

Some Remarks Concerning the Algorithmic Analysis of Gene Regulatory Networks

J. Andrés Montoya^{1,2} Carolina Mejía³

*Departamento de Matemáticas
Universidad Nacional de Colombia
Bogotá, Colombia*

Abstract

In this work we study an algorithmic problem related to gene regulatory networks. This problem is the counting of fixed points in boolean networks. We focus our attention on monomial networks, and we prove that the counting of fixed points is $\#P$ complete even in this restricted case.

Keywords: Gene regulatory networks, fixed points, counting, $\#P$ -completeness.

1 Introduction

In this work we study the complexity of an algorithmic problem related to the analysis of boolean networks. Boolean networks have been employed as elementary models of gene regulatory networks. Kauffman proposed a famous conjecture [9] which relates the structure-dynamics of boolean networks and the emergence of order in biosystems. Roughly speaking Kauffman's conjecture states that the number of limit cycles and fixed points that could occur in the dynamics of boolean networks is very small, very much smaller than one could expect. Kauffman identifies fixed points and limit cycles with cell types and cell replication cycles and uses his conjecture to explain the small number of cell types and the very short cell replication times observed in nature [9]. Kauffman's conjecture has been proved to be false [14], but in despite of this it is still widely studied and discussed. Kauffman's conjecture points out the relevance of studying the existence of efficient algorithms computing the number of fixed points in boolean networks. In this work we study

¹ Thanks to VIE-UIS. Part of this work was developed while both authors were at Mathematical Institute of Universidad Industrial de Santander, Bucaramanga, Colombia.

² Email: amontoyaa@googlemail.com

³ Email: caromejia77@gmail.com

the algorithmic hardness of counting fixed points, we study the restriction of this problem to classes of boolean networks whose transition functions are monomials.

Relations to previous work and contributions. This work is related to the works of R. Laubenbacher and his students (see [11]). Laubenbacher observed that any boolean network is a finite dynamical system over a finite vector space and he also observed that the functions whose domain are finite vector space and whose range are the corresponding fields of scalars are all polynomials. Laubenbacher and his students have studied boolean networks whose transition functions are polynomials satisfying some further constraints, so for example Hernandez-Toledo [7] studied linear networks (that is: boolean networks whose transition functions are linear functions) while Colon-Reyes studied boolean networks whose transition functions are monomials [3]. We prove that the counting of fixed points in monomial networks is $\#P$ complete, to this end we prove that the counting of antichains in partial orders is parsimoniously reducible to the former problem.

Outline of the paper. This paper is organized into three sections. In section one we introduce the basics of Boolean networks, we discuss, in some depth, the conjectures of Kauffman and we define the algorithmic problem that we study in this paper: the counting of fixed points in boolean networks. In section two we study the counting of fixed points when restricted to monomial networks, we prove that this restricted problem is $\#P$ -complete, and we prove that some further restrictions are $\#P$ -complete as well, those later results suggest the existence of a phase transition for the tractability of the problem, this phase transition tells us that for boolean networks linearity implies tractability while nonlinearity implies intractability. Finally in section three, which is a long *concluding remarks section*, we propose some directions for future work which are related to a classical counting problem proposed by R. Dedekind in 1897.

2 Boolean networks

Let \mathcal{B} be the set of all boolean functions. A boolean network is a pair (G, ξ) such that $G = ([n], E)$ is a finite digraph and ξ is a function from $V(G)$ to \mathcal{B} . Moreover, the function ξ satisfies the following constraint:

$$\text{Given } v \in V(G) \text{ the arity of } \xi_v \text{ is equal to } |N(v)|$$

From now on we use the symbol ξ_v to denote the function $\xi(v)$. We use the symbol $N(v)$ to denote the neighborhood of node v which is equal to the set of ancestors of v , (given that G is a directed graph).

That is, a boolean network is a finite digraph such that each one of its nodes is a boolean gate, (i.e a computational device that computes a fixed boolean function). The boolean functions computed by the nodes of G are called *the transition functions* of the network. We can think of the network (G, ξ) as it were a finite dynamical system. Suppose that, at instant t , each one of the nodes of G holds a boolean state. Let v be a node of G . The state of v , at time $t + 1$, depends on the states of its ancestors at time t . Recall that the arity of ξ_v is equal to the number of

ancestors of v , moreover we suppose that the set of v -ancestors is linearly ordered, then we can evaluate the function ξ_v in the boolean tuple determined by the states of the ancestors of v at time t . Let v_1, \dots, v_k the ordered list of ancestors of v and let s_1, \dots, s_k be their internal states at time t . Then, we have that the internal state of v , at time $t + 1$, is equal to $\xi_v(s_1, \dots, s_k)$.

Boolean networks have been employed in the modelling of *gene regulatory networks*. Let (G, ξ) be a boolean network. We can think of the nodes of G as gene, and we can think of the edges of G as representing the regulatory relations between them. A *configuration (global state)* of a boolean network is a function $w : V(G) \rightarrow \{0, 1\}$. Configurations of gene regulatory networks are also called *Gene Activity Profiles*. Given w_t , the gene activity profile of the network (G, ξ) , at time t , the set

$$\mathcal{E}_t = \{v \in V(G) : w_t(v) = 1\}$$

represents the set of gene that are expressed at this time instant. Let v be an arbitrary element of $V(G)$, w_{t+1} , the gene activity profile of the network (G, ξ) , at time $t + 1$, is given by the equations

$$w_{t+1}(v) = \xi_v(w_t[v_1], \dots, w_t[v_k])$$

where $N(v)$, the set of ancestors of v , is equal to $\{v_1, \dots, v_k\}$.

The *local* transition functions attached to the nodes of G determine a *global transition function* $\hat{\xi}$ which acts over the global states of (G, ξ) . Let $s = (s_1, \dots, s_n)$ be a global state, $\hat{\xi}(s)$ is equal to $(\xi_1(s_{11}, \dots, s_{1k_1}), \dots, \xi_n(s_{n1}, \dots, s_{nk_n}))$, where given $i \leq n$ the set $\{v_{i1}, \dots, v_{ik_i}\}$ is the set of ancestors of v_i . Given $w_0 \in \{0, 1\}^{V(G)}$ a configuration, it defines a sequence

$$w_0, \hat{\xi}(w_0), \hat{\xi}(\hat{\xi}(w_0)), \hat{\xi}(\hat{\xi}(\hat{\xi}(w_0))) \dots$$

called the *orbit* of w_0 . We use the symbol $\mathcal{O}_G(w_0)$ to denote the orbit of w_0 . Notice that $\mathcal{O}_G(w_0)$ is always finite, it implies that $\mathcal{O}_G(w_0)$ either reaches a fixed point or enters a cycle.

2.1 Kauffman conjectures

Let G be a boolean network and let w be a configuration of G . We know that $\mathcal{O}_G(w)$ either reaches a fixed point or enters a cycle. Fixed points and limit cycles encode the long-term behavior of G . If we think of a boolean network as a regulatory gene network, then the limit cycles of G represent possible *cell types* [9]. Stuart Kauffman has employed the model of boolean networks as a toy model of evolutionary theory which can explain, to some extent, the small number of cell types and the very short cell replication times observed in most organisms [9] (*the emergence of the order and stability we see in nature*). Kauffman conjectured that boolean networks of bounded indegree exhibit some unexpected features: stability in the long term, a small number of fixed points, short limit cycles. More precisely he conjectured that boolean networks of bounded indegree have a polynomial number of attractors.

Nowadays, we know that it is not the case, Samuelsson and Troein [14] constructed classes of boolean networks of bounded indegree but with a superpolynomial number of attractors (fixed points and limit cycles). Samuelsson-Troein results are not the last word concerning Kauffman's conjectures because *we must be able to explain the stable behavior of boolean networks observed in simulations*.

Given G a boolean network, we use the symbol $d(G)$ to denote the maximum indegree achieved by a node in the network. If $d(G) = 1$ the network exhibits a trivial behavior, while if $d(G) = 2$ the network is located *at the edge of chaos* [10]: its dynamics, stable and predictable, can be used to perform complex computations. On the other hand, it is known that Kauffman's conjectures are no longer true for larger values: if $d(G) \geq 3$ then the network can behave chaotically.

It has been observed that some other parameters can affect the stability of the network: if we restrict the set of available boolean functions to the set of *canalizing functions* [9], then the network becomes stable.

Which are the parameters that have some influence on the stability of the network? Are there sharp thresholds for stability? which parameters determine the soundness of Kauffman's conjectures? These are some few instances of the many questions that arise from the analysis of boolean networks.

3 The counting of fixed points

Kauffman conjectures motivates the introduction of the following algorithmic problem:

Problem 3.1 (*#FP, counting fixed points*)

- *Input:* G , where G is a boolean network.
- *Problem:* compute the number of fixed points of G .

Given \mathcal{S} a set of boolean networks we use the symbol $\#FP[\mathcal{S}]$ to denote the restriction of $\#FP$ to the set \mathcal{S} . In this work we study the complexity of some special restrictions of $\#FP$.

Let (G, ξ) be a boolean network, we suppose that $V(G) = [n]$ and we identify the network (G, ξ) with a tuple $(\xi_i)_{i \leq n}$ of *local transition functions*. Furthermore, we can identify the tuple $(\xi_i)_{i \leq n}$ with the *global transition function* $\widehat{\xi}$. The mapping $\widehat{\xi}$ determines *the topology of the network* (the digraph) in the following way:

Let $i \leq n$ and let $\text{supp}(i)$ be the *support* of $\pi_i(\widehat{\xi})$ which is the set

$$\{j \leq n : \exists w \in \{0, 1\}^n (\Psi(\xi, w, j))\}$$

where $\Psi(\xi, w, j)$ is the formula

$$\pi_i(\widehat{\xi})(w_1, \dots, w_j, \dots, w_n) \neq \pi_i(\widehat{\xi})(w_1, \dots, \neg w_j, \dots, w_n)$$

The topology of the network (determined by the function $\widehat{\xi}$) is given by

$$E(G) = \{(j, i) : j \in \text{supp}(i)\}$$

Recall that $\{0, 1\}$ can be endowed with a field structure, we use the symbol $\mathbb{GF}(2)$ to denote this field. Thus, boolean networks are functions between finite dimensional vector spaces over $\mathbb{GF}(2)$. Let \mathbb{F} be a finite field and let $p : \mathbb{F}^n \rightarrow \mathbb{F}$ be a function, function p can be represented as a polynomial

Lemma 3.2 *Given \mathbb{F} a finite field, given $n \geq 1$ and given $p : \mathbb{F}^n \rightarrow \mathbb{F}$, the function p can be represented as a polynomial over \mathbb{F} .*

Proof. Let $p : \mathbb{F}^n \rightarrow \mathbb{F}$ be a function and suppose that $q = |\mathbb{F}|$. Notice that $p(X_1, \dots, X_n)$ is equal to

$$\sum_{(c_1, \dots, c_n) \in \mathbb{F}^n} \left[p(c_1, \dots, c_n) \prod_{i=1}^n \left(1 - (X_i - c_i)^{q-1} \right) \right]$$

Also, the function p can be represented as a polynomial of degree $n(q-1)$. \square

Given a boolean network (G, ξ) , we can think of the global transition function $\widehat{\xi}$ as a vector $(\xi_i)_{i \leq n}$ whose entries are polynomial functions over the field $\mathbb{GF}(2)$. This approach has been employed by Laubenbacher and his students [11], [7], [3].

There are many natural measures of complexity for polynomial functions, two of them are the algebraic degree (*the deg-hierarchy*) and the number of monomials (*the sum-hierarchy*). According to the first measure, the most elementary polynomials are the linear and the affine functions, while according to the second measure the most elementary polynomials are the monomials.

Example 3.3 (*Affine networks*) Let (G, ξ) be a boolean network and suppose that $V(G) = \{1, \dots, n\}$. We say that (G, ξ) is an *affine network* if and only if given $i \leq n$ there exists $c_i \in \{0, 1\}$ such that $\xi_i = X_{i_1} \oplus \dots \oplus X_{i_k} \oplus c_i$, where i_1, \dots, i_k are the ancestors of i . Suppose that (G, ξ) is an affine network, let $M(G, \xi)$ be the boolean matrix $[m_{ij}]_{i, j \leq n}$ defined by

$$m_{ij} = \begin{cases} 1 & \text{if } j \text{ is an ancestor of } i \\ 0, & \text{otherwise} \end{cases}$$

and let $C(G, \xi)$ be the boolean vector $(c_i)_{i \leq n}$. We note that $X \in \{0, 1\}^n$ is a fixed point of (G, ξ) if and only if it is a solution of the linear system

$$X = M(G, \xi) X + C(G, \xi)$$

Then, we have that the number of fixed points of (G, ξ) is equal to the number of solutions of the system. Thus, we can compute in polynomial time the number of fixed points in affine networks. A fine-grained description of the phase space of linear and affine boolean networks can be found in [7] and [12].

The above example shows that the problem $\#FP$ becomes tractable when restricted to boolean networks whose *transition polynomials* belong to the first level of the deg-hierarchy. Let \mathcal{A}_2 be the class of boolean networks whose transition polynomials are degree two polynomials, it is not difficult to prove that the problem $\#FP[\mathcal{A}_2]$ is hard, hence the problem $\#FP$ becomes intractable when restricted to boolean networks whose transition functions belong to the second level of the deg-hierarchy.

3.1 Monomial networks and the sum-hierarchy

Let (G, ξ) be a boolean network with $V(G) = [n]$. We say that it is a monomial network if and only if for all $i \leq n$ we have that ξ_i is a monomial function, that is: given i_1, \dots, i_k the ancestors of i we have that the equation $\xi_i = X_{i_1} \dots X_{i_k}$ holds.

Remark 3.4 Let (G, ξ) be a gene regulatory network, the sites of G represent the gene and the directed edges the regulatory relations between them. If v is an ancestor of w and X_v belongs to the support of ξ_w then gene v exerts an effective influence on gene w . This influence can be either positive or negative. We say that the influence is positive if the expression of gene v implies or facilitates the expression of gene w . If (G, ξ) is a monomial network then all its local transition functions are conjunctions of positive literals, and then all the regulatory relations between its gene are positive. The class of monomial networks is constituted by the boolean networks whose regulatory relations between gene are all positive. Also, the class of monomial networks is a very restricted class of boolean networks but it is a relevant class.

Given $i \geq 1$ we use the symbol \mathcal{M}_i to denote the boolean networks whose transition polynomials (functions) are polynomials constituted by no more than i monomials. Thus, \mathcal{M}_i denotes the class of boolean networks whose transition functions are polynomials that belong to the i^{th} level of the *sum*-hierarchy, and hence \mathcal{M}_1 denotes the class of monomial networks. We prove, in this section, that the problem $\#FP[\mathcal{M}_1]$ is $\#P$ -complete.

Counting problems. Counting problems are ubiquitous in applications and theoretical informatics.

Definition 3.5 A counting problem is a function $f : \Sigma^* \rightarrow \mathbb{N}$, where Σ is a finite alphabet. If $|\Sigma| = 1$ we say that f is a *tally counting problem*. Given f, g two counting problems we say that f is *parsimoniously reducible* to g if and only if there exists a ptime computable function h such that given x , an instance of f , the equation $f(x) = g(h(x))$ holds. Notice that if f is parsimoniously reducible to g and g can be computed in polynomial time then f can be computed in polynomial time as well.

Counting problems could be very much harder than decision problems, a striking example is the counting of antichains in partial orders. Let (P, \leq) be a poset, an antichain is a set $I \subseteq P$ such that for all $u, v \in I$ we have that if $v \neq u$ then $u \not\leq v$ and $v \not\leq u$. Let DA be the problem in NP consisting in detecting the partial

orders that contain at least one antichain. It should be clear that the problem DA is a trivial one: any poset contains at least one antichain (any subset of size 1 is an antichain). The class $\#P$ is the class of counting problems that can be solved employing a *Polynomial Time Counting Turing Machine*. We can associate to DA , as to any problem in NP , a counting problem in $\#P$, let $\#DA$ be the counting problem consisting in computing the number of antichains contained in a given input poset. Interesting enough the problem $\#DA$ is very much harder than the problem DA : the problem $\#DA$ is $\#P$ -complete [1], it implies that $\#DA$ is NP -hard and it also implies that $\#DA$ is hard for the polynomial hierarchy [16].

Now, we prove that $\#FP[\mathcal{M}_1]$ is $\#P$ -complete, to this end we prove that $\#DA$ is parsimoniously reducible to it.

Theorem 3.6 *The problem $\#FP[\mathcal{M}_1]$ is $\#P$ -complete.*

Proof. Given G a digraph, we use the symbol $CC(G)$ to denote the digraph of maximal strongly connected components of G . Notice that $CC(G)$ is a directed acyclic graph, i.e. a partial order.

Let (G, ξ) be a monomial network. Given $x \in 2^{V(G)}$ we set

$$S_0(x) = \{i \in V(G) : x[i] = 0\}$$

$$S_1(x) = \{i \in V(G) : x[i] = 1\}$$

We note that x is a fixed point if and only if $S_0(x)$ is upper closed and $S_1(x)$ is lower closed.

Now we show that the counting of antichains in partial orders is *ptime* reducible to $\#FP[\mathcal{M}_1]$. Let (P, \leq) be a partial order, we can compute in polynomial time a monomial network (G, ξ) such that $CC(G)$ is isomorphic to (P, \leq) (notice that $CC((P, \leq))$ is isomorphic to (P, \leq)). We use the symbol $\#DA(P)$ to denote the number of antichains contained in (P, \leq) and we show that $\#DA(P)$ is equal to the number of fixed points of (G, ξ) .

Let x be a fixed point of (G, ξ) . Recall that the set $S_1(x)$ is lower closed. It means that $S_1(x)$ contains, at least, one maximal strongly connected component of G . The set of maximal components contained in $S_1(x)$, denoted by \mathcal{C}_x , is an antichain of $CC(G)$. Now, we pick an antichain of $CC(G)$, say $\mathcal{C} = \{C_1, \dots, C_k\}$, and we define a fixed point of (G, ξ) , denoted with the symbol $P_{\mathcal{C}}$, in the following way

$$P_{\mathcal{C}}[i] = 1 \text{ if and only if there exists } l \leq k \text{ such that } C_l \rightarrow i$$

where the symbol $C_l \rightarrow i$ denotes that node i is accessible from C_l . Now, we notice that the functions

$$x \mapsto \mathcal{C}_x \text{ and } \mathcal{C} \mapsto P_{\mathcal{C}}$$

are inverse the one of the other. Thus, we have that $\#DA(P)$ is equal to the number of fixed points of (G, ξ) . Therefore, we have that $\#DA$ is *ptime* reducible to $\#FP[\mathcal{M}_1]$ and then we can claim that $\#FP[\mathcal{M}_1]$ is $\#P$ -complete. \square

Also, once we left the linear world the problem $\#FP[\mathcal{M}_1]$ become $\#P$ -complete. Now, we consider a subclass of \mathcal{M}_1 constituted by the monomial networks of maximum degree 2. This new class of networks corresponds to the intersection of \mathcal{A}_2 ,

the second level of the \times -hierarchy, with \mathcal{M}_1 . we use the symbol \mathcal{M}_1^2 to denote this class and we prove that $\#FP[\mathcal{M}_1^2]$ is $\#P$ -complete. This theorem allows us to detect a *gap theorem* for tractability which tell us that *for boolean networks, nonlinearity implies intractability*.

Theorem 3.7 *The problem $\#FP[\mathcal{M}_1^2]$ is $\#P$ -complete.*

Proof. We prove that $\#FP[\mathcal{M}_1]$ is *ptime* reducible to $\#FP[\mathcal{M}_1^2]$. Let (G, ξ) be a monomial network, the number of fixed points of (G, ξ) is equal to the number of antichains of $CC(G)$. We show that we can compute in polynomial time a quadratic-monomial network (G^*, ξ^*) such that $CC(G^*) \simeq CC(G)$. We suppose that $V(G)$ is equal to $\{1, \dots, n\}$, the main idea behind the construction of G^* is the following one: let i be a node of G and let i_1, \dots, i_k be its ancestors, we replace *the star with k rays* determined by i and its k ancestors by a binary tree whose leaf are the nodes i_1, \dots, i_k , whose root is equal to i and which is directed from the leaf to the root. Then, we add edges connecting i and the inner nodes of this binary tree, all those edges are directed from i to the inner nodes of the tree. We repeat this process for each one of the nodes of G . At the end we get a graph G^* such that for all $v \in V(G^*)$ the inequality $\deg_+(v) \leq 2$ holds, (recall that $\deg_+(v)$ denotes the indegree of v).

From now on, given a digraph G , we use the symbol $v \rightarrow_R u$ to indicate that node u is accessible from node v in the digraph R . We note that $CC(G)$ is isomorphic to $CC(G^*)$. It is the case given that:

- (i) If $v, w \in V(G)$ we have that $v \rightarrow_G w$ if and only $v \rightarrow_{G^*} w$.
- (ii) Let $u \neq v$ be two nodes of G and let r be one of the inner nodes of the binary tree attached to u . We have that $u \rightarrow_{G^*} r$, $r \rightarrow_{G^*} u$, $v \not\rightarrow_{G^*} r$ and $r \not\rightarrow_{G^*} v$.

Given $v \in V(G^*)$ and given w_1, \dots, w_{t_v} ($t_v \leq 2$) its ancestors, we set $\xi_v^* = \prod_{s \leq t_v} X_s$.

Let $\xi^* = (\xi_v^*)$, we have that the number of fixed points of (G^*, ξ^*) is equal to the number of antichains of $CC(G)$ and this later quantity is equal to the number of fixed points of (G, ξ) . Notice that (G^*, ξ^*) can be computed in polynomial time from (G, ξ) . Thus, we have proven that $\#FP[\mathcal{M}_1^2]$ is $\#P$ -complete. \square

The last theorem suggests the existence of a phase transition for tractability: *degree one implies tractability, while degree two implies intractability*. It is not really a phase transition given that we are employing a discrete parameter (the degree of the polynomials governing the dynamics). We will define a suitable parametrization which allows us to detect a genuine phase transition for tractability, a phase transition telling us that: *nonlinearity implies intractability*.

Definition 3.8 Let G be a digraph, we define *the average degree* of G as

$$\frac{\sum_{v \in V(G)} \max\{1, \deg_+(v)\}}{|V(G)|}$$

We use the symbol $\mathcal{M}_1^{(1+\epsilon)}$ to denote the class of monomial networks whose average degree is bounded above by $1 + \epsilon$, we prove that for all $\epsilon \geq 0$ the problem

$\#FP[\mathcal{M}_1^{(1+\epsilon)}]$ is $\#P$ -complete, it should be clear that $\#FP[\mathcal{M}_1^{(1+0)}]$ can be solved in polynomial time.

Theorem 3.9 *If $\epsilon \geq 0$ the problem $\#FP[\mathcal{M}_1^{(1+\epsilon)}]$ is $\#P$ -complete.*

Proof. Let $\epsilon \geq 0$, we prove that $\#FP[\mathcal{M}_1^2]$ is *ptime* reducible to $\#FP[\mathcal{M}_1^{(1+\epsilon)}]$. Let (G, ξ) be an instance of $\#FP[\mathcal{M}_1^2]$, suppose that $n = |V(G)|$ and let m be the smallest positive integer such that the inequality

$$\frac{2n+m}{n+m} \leq 1 + \epsilon$$

holds. Given m we replace each edge of G , say the edge (u, v) , by a directed path

$$u \rightarrow x_1^{uv} \rightarrow \dots \rightarrow x_m^{uv} \rightarrow v$$

Then, given $i \leq m$ we attach to node i the boolean function $X_{v_{i-1}}$. If $\xi_v = X_{s_1} \dots X_{s_k}$ ($k \leq 2$) we attach to node v the boolean monomial $X_{x_m^{s_1 v}} \dots X_{x_m^{s_k v}}$. Let (G^*, ξ^*) be the boolean network obtained in this way, note that:

- $|V(G^*)| = n + m |E(G)|$
- The average degree of G^* is bounded above by $\frac{2n+m}{n+m} \leq 1 + \epsilon$.
- The number of fixed points of (G^*, ξ^*) is equal to the number of fixed points of (G, ξ) .
- The network (G^*, ξ^*) can be computed in polynomial time from (G, ξ) .

Thus, we have that $\#FP[\mathcal{M}_1^2]$ is *ptime* reducible to $\#FP[\mathcal{M}_1^{(1+\epsilon)}]$. \square

4 Directions for future research: on the approximate counting of fixed points

The counting of antichains is a $\#P$ complete problem [1], and it means that it is very unlikely that there exists a polynomial time algorithm solving it. When one has to cope with a hard counting problem one could look for approximate solutions instead of looking for exact solutions. We proved that $\#FP[\mathcal{M}_1]$ is $\#P$ -complete by showing that the problem $\#DA$ is parsimoniously reducible to it. It is worth to remark that $\#DA$ is parsimoniously reducible to $\#FP[\mathcal{M}_1]$ as well. Then, the problems $\#FP[\mathcal{M}_1]$ and $\#DA$ are polynomial time equivalent. Therefore, the existence of (randomized) polynomial time approximation algorithms for $\#DA$ implies the existence of efficient (randomized) approximation algorithms for $\#FP[\mathcal{M}_1]$. We finish this paper discussing some facts related to the approximability of the problems $\#DA$ and $\#FP[\mathcal{M}_1]$.

Inapproximability: counting independent sets. Given G a graph, an *independent set* is a set $I \subseteq V(G)$ such that for all $v, w \in I$ if $v \neq w$ then $\{v, w\} \notin E(G)$. Let (P, \leq) be a partial order, i.e. an acyclic digraph, and let G_P be its *relation graph* defined by:

$V(G_P) = P$ and given $x, y \in P$ we have that $\{x, y\} \in E(\overline{P})$ if and only if either $x \leq y$ or $y \leq x$.

Note that I is an antichain of (P, \leq) if and only if I is an independent set of G_P . Thus, the counting of antichains is closely related to the counting of independent sets in graphs. We use the symbol $\#IS$ to denote the later problem. Problem $\#IS$ is one of the most studied counting problems. It can be proved that $\#IS$ is hard to approximate [5]. The core of the proof consists in proving that if $\#IS$ can be approximated in random polynomial time then *The maximum independent set problem* ($\max IS$, for short) belongs to RP , the probabilistic version of P . If $\max IS \in RP$ then $NP = RP$, given that $\max IS$ is NP complete. Thus, it is very unlikely that $\#IS$ could be probabilistic approximated, and then we claim that $\#IS$ is hard to approximate. Is $\#DA$ hard to approximate? We can prove that if $\#DA$ can be approximated in random polynomial time then the problem $\max AC$ (*the maximum antichain problem*) belongs to RP . Unfortunately (fortunately?) it does not imply that $\#DA$ cannot be approximated given that $\max AC$ can be solved in time $O(n^{2.5})$ [6].

Sampling from distributive lattices. The existence of a *fully polynomial time randomized approximation scheme* for $\#DA$ (a *FPRAS*, for short) have many practical consequences. It is well known that a *FPRAS* yields a *polynomial time sampler* [8]. It would be useful to count with an efficient algorithm which, on input (G, ξ) , generates almost uniformly at random fixed points of the input network. Recall that fixed points represent cell types, then a polynomial time sampler of fixed points would allow us to access generic cell types, which in turn would allow us to compute in polynomial time some relevant cell statistics.

Given a partial order (P, \leq) an *ideal* of (P, \leq) is a downward closed subset of P . Let $J(P)$ be the set of ideals of (P, \leq) , it is easy to check that $(J(P), \cup, \cap)$ is a distributive lattice. It is known that given a finite distributive lattice K there exists a finite poset P_K such that $K = J(P_K)$. On the other hand, there exists a ptime computable bijection between the set of antichains of (P, \leq) and the elements of $J(P)$: the set of maximal elements of I (denoted with the symbol m_I) is an antichain and the function $I \mapsto m_I$ is a polynomial time bijection. Notice that we used this bijection in the proof of theorem 3.6. Thus, approximate counting of fixed points in monomial networks is equivalent to approximate sampling in distributive lattices. ¿There does exist an efficient algorithm for sampling from distributive lattices?

Interesting restrictions: Dedekind's problem. Let $(D_i)_{i \leq n}$ be the sequence defined by: for all $n \geq 1$ the number $D(n)$ is equal to the number of antichains included in \mathcal{B}_n , where \mathcal{B}_n is the boolean algebra of size 2^n . The numbers $D(1), D(2), \dots$ are called *Dedekind numbers*. The history of $\#DA$ goes back to Dedekind [4] who studied the tally counting problem consisting in computing the sequence of Dedekind numbers. *Dedekind's problem* is a strong restriction of the problem $\#DA$, nevertheless it remains unsolved after more than 115 years: we don't know if there exists a closed formula for the n^{th} Dedekind number (although many depth asymptotic results are known, see for example [15]) and we don't know

of the existence of efficient algorithms solving this tally counting problem. As far as we know only the first eight Dedekind numbers have been computed, the first four 3, 6, 20, 168 were computed by Dedekind [4], R. Church computed the next three numbers which are equal to 7581, 7828354 and 2414682040998 [2], and only till 1991 D. Wiedemann succeed in computing the eighth Dedekind number [18] which is equal to 56130437228687557907788. Also, it seems that Dedekind's problem is a very hard tally counting problem. We think that Dedekind's problem is a subproblem of $\#DA$ that has all the main issues. We propose, as a test problem, studying the approximability of Dedekind's function.

References

- [1] M. Ball, J. Provan. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM journal of computing* 12: 777-788, 1983.
- [2] R. Church. Enumeration by rank of the elements of the free distributive lattice with seven generators. *Not. Amer. Math. Soc.*, 1965,12,724.
- [3] O. Colon-Reyes. A control theory for boolean monomial dynamical systems. *Discrete event systems* 20(1):19-35, 2010.
- [4] R. Dedekind. Ueber Zerlegungen von Zahlen durch ihre grossten gemeinsamen Teiler. In *Fest. Hoch. Braunschweig u. ges. Werke II*, pp. 103-148, 1897
- [5] M. Dyer, A. Frieze, M. Jerrum. On counting independent sets in sparse graphs. *Proceedings of FOCS*: 210-217, 1999.
- [6] S. Felsner, V. Raghavan, J. Spinrad. Recognition algorithms for orders of small width and graphs of small Dilworth number. *Order* 20(4):351-364, 2003.
- [7] R. Hernandez-Toledo. Linear finite dynamical systems. *Communications in Algebra* 33: 678-691, 2005.
- [8] M. Jerrum, V. Vazirani, L. Valiant. Random generation of combinatorial structures from an uniform distribution. *Theoretical Computer Science* 43:169-188, 1986.
- [9] S. Kauffman. *The origins of order*. Oxford University Press, 1993.
- [10] C. Langton. Computation at the edge of chaos: phase transitions and emergent computation. *Physica D*, 42: 12-37, 1990.
- [11] R. Laubenbacher, B. Stigler. A computational algebra approach to the reverse-engineering problem of gene regulatory networks. *Journal of theoretical Biology* 229: 523-538, 2004.
- [12] D. Milligan, M. Wilson. The behavior of affine boolean sequential networks. *Connection Science* 5(2):153-167, 1993.
- [13] C. Papadimitriou. *Computational Complexity*. Addison Wesley, Reading MA, 1994.
- [14] B. Samuelsson, C. Troein. Superpolynomial growth in the number of attractors in Kauffman networks. *Physical Review Letters* 90, 098701, 2003.
- [15] A. Sapozhenko. The number of antichains in multilayered ranked sets (Russian). *Diskret. Mat.* 1:110-128, 1989.
- [16] S. Toda. PP is as hard as the polynomial hierarchy. *SIAM Journal on Computing* 20(5):865-877, 1991.
- [17] L. Valiant. The complexity of enumeration and reliability enumeration problems. *SIAM Journal on Computing* 8(3):410-421, 1979.
- [18] D. Wiedemann. A computation of the eighth Dedekind number. *Order* 8:5-6, 1991.