n, 0) & \equiv & \bigvee_{i=1}^{m} (x_{i1} \wedge x_{i2}) \\ P_{n+1}(1, x_1, \ldots, x_n, 1) & \equiv & \bigvee_{i=1}^{n} x_i \end{array}$$

Thus, the number of fixed-point configurations of S_H is exactly $2\#_+(H) + 2^{n+1} - 2$. Hence, #Pos 2SAT reduces to $\#\textsc{FixedPoints}_{\mathrm{F}}(\mathrm{D}_2, \mathrm{Forb}_{\preceq}(K^3 \oplus K^2))$. □

Finally, we combine all results to obtain the following dichotomy theorem.

Theorem 18. *Let $\mathcal{F}$ be a Post class of boolean functions and let $\mathcal{G}$ be a graph class closed under taking minors. Then, $\#\textsc{FixedPoints}_{\mathrm{F}}(\mathcal{F}, \mathcal{G}))$ is intractable if one of the following conditions is satisfied.*

1. $\big(\mathcal{F} \supseteq \mathrm{S}_{00}$ *or* $\mathcal{F} \supseteq \mathrm{S}_{10}$ *or* $\mathcal{F} \supseteq \mathrm{D}_2\big)$ *and* $\mathcal{G} \supseteq \mathrm{Forb}_{\preceq}(K^3 \oplus K^2)$.
2. $\big(\mathcal{F} \supseteq \mathrm{V}_2$ *or* $\mathcal{F} \supseteq \mathrm{E}_2\big)$ *and* $\mathcal{G} \supseteq \mathrm{Forb}_{\preceq}(K_{3,3}, K^5)$.

Otherwise, $\#\textsc{FixedPoints}_{\mathrm{F}}(\mathcal{F}, \mathcal{G})$ is tractable. Moreover, the same classification is true for $\#\textsc{FixedPoints}_{\mathrm{C}}(\mathcal{F}, \mathcal{G})$.

Proof. If for $\mathcal{F}$ and $\mathcal{G}$ the first conditions is satisfied, then the intractability follows from Lemmas 15, 16, and 17. If $\mathcal{G} \not\supseteq \mathrm{Forb}_{\preceq}(K^3 \oplus K^2)$, then, as argued in [18], there is a graph X having a vertex cover of size one such that $\mathcal{G} \in \mathrm{Forb}_{\preceq}(X)$. Lemma 12 shows that $\#\textsc{FixedPoints}_{\mathrm{F}}(\mathrm{BF}, \mathrm{Forb}_{\preceq}(\mathcal{G}))$ is solvable in polynomial time. Assume that $\mathcal{F} \not\supseteq \mathrm{S}_{00}$, $\mathcal{F} \not\supseteq \mathrm{S}_{10}$, and $\mathcal{F} \not\supseteq \mathrm{D}_2$. The maximal Post classes satisfying this are V, E, and L. Thus, we only consider subclasses of these three classes. If $\mathcal{F}$ and $\mathcal{G}$ satisfy the second condition, then the Lemmas 13 and 14 establish the intractability. Suppose the second condition does not hold. The maximal class $\mathcal{F}$ such that $\mathcal{F} \not\supseteq \mathrm{V}_2$ and $\mathcal{F} \not\supseteq \mathrm{E}_2$ is L. Lemma 9 states that for L counting fixed-points can be done in polynomial time. If $\mathcal{G} \not\supseteq \mathrm{Forb}_{\preceq}(K_{3,3}, K^5)$, then we know that $\mathcal{G} \in \mathrm{Forb}_{\preceq}(X)$ for some planar graph X. Lemmas 10 and 11 imply that $\#\textsc{FixedPoints}_{\mathrm{F}}(\mathrm{E}, \mathcal{G})$ and $\#\textsc{FixedPoints}_{\mathrm{F}}(\mathrm{V}, \mathcal{G})$ are solvable in polynomial time. □

5 Conclusion

Fixed points are an important and robust (in the sense that they exist independently of any update schedule) feature of discrete dynamical systems. We presented two dichotomy theorems on the complexity of counting the number of fixed points in such a system. Both results demonstrate that the linear boolean functions are the only function class such that fixed point counting is tractable independent of representations and of degrees of variable dependency.

The main open issue of this paper is resolving the intractability conjecture for fixed point counting in systems with local transition functions from D_2 over planar graphs. More generally, it is tempting to apply our analysis framework (Post classes and forbidden minors) to a precise identification of islands of predictability for more schedule-based behavioral patterns, e.g., gardens of Eden, predecessors, or fixed-point reachability.

References

[1] Y. Bar-Yam. *Dynamics of Complex Systems*. Studies in Nonlinearity. Addison-Wesley Publishing Co., Reading, MA, 2003.

[2] C. L. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. Predecessor and permutation existence problems for sequential dynamical systems. In *Proceedings of the Conference on Discrete Models for Complex Systems (DMCS'03)*, volume AB of *Discrete Mathematics and Theoretical Computer Science Proceedings*, pages 69–80, 2003.

[3] C. L. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. Reachabillity problems for sequential dynamical systems with threshold functions. *Theoretical Computer Science*, 295(1–3):41–64, 2003.

[4] C. L. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. Complexity of reachabillity problems for finite discrete dynamical systems. *Journal of Computer and System Sciences*, 72(7):1317–1345, 2006.

[5] C. L. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, R. E. Stearns, and P. T. Tošić. Gardens of Eden and fixed points in sequential dynamical systems. In *Proceedings of the 1st International Conference on Discrete Models: Combinatorics, Computation and Geometry (DM-CCG'01)*, volume AA of *Discrete Mathematics and Theoretical Computer Science Proceedings*, pages 241–259, 2001.

[6] C. L. Barrett, H. S. Mortveit, and C. M. Reidys. Elements of a theory of computer simulation II: Sequential dynamical systems. *Applied Mathematics and Computation*, 107(2–3):121–136, 2000.

[7] C. L. Barrett, H. S. Mortveit, and C. M. Reidys. Elements of a theory of computer simulation III: Equivalence of SDS. *Applied Mathematics and Computation*, 122(3):325–340, 2001.

[8] C. L. Barrett and C. M. Reidys. Elements of a theory of computer simulation I: Sequential CA over random graphs. *Applied Mathematics and Computation*, 98(2–3):241–259, 1999.

[9] E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory. *ACM SIGACT News*, 34(4):38–52, 2003.

[10] S. R. Buss, C. H. Papadimitriou, and J. N. Tsitsiklis. On the predictability of coupled automata: An allegory about chaos. *Complex Systems*, 5:525–539, 1991.

[11] K. Cattell and M. J. Dinneen. A characterization of graphs with vertex cover up to five. In *Proceeding of the International Workshop on Orders, Algorithms, and Applications (ORDAL'94)*, volume 831 of *Lecture Notes in Computer Science*, pages 86–99. Springer-Verlag, Berlin, 1994.

[12] N. Creignou and M. Hermann. Complexity of generalized satisfiability counting problems. *Information and Computation*, 125(1):1–12, 1996.

[13] R. Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer-Verlag, Berlin, 3rd edition, 2003.

[14] J. Flum and M. Grohe. The parameterized complexity of counting problems. *SIAM Journal on Computing*, 33(4):892–922, 2004.

[15] F. Green. NP-complete problems in cellular automata. *Complex Systems*, 1(3):453–474, 1987.

[16] L. A. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion.* Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2002.

[17] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, 1982.

[18] S. Kosub. Dichotomy results for fixed-point existence problems for boolean dynamical systems. Technical Report TUM-I0701, Fakultät für Informatik, Technische Universität München, January 2007.

[19] N. Linial. Hard enumeration problems in geometry and combinatorics. *SIAM Journal on Algebraic and Discrete Methods*, 7(2):331–335, 1986.

[20] R. Milner. *Communicating and Mobile Systems: the π-Calculus.* Cambridge University Press, Cambridge, 1999.

[21] E. L. Post. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies*, 5:1–122, 1941.

[22] J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4):777–788, 1983.

[23] A. M. Rabinovich. Complexity of equivalence problems for concurrent systems of finite agents. *Information and Computation*, 139(2):111–129, 1997.

[24] N. Robertson and P. D. Seymour. Graph minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, 1986.

[25] N. Robertson and P. D. Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.

[26] K. Sutner. On the computational complexity of finite cellular automata. *Journal of Computer and System Sciences*, 50(1):87–97, 1995.

[27] P. T. Tošić. On complexity of counting fixed point configurations in certain classes of graph automata. *Electronic Colloquium on Computational Complexity*, 12(51), 2005.

[28] P. T. Tošić. On the complexity of counting fixed points and gardens of Eden in sequential dynamical systems on planar bipartite graphs. *International Journal of Foundations of Computer Science*, 17(5):1179–1203, 2006.

[29] P. T. Tošić and G. A. Agha. On computational complexity of counting fixed points in symmetric boolean graph automata. In *Proceedings of the 4th International Conference on Unconventional Computation (UC'05)*, volume 3699 of *Lecture Notes in Computer Science*, pages 191–205. Springer-Verlag, Berlin, 2005.

[30] S. P. Vadhan. The complexity of counting in sparse, regular, and planar graphs. *SIAM Journal on Computing*, 31(2):398–427, 2002.

[31] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[32] J. von Neumann. *Theory of Self-Reproducing Automata*. Arthur W. Burks (ed.). University of Illinois Press, Champaign, IL, 1966.

[33] S. Wolfram. *Cellular Automata and Complexity. Collected Papers*. Addison-Wesley Publishing Co., Reading, MA, 1994.