Theoretical Computer Science 412 (2011) 2468–2485

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

A computational proof of complexity of some restricted counting problems

Jin-Yi Cai[a,*], Pinyan Lu[b], Mingji Xia[c]

[a] Computer Sciences Department, University of Wisconsin-Madison, Madison, WI 53706, USA

[b] Microsoft Research Asia, Beijing, 100190, PR China

[c] State Key Laboratory of Computer Science, Institute of Software, CAS, Beijing, 100190, PR China

ARTICLE INFO

Keywords:
Holant problem
Holographic reduction

ABSTRACT

We explore a computational approach to proving the *intractability* of certain counting problems. These problems can be described in various ways, and they include concrete problems such as counting the number of vertex covers or independent sets for 3-regular graphs. The high level principle of our approach is algebraic, which provides sufficient conditions for *interpolation* to succeed. Another algebraic component is *holographic reductions*. We then analyze in detail polynomial maps on $\mathbb{R}^2$ induced by some combinatorial constructions. These maps define sufficiently complicated *dynamics* of $\mathbb{R}^2$ that we can only analyze them computationally. In this paper we use both numerical computation (as intuitive guidance) and symbolic computation (as proof theoretic verification) to derive that a certain collection of combinatorial constructions, in myriad combinations, fulfills the algebraic requirements of proving #P-hardness. The final result is a dichotomy theorem for a class of counting problems. This includes a class of generic *holant problems* with an arbitrary real valued edge signature over (2, 3)-regular undirected graphs. In particular, it includes all *partition functions* with 0–1 vertex assignments and an arbitrary real valued edge function over all 3-regular undirected graphs.

1. Introduction

In this paper we study some counting problems which can be described in the following way. We are given a graph $G = (V, E)$. At each vertex $v \in V$ there is a function f_v, and at each edge $e \in E$ there is a function g_e. We also call these functions f_v and g_e *signatures*. These functions take 0–1 inputs and output real values in $\mathbb{R}$. Now consider all 0–1 assignments σ at each *end* of every edge $e = (x, y)$, i.e., a value $\sigma(e, x)$ and $\sigma(e, y)$. The *counting problem* is to compute $\sum_\sigma \prod_v f_v(\sigma \mid_v) \prod_e g_e(\sigma \mid_e)$, where the sum is over all 0–1 assignments σ, of products of function evaluations over all $v \in V$ and $e \in E$. Here $\sigma \mid_v$ denotes the values assigned locally by σ at v, i.e., the ends of all edges incident to v, and $\sigma \mid_e$ denotes the values assigned by σ at the two ends of e. If each f_v is the Equality function (of arity $= \deg(v)$), then σ can be thought of as 0–1 assignments over the vertex set V. Similarly if each g_e is the Equality function (of arity two), then σ can be taken as 0–1 assignments over E. In this paper we will actually only consider the *standard* form of Holant problems where each edge function g_e is the Equality function. This is without loss of generality, as will be explained in Section 2. This will be our default form of Holant problems (see (1)).

[*] Corresponding author.
E-mail addresses: jyc@cs.wisc.edu (J.-Y. Cai), pinyanl@microsoft.com (P. Lu), mingji@ios.ac.cn (M. Xia).

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.10.039

For example, choosing EQUALITY for every edge, the problems of counting Matchings or Perfect Matchings correspond to taking the AT-MOST-ONE or EXACT-ONE function at each vertex, respectively. Similarly counting all vertex covers on a graph $G = (V, E)$ corresponds to choosing each f_v to be the EQUALITY function of arity $\deg(v)$, and g_e the OR function on two inputs. For 3-regular graphs, each f_v is the EQUALITY function of arity three. Yet another example is Independent Set, which corresponds to EQUALITY for f_v and AT-MOST-ONE for g_e. This framework of counting problems is called *Holant Problems* [9,11], and in general the assignments σ can take values in any finite set $[q]$. Classically, when f_v is fixed to be EQUALITY, and each edge is given the same Boolean function (σ takes values in $[q]$, the edge function maps $[q] \times [q]$ to $\{0, 1\}$), this problem is known as the graph homomorphism problem (or H-colorings or H-homomorphisms, or partition functions) [17–20]. Here H is a fixed directed or undirected graph (with possible self loops) given by a $q \times q$ Boolean adjacency matrix. A mapping $\sigma : V(G) \to V(H)$ is a homomorphism iff for every edge $(x, y) \in E(G)$, $H(\sigma(x), \sigma(y)) = 1$. Then the quantity $\sum_{\sigma} \prod_{(x,y)\in E(G)} H(\sigma(x), \sigma(y))$ counts the number of H-homomorphisms. Vertex cover is the special case where H is the two-vertex graph $(\{0, 1\}, \{(0, 1), (1, 0), (1, 1)\})$. Dichotomy theorems (i.e., the problem is either in P or #P-hard, depending on H) for H-homomorphisms with undirected graphs H and directed acyclic graphs H are given in [18,17] respectively. H-homomorphisms can also be studied for more general functions than Boolean valued functions. A dichotomy theorem for any symmetric matrix H with non-negative real entries is proved in [5]. Very recently Goldberg et al. in a most impressive 73-page paper [19] have proved a dichotomy theorem for any real symmetric matrix H. We will make use of these results [5,19].[1]

Another related incarnation of these problems is known as Constraint Satisfaction Problems (CSP) [4,14–16]. In a Boolean CSP, there is a set of Boolean variables represented by vertices U on the left-hand side (LHS) of a bipartite graph (U, W, E). The right-hand side (RHS) vertices W represent constraint functions. This is equivalent to a Holant problem where it is implicitly assumed that each vertex in U is labeled by an EQUALITY function and each vertex in W is labeled by a constraint function. Thus EQUALITY of arbitrary arity is implicitly assumed to exist in input instances for CSP. If each vertex $w \in W$ is of degree 2 and is assigned the same function, then effectively we can treat w as "an edge" (by merging the two edges incident to w), and we return to the setting of H-homomorphisms. Furthermore if each $u \in U$ has degree 3 then this is effectively a 3-regular graph. We call a bipartite graph (U, W, E) 2–3 regular if $\deg(u) = 3$ and $\deg(w) = 2$ for $u \in U$ and $w \in W$. As indicated, this encompasses 3-regular graphs. It turns out that if EQUALITY gates of arbitrary arity are freely available in possible input graphs then it is technically easier to prove #P-hardness. For holant problems the EQUALITY gates are not freely available unless explicitly given, proofs of #P-hardness become more challenging, because we are more constricted in the design of gadgets in possible reductions. Furthermore there are indeed cases within this class of counting problems where the problem is #P-hard for general graphs, but solvable in P when restricted to 3-regular (or 2–3 regular) graphs.

In this paper we consider mainly 2–3 regular graphs (U, W, E) where each $u \in U$ is assigned the EQUALITY function (of arity 3) and each $w \in W$ is assigned a real symmetric function on two bits. This is the same as holant problems with 0–1 vertex assignments over 3-regular undirected graphs with an arbitrary real valued edge signature. We denote a symmetric function on n bits as $[f_0, f_1, \ldots, f_n]$ where f_i is the value of the function on inputs of Hamming weight i. Then our problem can be denoted as $\#[1, 0, 0, 1] \mid [x_0, x_1, x_2]$, where $x_0, x_1, x_2 \in \mathbb{R}$. Our main result in this paper is a complexity dichotomy theorem for this class of problems.

It turns out that studying counting problems in this framework has a close connection with holographic algorithms and reductions [25]. One can transform the general counting problem $\#[y_0, y_1, y_2, y_3] \mid [x_0, x_1, x_2]$ on 2–3 regular graphs for any pair of symmetric functions to either $\#[1, 0, 0, 1] \mid [z_0, z_1, z_2]$ or $\#[1, 1, 0, 0] \mid [z_0, z_1, z_2]$ by holographic reductions [9], except in some degenerate cases, which are trivially tractable. In [9] a dichotomy theorem was shown for all problems in this class where x_i and y_j are 0–1 valued. The two cases $\#[1, 0, 0, 1] \mid [z_0, z_1, z_2]$ or $\#[1, 1, 0, 0] \mid [z_0, z_1, z_2]$ correspond to a certain characteristic polynomial (which has coefficients defined in terms of x_i's and y_j's) having distinct roots or double roots, with the first case being the generic case of distinct roots. Thus our problem $\#[1, 0, 0, 1] \mid [x_0, x_1, x_2]$ for $x_0, x_1, x_2 \in \mathbb{R}$ corresponds to the generic case with two distinct real characteristic roots. By holographic reductions our dichotomy theorem for $\#[1, 0, 0, 1] \mid [x_0, x_1, x_2]$ has extensions to more general forms. The framework of holant problems was formally introduced in our previous work; we refer to [9,11] for formal definitions and notations. The problems studied in this paper are a very restricted class, over 2–3 graphs, but we find it the simplest class for which it is still non-trivial to prove a dichotomy theorem. It is also a simple class which includes some interesting combinatorial problems. Compared to CSP problems, generally it is more difficult to prove hardness for holant problems. This is because in holant problems EQUALITY gates are not assumed to be present implicitly (if they are, they can be used in reductions for free). As shown in [11], the 2–3 regular graphs are the most basic and also technically the most difficult cases. The dichotomy theorems of general cases in [11] are reduced to these 2–3 regular ones.

The absence of EQUALITY gates of arbitrary arity in problem specification is a real hindrance to proving #P-hardness. Proofs of previous dichotomy theorems make extensive use of constructions called thickening, stretching and pinning. Unfortunately all these constructions require the availability of EQUALITY gates of arbitrary arity to carry out.

Our approach is to reduce H-homomorphism problems (where vertices take 0–1 values) to our problem. There is a dichotomy theorem for H-homomorphism problems. This amounts to proving the reduction

$$\#\{=_1, =_2, =_3, \ldots, =_k, \ldots\} \mid [x_0, x_1, x_2] \le_T \#[1, 0, 0, 1] \mid [x_0, x_1, x_2],$$

[1] In a forthcoming paper [6,7] a general dichotomy theorem for any complex symmetric matrix H is proved.

where $=_k$ is the Equality gate of arity k (e.g., $=_3$ is the same as $[1, 0, 0, 1]$). We use a set of signatures on one side to mean that any signature from that set can be used for vertices on that side of the bipartite graph. The desired Equality gates $\{=_1, =_2, =_3, \dots, =_k, \dots\}$ will be "produced" by *simulation* in a chain of reductions.

The main effort of this paper is to prove that a suitable collection of combinatorial constructions succeed *in the aggregate*. Our constructions lead to polynomial maps on the plane $\mathbb{R}^2$ which are sufficiently complicated that a traditional analysis seems very difficult. First we formulate some algebraic conditions that are sufficient for *interpolation* to succeed. Then we show that a set of combinatorial constructions succeed iff the intersection of certain *failure sets* is empty. This set can be shown to be a semi-algebraic set, i.e., a set defined by a finite set of polynomials ($F = 0$ or $F \geq 0$).

Theoretically, then, we may appeal to Tarski's theorem [21] on the decidability of semi-algebraic sets to finish the proof (assuming indeed that the constructions succeeded). However, decision algorithms for semi-algebraic sets are of notoriously high complexity (in worst case bound *and* in actual execution [13,2,12,1]).

The real work of this paper is to find a way to give a *computational proof* that some collections of combinatorial constructions *jointly* succeed. We use computation in our investigations in two separate ways. First we use numerical computation (mainly Matlab$^{\text{TM}}$) to guide our choice and pruning of combinatorial designs. Second we use symbolic computation (mainly `CylindricalDe- composition` in Mathematica$^{\text{TM}}$) to produce proofs about semi-algebraic sets. Along the way many "engineering" approaches were needed to coax symbolic computation to terminate and produce a definite result.

2. Definitions and background

In this section, we state the counting framework more formally. A *signature grid* $\Omega = (G, \mathcal{F})$ is a tuple, where $G = (V, E)$ is a graph, and each $v \in V(G)$ is assigned a function $f_v \in \mathcal{F}$, taking its incident edges as inputs. A Boolean assignment σ for every $e \in E$ gives an evaluation $\prod_{v \in V} f_v(\sigma \mid_{E(v)})$, where $E(v)$ denotes the incident edges of v. The counting problem on the instance Ω is to compute[2]

$$\text{Holant}_\Omega = \sum_\sigma \prod_{v \in V} f_v(\sigma \mid_{E(v)}). \tag{1}$$

The functions f_v are also called *signatures*. We can express f_v as a vector indexed by $\{0, 1\}^n$. Our functions are real valued functions. Our results are not sensitive to the exact models of real number computation, as long as ordinary addition and multiplication operations are included (see [22,3]).

In Section 1 we started with the slightly more general setting where signatures are given at both vertices and edges, and we sum over all assignments which assign values at *both ends* of an edge. But this generalization can be easily simulated by just edge assignments in a signature grid as follows: Replace each edge $e = (x, y)$ by a path of length two from x to y, and introduce a new vertex v_e of degree 2 in the middle. The function g_e given at e becomes the function at v_e. The values $\sigma(e, x)$ and $\sigma(e, y)$ assigned at the two ends of e become the values assigned to the two new edges of the path. Then we can consider all assignments for the new edge set. This substitution makes G a bipartite graph, where on one side every vertex (the new ones) has degree 2. By giving Equality to all original vertices the Holant becomes a sum over vertex assignments in the original graph. More generally, for bipartite graphs, giving Equality to all vertices on one side defines a CSP problem.

As discussed in the previous section, many important counting problems can be viewed as computing Holant_Ω for appropriate signature grids, such as counting (perfect) matchings and counting vertex covers. Many counting problems not directly defined in terms of graphs can also be formulated as holant problems, e.g., the #SAT problem.

A signature is called symmetric, if each signature entry only depends on the Hamming weight of the input. We use $[f_0, f_1, \dots, f_n]$ to denote a symmetric signature on n inputs, where f_i is the value on inputs of weight i. The signature $[1, 0, \dots, 0, 1]$, where there are $k - 1 \geq 0$ many 0's, is the Equality function of arity k. Another notation for this is $=_k$.

2.1. $\mathcal{F}$-Gate

Any signature from $\mathcal{F}$ is available at a vertex as part of an input graph. Instead of a single vertex, we can use graph fragments to generalize this notion. An $\mathcal{F}$-gate Γ is a pair $(H, \mathcal{F})$, where $H = (V, E, D)$ is a graph with some dangling edges D. (See Fig. 1 for one example.) Other than these dangling edges, an $\mathcal{F}$-gate is the same as a signature grid. The role of dangling edges is similar to that of external nodes in Valiant's notion [24], however we allow more than one dangling edge for a node. In $H = (V, E, D)$ each node is assigned a function in $\mathcal{F}$ (we do not consider "dangling" leaf nodes at the end of a dangling edge among these), E are the regular edges, denoted as $1, 2, \dots, m$, and D are the dangling edges, denoted as $m + 1, m + 2, \dots, m + n$. Then we can define a function for this $\mathcal{F}$-gate $\Gamma = (H, \mathcal{F})$,

$$\Gamma(y_1, y_2, \dots, y_n) = \sum_{x_1, x_2, \dots, x_m \in \{0,1\}^m} H(x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n),$$

where $(y_1, y_2, \dots, y_n) \in \{0, 1\}^n$ denotes an assignment on the dangling edges and $H(x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n)$ denotes the value of the signature grid on an assignment of all edges, i.e., the product of evaluations at every vertex of H, for

[2] The term Holant was first introduced by Valiant in [25] to denote a related exponential sum.

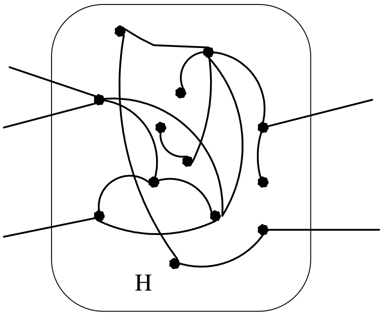


Fig. 1. An $\mathcal{F}$-gate.

$(x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_n) \in \{0, 1\}^{m+n}$. We will also call this function the signature of the $\mathcal{F}$-gate Γ. An $\mathcal{F}$-gate can be used in a signature grid as if it were just a single node with the particular signature. We note that even for a very simple signature set $\mathcal{F}$, the signatures for all $\mathcal{F}$-gates can be quite complicated and expressive. Matchgate signatures are an example [24].

In this paper, we mainly consider bipartite graphs. We use $\#\mathcal{G}|\mathcal{R}$ to denote all the counting problems, expressed as holant problems, on bipartite graphs $H = (U, W, E)$, where each signature for a vertex in U or W is from $\mathcal{G}$ or $\mathcal{R}$, respectively. An input instance of the holant problem is a signature grid and is denoted as $\Omega = (H, \mathcal{G}|\mathcal{R})$. In this bipartite setting, we should also specify, for each dangling edge of an $\mathcal{F}$-gate, which part it is expected to connect to.

2.2. Holographic reduction

To introduce the idea of holographic reductions, we consider bipartite graphs. We note that this is without loss of generality. For any general graph, we can make it bipartite by adding an additional vertex on each edge, and giving the new vertex the Equality function [1, 0, 1].

One can perform linear transformations on the signatures. We will define a simple version of holographic reductions, which is invertible. Suppose $\#\mathcal{G}|\mathcal{R}$ and $\#\mathcal{G}'|\mathcal{R}'$ are two holant problems defined for the same family of graphs, and let $T \in \mathbf{GL}_2$ be an invertible 2×2 matrix. We say that there is a holographic reduction from $\#\mathcal{G}|\mathcal{R}$ to $\#\mathcal{G}'|\mathcal{R}'$, if the *contravariant* transformation $G' = T^{\otimes g}G$ and the *covariant* transformation $R = R'T^{\otimes r}$ map $G \in \mathcal{G}$ to $G' \in \mathcal{G}'$ and $R \in \mathcal{R}$ to $R' \in \mathcal{R}'$ where G and R have arity g and r respectively. (Notice the reversal of directions when the transformation $T^{\otimes n}$ is applied. This is the meaning of *contravariance* and *covariance*. Here G and G' are identified with 2^g-dimensional column vectors. Similarly R and R' are identified with 2^r-dimensional row vectors. $T^{\otimes n}$ is the n-th fold tensor product of T.)

Theorem 1 (*Valiant's Holant Theorem*). *Suppose there is a holographic reduction from $\#\mathcal{G}|\mathcal{R}$ to $\#\mathcal{G}'|\mathcal{R}'$ mapping signature grid Ω to Ω', then* $\text{Holant}_\Omega = \text{Holant}_{\Omega'}$.

In particular, for invertible holographic reductions that are 1–1 and onto from $\#\mathcal{G}|\mathcal{R}$ to $\#\mathcal{G}'|\mathcal{R}'$, one problem is in P iff the other one is, and similarly one problem is #P-hard iff the other one is also.

3. A dichotomy theorem and reduction chain

Our main theorem is the following dichotomy theorem.

Theorem 2. *The counting problem* $\#[1, 0, 0, 1] \mid [x_0, x_1, x_2]$ *is #P-hard unless one of the following conditions holds:* (1) $x_1^2 = x_0x_2$; (2) $x_0 = x_2 = 0$ *or* $x_1 = 0$; (3) $x_0 = x_1 = -x_2$ *or* $x_0 = -x_1 = -x_2$; *the problem* $\#[1, 0, 0, 1] \mid [x_0, x_1, x_2]$ *is polynomial time computable in these three cases.*

We remark that if we restrict ourselves to planar graphs, there is a 4th category of tractable cases $x_0 = x_2$, which can be solved in polynomial time by holographic algorithms with matchgates [25,8]. All cases other than those listed in the 4 categories are #P-hard for planar graphs.

Now we return to general (not necessarily planar) graphs. If $x_1 = 0$, the problem is easily computable in polynomial time, by a connectivity argument. So we consider the case $x_1 \neq 0$, and by a scalar factor, we can assume $x_1 = 1$. Then the general problem can be stated as $\#[1, 0, 0, 1] \mid [a, 1, b]$, and naturally it can be identified with a point (a, b) in the real plane $\mathbb{R}^2$.

We first note that if $ab = 1$ or $(a, b) \in \{(0, 0), (1, -1), (-1, 1)\}$, the more general problem $\#\{=_1, =_2, =_3, \ldots, =_k, \ldots\} \mid [a, 1, b]$ is tractable in P (see [5,19]). It follows easily that our problem $\#[1, 0, 0, 1] \mid [a, 1, b]$ is also tractable in P. We remark that the problem $\#[1, 0, 0, 1] \mid [1, 1, -1]$ has the following natural interpretation. Given a 3-graph G with n vertices, any subset of vertices induces a subgraph with either an odd or an even number of edges. Let X and Y denote the number of induced subgraphs with an odd and even number of edges, respectively. Then the solution to the problem $\#[1, 0, 0, 1] \mid [1, 1, -1]$ on input G is $Y - X$, with $X + Y = 2^n$. Thus the problem computes the value X (and Y).

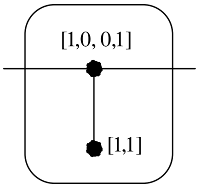


Fig. 2. A small gadget.

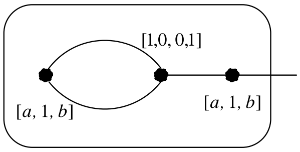


Fig. 3. A gadget for $[1, 1]$ when $a = b$.

Now suppose $(a, b) \in \mathbb{R}^2$ such that $ab \neq 1$ and $(a, b) \notin \{(0, 0), (1, -1), (-1, 1)\}$. It is known that in this case $\#\{=_1, =_2, =_3, \ldots, =_k, \ldots\} \mid [a, 1, b]$ is #P-hard [5,19]. For these (a, b), we show our problem $\#[1, 0, 0, 1] \mid [a, 1, b]$ is #P-hard by giving a chain of reductions,

$$\begin{aligned}\#\{=_1, =_2, =_3, \ldots, =_k, \ldots\} \mid [a, 1, b] \ &\leq_T \#[1, 0, 0, 1] \mid \{[a, 1, b], [1, 0, 1]\} &(2)\\ &\leq_T \#\{[1, 0, 0, 1], [1, 0, 1]\} \mid [a, 1, b] &(3)\\ &\leq_T \#[1, 0, 0, 1] \mid \{[a, 1, b], [1, 1]\} &(4)\\ &\leq_T \#[1, 0, 0, 1] \mid [a, 1, b]. &(5)\end{aligned}$$

The goal of this reduction chain is to “simulate" Equality of arbitrary arity. Step (2) is easy. With $[1, 0, 1]$ on the RHS and $[1, 0, 0, 1]$ on the LHS we can simulate any $=_k$. To prove step (3) we use the following lemma. It shows that if we have $[1, 0, 1]$ on the LHS, we can perform stretching and interpolate $[1, 0, 1]$ on the RHS. It is a special case of Lemma 3.4 in [18] by Dyer and Greenhill:

Lemma 3. *If $ab \neq 1$ and $\mathcal{F}$ is a set of signatures, then*

$$\#\mathcal{F} \mid \{[a, 1, b], [1, 0, 1]\} \leq_T \#\mathcal{F} \cup \{[1, 0, 1]\} \mid [a, 1, b].$$

This proves Step (3). However, compared to our problem $\#[1, 0, 0, 1] \mid [a, 1, b]$ we do not have the signature $[1, 0, 1]$ on the LHS. The way we accomplish this is to realize a unary signature $[1, 1]$ on the RHS. If we have $[1, 1]$ on the RHS, we can realize $[1, 0, 1]$ on the LHS by the simple gadget in Fig. 2, which proves step (4).

Then, the main task is step (5): to realize $[1, 1]$ on the RHS. If $a = b \notin \{0, -1\}$, we can realize $[1, 1]$ by the gadget in Fig. 3. The signature of the $\mathcal{F}$-gate is $[a^2 + a, a^2 + a]$, and we can take the common factor $a^2 + a \neq 0$ out to get $[1, 1]$ given that $a \notin \{0, -1\}$. Note that $a = b = 0$ or $a = b = -1$ fall in the tractable cases in Theorem 2.

If $a \neq b$, we do not know how to realize $[1, 1]$ directly for a generic pair (a, b). However, it turns out that we can interpolate all the unary functions on the RHS. This is our main lemma in this paper.

Lemma 4. *If $ab \neq 1$, $a \neq b$ and $(a, b) \notin \{(1, -1), (-1, 1)\}$, then*

$$\#[1, 0, 0, 1] \mid \{[a, 1, b]\} \cup \mathcal{U} \leq_T \#[1, 0, 0, 1] \mid [a, 1, b],$$

where $\mathcal{U}$ denotes any finite subset of all unary signatures.

The proof of Lemma 4 will show the reduction where $\mathcal{U}$ consists of a single unary signature. By the transitivity of polynomial-time Turing reductions (or Cook reductions $\leq_T$) the reduction holds for any finite set $\mathcal{U}$ of unary signatures.

We remark that when $a = b$, the reduction in Lemma 4 does not hold. So the case $a = b$ must be handled separately as above.

4. Interpolation method

In this section, we discuss the interpolation method we will use for our main lemma. Polynomial interpolation is a powerful tool in the study of counting problems initiated by Valiant and further developed by Vadhan et al. [23,18]. We want to show that for any unary signature $f = [x, y]$, we have $\#[1, 0, 0, 1] \mid \{[a, 1, b], [x, y]\} \leq_T \#[1, 0, 0, 1] \mid [a, 1, b]$, under some conditions on a and b. Let $\Omega = (G, [1, 0, 0, 1] \mid \{[a, 1, b], [x, y]\})$ be a signature grid. Here $[x, y]$ appears on the RHS. We want to compute Holant_Ω in polynomial time using an oracle for $\#[1, 0, 0, 1] \mid [a, 1, b]$.

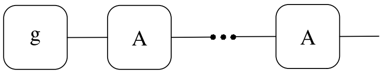


Fig. 4. Recursive construction.

Let V_f be the subset of vertices in G assigned f in Ω. Suppose $|V_f| = n$. These vertices are on the RHS, all with degree 1, and all connected to some vertex on the LHS of degree 3. We can classify all 0–1 assignments σ in the Holant sum according to how many vertices in V_f whose incident edge is assigned a zero or a one. Then the Holant value (1) can be expressed as

$$\text{Holant}_\Omega = \sum_{0\le i\le n} c_i x^i y^{n-i}, \tag{6}$$

where c_i is the sum over all edge assignments σ, of products of evaluations at all $v \in V(G) - V_f$, where σ is such that exactly i vertices in V_f have their incident edges assigned 0 (and $n - i$ have their incident edges assigned 1). If we can evaluate these c_i, we can evaluate Holant_Ω.

Now suppose $\{G_s\}$ is a sequence of $\mathcal{F}$-gates using signature pairs $[1, 0, 0, 1] \mid [a, 1, b]$. Each G_s has one dangling edge which is to be connected externally to a vertex of degree 3. Denote the signature of G_s by $f_s = [x_s, y_s]$, for $s = 0, 1, \ldots$. If we replace each occurrence of f by f_s in Ω we get a new signature grid Ω_s on signature pairs $[1, 0, 0, 1] \mid [a, 1, b]$ with

$$\text{Holant}_{\Omega_s} = \sum_{0\le i\le n} c_i x_s^i y_s^{n-i}. \tag{7}$$

One can evaluate Holant_{Ω_s} by oracle access to $\#[1, 0, 0, 1]\,|\,[a, 1, b]$. Note that the same values c_i appear in the sums Holant_Ω and Holant_{Ω_s}. We can treat c_i in (7) as a set of unknowns in a linear system. The idea of interpolation is to find a suitable sequence $\{f_s\}$ such that the evaluations of Holant_{Ω_s}, for polynomially many s, give a linear system (7) of full rank, from which we can solve for all c_i.

In this paper, the sequence $\{G_s\}$ will be constructed recursively using suitable gadgetry. There are two gadgets in a recursive construction: one gadget has arity 1, giving the initial signature $g = [x_0, y_0]^{\mathrm{T}}$; the other has arity 2, giving the recursive iteration. It is more convenient to use a 2×2 matrix A to denote it. We remark that the dangling edge of the arity 1 gadget is expected to connect externally to a vertex of degree 3; from the gadget of arity 2, one dangling edge is to be connected externally to a vertex of degree 3 and the other to a vertex of degree 2. So we can recursively connect them as in Fig. 4 and get $\{G_s\}$.

The signatures of $\{G_s\}$ have the relation $\begin{bmatrix} x_s \\ y_s \end{bmatrix} = A \begin{bmatrix} x_{s-1} \\ y_{s-1} \end{bmatrix}$, where $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ and $g = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$. In the following, we will call this gadget pair (A, g) the recursive construction. It follows from Lemma 6.1 in [23] that

Lemma 5. *Let α and β be the two eigenvalues of A. If $\det(A) \neq 0$, g is not a column eigenvector of A (nor the zero vector), and α/β is not a root of unity, then the recursive construction (A, g) can be used to interpolate all unary signatures.*

Notice that both A and g are functions of (a, b). Here we relax the condition that $\frac{\alpha}{\beta}$ is not a root of unity to $|\frac{\alpha}{\beta}| \neq 1$ so that all the conditions can be described by polynomial equalities or inequalities of (a, b). We denote by $[Ag, g]$ the 2×2 matrix with first column Ag and second column g. Then $\det[Ag, g] = 0$ is equivalent to g being a column eigenvector of A (or the zero vector). If $\operatorname{tr} A \neq 0$ and the discriminant $(\operatorname{tr} A)^2 - 4\det(A) > 0$, then it is easy to see that the two eigenvalues α and β have unequal norms.

Definition 6. The *failure set* of a recursive construction (A, g) is the following semi-algebraic set $\mathcal{F}(A, g)$:

$$[\det(A) = 0] \text{ or } [\operatorname{tr} A = 0] \text{ or } [(\operatorname{tr} A)^2 - 4\det(A) \le 0] \text{ or } [\det[Ag, g] = 0].$$

Denote by

$$E = \{(a, b) \in \mathbb{R}^2 \mid ab = 1 \text{ or } a = b \text{ or } (a, b) = (1, -1) \text{ or } (a, b) = (-1, 1)\},$$

the exceptional cases of Lemma 4. We prove there are a finite number of gadgets (A_i, g_i), where $i = 1, 2, \ldots, C$, such that

$$\bigcap_i \mathcal{F}(A_i, g_i) \subseteq E. \tag{8}$$

This would imply Lemma 4.

5. Computational proof

We prove Lemma 4 by establishing (8). We will give an account of the many steps taken to overcome various difficulties. Some of the difficulties are not of a logical nature, but a matter of computational complexity, in a practical sense. The

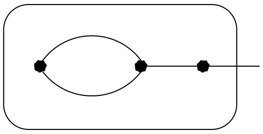


Fig. 5. The initiation component g.

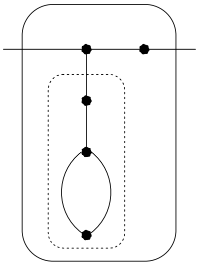


Fig. 6. The iterative component A.

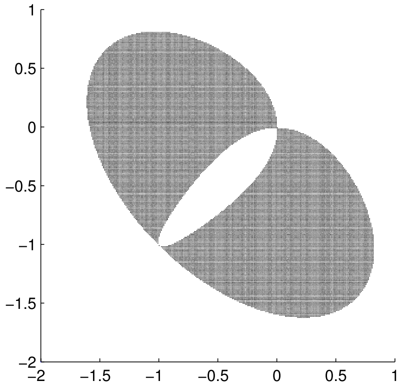


Fig. 7. The failure set of the first gadget.

proof will be a combination of mathematical derivation (in a traditional sense) and an "engineering" undertaking. We find it amusing that we must contend with practical computational complexity in proving theorems of computational complexity.

We will construct some combinatorial gadgets. As described earlier each gadget will depend on two components: an initiation component and an iterative component. We start with the simplest gadgets in Figs. 5 and 6. (In our figures, unless otherwise specified, all vertices of degree 3 and 2 have signatures $[1, 0, 0, 1]$ and $[a, 1, b]$, respectively.)

The gadget in Fig. 5 has a unary signature $[a^2 + b, a + b^2]$, and must be externally connected to a vertex of degree 3. The gadget in Fig. 6 has one dangling edge connecting to a degree 2 vertex and another dangling edge connecting to a degree 3 vertex. The matrix A for the gadget in Fig. 6 is $\begin{bmatrix} a(a^2+b) & a+b^2 \\ a^2+b & b(a+b^2) \end{bmatrix}$.

We consider the failure set $\mathcal{F}(A, g)$ for this gadget. The inequality is our main concern as the equalities define a lower dimensional set. We now focus on the *main failure set*

$$\mathcal{F}^*(A) = \{(a, b) \in \mathbb{R}^2 - E \mid (\mathrm{tr}(A))^2 - 4\det(A) \le 0\}.$$

This set is depicted in Fig. 7. We remark that this main failure set only depends on the iterative gadget A and in the following discussion we focus on this component.

This picture is produced by numerical computation. We will use numerical computation, not as proof, but as intuitive guidance in our search for gadgets. For example, suggested by the numerical computation, after some standard estimation we can prove (traditionally) that this set $\mathcal{F}^*(A)$ is bounded by the square $[-2, 1] \times [-2, 1]$. A consequence of this is that for every $(a, b) \notin [-2, 1] \times [-2, 1]$, except on a lower dimensional set, the problem is already proved to be #P-hard. Another side benefit of this boundedness is that, going forward, all numerical detective work will be restricted to a bounded region, which would have been hard to do without the boundedness.

Our goal, then, is to somehow shrink this failure set, by finding good gadgets. In Fig. 6, the dashed box can be replaced by another unary gadget whose dangling edge is to be connected to a vertex of degree 3. Let this unary gadget have signature $[c, d]$, then the iterative component will have signature $A = \begin{bmatrix} ac & d \\ c & bd \end{bmatrix}$. Thus a natural idea is to design another gadget replacing that part with another unary gadget. Here is another such gadget.

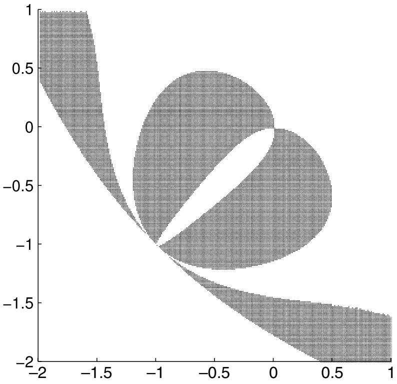
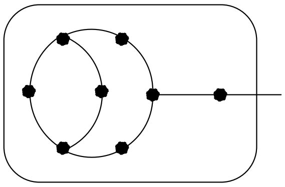


Fig. 8. The failure set using g'.



This gadget has the unary signature $g' = [a^5 + 3a^2 + ab^2 + 2b + b^4, b^5 + 3b^2 + ba^2 + 2a + a^4]$, and can be used instead of g in the dashed box. The result of using g' instead of g is illustrated in Fig. 8.

If we take the numerical computation as trustworthy, then outside of the intersection of Figs. 7 and 8, modulo a lower dimensional set, #P-hardness has already been established. Our hope then is to find enough gadgets such that the intersection becomes empty (as a subset of $\mathbb{R}^2 - E$). We note crucially that the intersection, together with the lower dimensional set, is a semi-algebraic set. Thus it is decidable, by Tarski's theorem [21], whether a particular finite collection of gadgets produces an empty intersection. Thus to prove that for all (a, b) not in the known tractable set the problem is #P-hard, "all we need to do" is to find a sufficient number of gadgets, and apply Tarski's theorem. Of course this plan can only succeed if the statement is actually true, and we can in fact find a finite number of gadgets whose intersection of failure sets is indeed empty. Certainly there is no point in applying Tarski's theorem when numerical computation indicates that the gadgets found so far are manifestly insufficient.

Off we go to hunt for more gadgets. The next idea is to use the iterative construction for a different purpose. We will use the iterative construction now to construct many unary signatures, each to be used inside the dashed box for the original iterative construction. More precisely, if we use g (or g') as the initial unary signature inside the dashed box, and iterate the construction k times, we will obtain k new unary signatures, say, $g_1, g_2, \ldots, g_k$, each of which can be used as the initial signature $[c, d]$ inside the dashed box to start the iterative construction for the purpose of interpolation.

After some numerical computation the evidence is that while the intersection of failure sets gets thinner, these gadgets are still not enough. The next idea is that in our iterative construction for the initial unary signatures, we can use either g or g' in the dashed box interchangeably, per each iteration. Thus, to iterate this process k times, we can produce 2^k initial signatures usable as $[c, d]$, with which to start its own iterative construction for the purpose of interpolation. After some experiment (numerical computation again) we decided that this is not so trivial. And so we started a computer search, with iteration depth $k = 1, 2, \ldots$, and with a random choice of g or g' per each iteration.

Our computation reveals that at depth $k = 15$ certain sequences g and g''s seem to have produced a collection of $[c, d]$'s to start the iterative construction, whose failure sets have an empty intersection. We then hand-pick and prune it down: A particular sequence of 7 copies of g and g' in succession produced a collection, whose failure sets seem to have an empty intersection. All of this is not proved, but strongly suggested by numerical computation.

At this point, it seems that we just need to hand this to Tarski's theorem. But we encountered an unexpected problem. The 7 gadgets produced a well defined semi-algebraic set which is presumably empty and this fact is decidable; however, the emptiness computation simply will not terminate. (In fact it did not terminate even for 6 gadgets.) Thus our problem has turned into a practical misfortune: fewer gadgets are not sufficient (numerical evidence); more gadgets seem to suffice, but they are too complicated to handle analytically and beyond the capability of symbolic computation. We should note that this insistence on proofs beyond numerical evidence is absolutely necessary; we had many instances in this proof where numerically indicated assertions are actually false.

We next assessed what feature of a gadget appears to have the greatest impact on the practical performance of the decision procedure on semi-algebraic sets. And this appears to be the degree of the polynomial, which translates to the number of degree 2 vertices. For 2–3 regular graphs, this is proportional to the number of degree 3 vertices. We call this number m. By a parity argument, taking into account of dangling edges, it can be shown that m must be even.

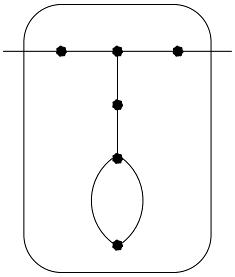


Fig. 9. One edge transformation.

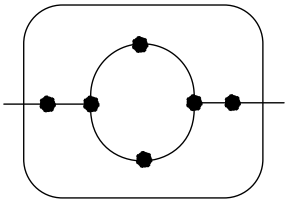


Fig. 10. Another edge transformation.

So we systematically enumerated all gadgets with $m = 2$, 4 or 6. There are over 170 gadgets with $m = 6$. On the other hand, symbolic computation cannot even handle a single gadget with $m = 8$ (the computation does not terminate). Yet, numerically even all gadgets with $m = 6$ together do not suffice.

At this point we decided to modify our strategy. Instead of looking for a set of gadgets which will completely cover all the points in $\mathbb{R}^2 - E$, we will settle for a set of gadgets with relatively low degree signature polynomials ($m \leq 6$, so that symbolic computation can handle) and whose failure set is *small* and *easy to delineate*. We search for such gadgets numerically, and once we settle on such a set of gadgets, we will bound it by a *box* defined by piece-wise linear segments such as a triangle. We use symbolic computation to confirm that the box indeed contains the failure set. Outside such boxes the problem is already proved #P-hard. Then we deal with the boxes separately.

The next idea is somewhat different. We consider an $\mathcal{F}$-gate with two dangling edges, both to be connected externally to vertices of degree 3. In Fig. 9 we depict such an $\mathcal{F}$-gate. Note that such an $\mathcal{F}$-gate can replace a vertex of degree 2 everywhere. It can be verified that its signature is $[a^2(a^2+b)+a+b^2, a(a^2+b)+b(a+b^2), a^2+b+b^2(a+b^2)]$. Logically, if the counting problem with the above transformed signature is #P-hard, then so is the counting problem with $[a, 1, b]$.

The dehomogenized form of the signature map is

$$f_1 : (a, b) \mapsto \left(\frac{a^2(a^2+b)+a+b^2}{a^3+2ab+b^3}, \frac{a^2+b+b^2(a+b^2)}{a^3+2ab+b^3} \right). \tag{9}$$

The subset of (a, b) in a box which is mapped by f_1 to a point within the box is a semi-algebraic set. Because of the relaxation of the original failure set to the box, the defining polynomials of this semi-algebraic set are of relatively low degree. Had we not enlarged the failure set to the box, this degree would have been too high for symbolic computation.

In fact we will need another $\mathcal{F}$-gate (Fig. 10), where it has the dehomogenized form of transformation

$$f_2 : (a, b) \mapsto \left(\frac{a(a^3+1)+a+b^2}{a^3+ab+1+b^3}, \frac{a^2+b+b(1+b^3)}{a^3+ab+1+b^3} \right). \tag{10}$$

Note that at least one map of f_1 and f_2 is well defined at every $(a, b) \in \mathbb{R}^2 - E$, since $a^3 + 2ab + b^3$ and $a^3 + ab + 1 + b^3$ simultaneously vanish only at $ab = 1$.

There is another idea that we use to lower the degree, which makes symbolic computation feasible. It can be seen that the various signatures are symmetric functions of a and b. Once we make a change of coordinates, such that the y-axis is the line $a = b$, then the functions all become even functions of x. Also it appears that the point $(a, b) = (-1, -1)$ is where most of the trouble resides (at least numerically). Thus we use the transformation $a = -\sqrt{x}+y-1$, $b = \sqrt{x}+y-1$, and we still get signatures which are polynomial functions of x and y. This transformation further lowers the degree. In fact from now on we operate in the xy plane, forming our boxes there. We also note that on the xy plane we only need to consider $x > 0$. (Note that $x = 0$ corresponds to the line $a = b$ and is excluded in $\mathbb{R}^2 - E$.)

It turns out that we can find seven gadgets (one with $m = 2$, one with $m = 4$ and five with $m = 6$) such that the combined failure set can be proved by symbolic computation to be the union of: (1) a small region bounded by a small box (2) a small number of curves defined by polynomial equations (each with several branches), and (3) a small number of isolated points.

It is somewhat awkward to bound the curves in part (2) individually. However we can prove by symbolic computation that it is contained in the box $0 < x < 0.14$ and $0 \leq y \leq 1$.

Let B be the union of two boxes from part (1) and (2). It can be proved by symbolic computation that both f_1 and f_2 are well-defined within the box for part (1) and for any point of part (2) belonging to the curves. Every point (x, y) not in B and not in part (3) has its corresponding $[a, 1, b]$ already proved #P-hard. We use (9) and (10) (and in one case, a third rational transformation corresponding to another $\mathcal{F}$-gate) to prove that for all points in part (3), after a finite number of transformations they all fall outside of the union of three parts.

For points in B, we apply the rational transformations (9) and (10), again in combination and in iteration. Each iteration is as follows. Starting with B, in one iteration we define $\tilde{B} = \{(x, y) \in B \mid f_1(x, y) \in B \text{ and } f_2(x, y) \in B\}$. $\tilde{B}$ is a smaller subset of B. We then bound $\tilde{B}$ by a slightly larger new "box" B'. B' is still smaller than B, and this re-bounding is necessary since our symbolic computation can only handle polynomials of degrees with an absolute upper bound. Then we replace B with B', and iterate.

After several iterations the "box" becomes a very thin strip, but it does not vanish. Our final knock is to realize that the "box" after several iterations has become so small, that we can in fact apply one of the seven gadgets to eliminate it completely in one step. This concludes our description of the proof of (8). In Appendix B, we list the logical steps of this proof.

6. Some corollaries

Here we list some further theorems which follow from Theorem 2 by holographic reductions. This uses Theorem 1. Thus Theorem 2 applies to more general pairs of signatures.

Corollary 7. *Let $\alpha_1, \alpha_2, \beta_1$ and β_2 be four real numbers and $\alpha_1\beta_2 - \alpha_2\beta_1 \neq 0$. Let $f = (\alpha_1, \beta_1)^{\otimes 3} + (\alpha_2, \beta_2)^{\otimes 3}$. Then $\#f \mid [x_0, x_1, x_2]$ is #P-hard unless one of the following is true:*

(1) $x_1^2 = x_0x_2$;
(2) $x_0\alpha_1^2 + 2x_1\alpha_1\beta_1 + x_2\beta_1^2 = x_0\alpha_2^2 + 2x_1\alpha_2\beta_2 + x_2\beta_2^2 = 0$ *or*
$x_0\alpha_1\alpha_2 + x_1\alpha_1\beta_2 + x_1\beta_1\alpha_2 + x_2\beta_1\beta_2 = 0$.
(3) $x_0\alpha_1^2 + 2x_1\alpha_1\beta_1 + x_2\beta_1^2 = x_0\alpha_1\alpha_2 + x_1\alpha_1\beta_2 + x_1\beta_1\alpha_2 + x_2\beta_1\beta_2 = -(x_0\alpha_2^2 + 2x_1\alpha_2\beta_2 + x_2\beta_2^2)$, *or*
$x_0\alpha_1^2 + 2x_1\alpha_1\beta_1 + x_2\beta_1^2 = -(x_0\alpha_1\alpha_2 + x_1\alpha_1\beta_2 + x_1\beta_1\alpha_2 + x_2\beta_1\beta_2) = -(x_0\alpha_2^2 + 2x_1\alpha_2\beta_2 + x_2\beta_2^2)$.

On the other hand, the counting problem is computable in polynomial time in these three cases.

It may be interesting to see some concrete examples.

Corollary 8. $\#[1, 0, 1, 0] \mid [x_0, x_1, x_2]$ *is #P-hard unless one of the following is true:* (1) $x_1^2 = x_0x_2$; (2) $x_0 + x_2 = x_1 = 0$ *or* $x_0 = x_2$; (3) $x_0 = x_1 = -x_2$ *or* $x_0 = -x_1 = -x_2$. *The counting problem is computable in polynomial time in these three cases.*

Corollary 9. $\#[0, 1, 1, 1] \mid [x_0, x_1, x_2]$ *is #P-hard unless one of the following is true:* (1) $x_1^2 = x_0x_2$; (2) $x_0 = 2x_1 + x_2 = 0$ *or* $x_0 + x_1 = 0$; (3) $2x_0 + x_2 = x_1 = 0$ *or* $2x_0 + x_1 = x_1 + x_2 = 0$. *The counting problem is computable in polynomial time in these three cases.*

In conclusion, we proved in this paper a complexity dichotomy theorem for counting problems over 2–3 regular graphs. The restriction of regularity 2 and 3 makes the proof of hardness more difficult. We remark that over complex numbers, there are new tractable cases. For example $\#[1, 0, 0, 1] \mid [1, \omega, -\omega^2]$, where $\omega = e^{\frac{2\pi i}{3}}$, is tractable. The algorithm is by holographic reduction: under the basis transformation $T = \begin{bmatrix} 1 & 0 \\ 0 & \omega \end{bmatrix}$, $[1, 0, 0, 1]$ remains unchanged, while $[1, \omega, -\omega^2]$ goes to $[1, 1, -1]$. The polynomial time algorithm for $\#[1, 0, 0, 1] \mid [1, 1, -1]$ also gives a polynomial time algorithm for $\#[1, 0, 0, 1] \mid [1, \omega, -\omega^2]$. On the other hand, if Equality gates of arbitrary arity are available, the problem $\#\{=_1, =_2, =_3, \ldots, =_k, \ldots\} \mid [1, \omega, -\omega^2]$ is #P-hard.

Acknowledgements

A preliminary version of this paper appeared in TAMC 2009 [10]. We thank the three anonymous referees of our paper, for a very careful reading of our paper and insightful comments. We would also like to thank very much Xi Chen, Michael Kowalczyk and Leslie Valiant for comments and discussions.

The first author was supported by NSF CCF-0830488 and CCF-0914969 and the third author was supported by the Grand Challenge Program "Network Algorithms and Digital Information" of the Institute of Software, CAS, and NSFC 60970003.

Appendix A. Figures

Here we exhibit some numerical computation results, which show the complexity of the failure sets.

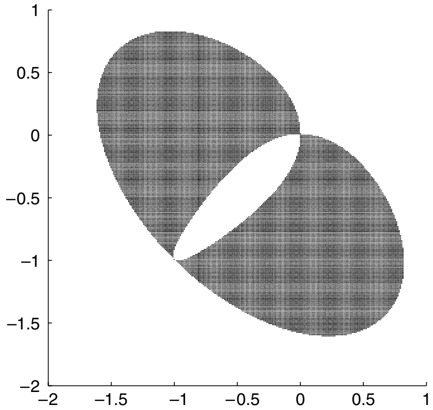


Fig. 11. The failure set of the 1st gadget.

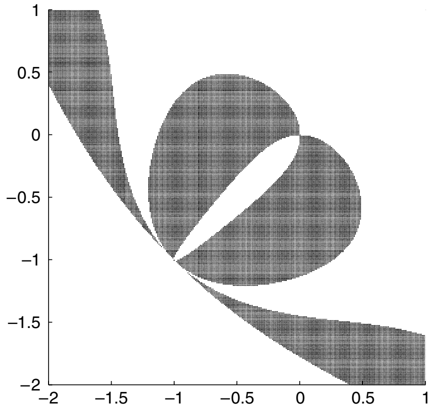


Fig. 12. The failure set of the 2nd gadget.

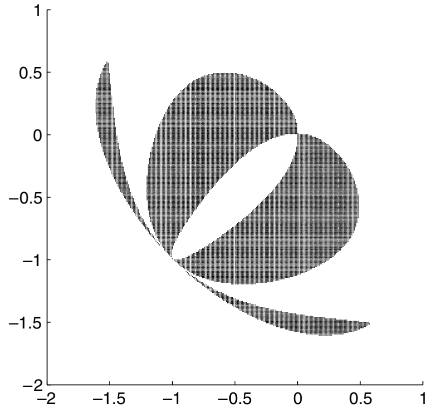


Fig. 13. The intersection of the first 2 failure sets.

The first batch of figures (Figs. 11–23) shows that the initial seven gadgets produce progressively thinner intersections of failure sets, and ultimately the intersection seems to have vanished totally. Unfortunately this numerical truth is beyond the capability to be verified by symbolic proof, which causes all sorts of complications.

The next two figures (Figs. 24 and 25) show what happens if we use all the gadgets that symbolic computation can handle. It becomes thin, but not quite vanishing.

We omit the pictures for the seven gadgets that we did use, which leave a small uncovered region.

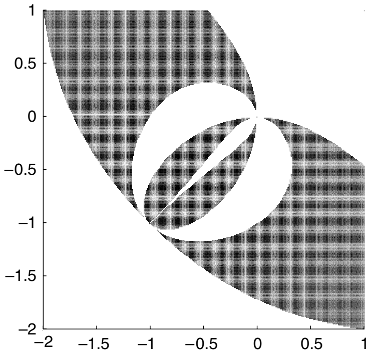


Fig. 14. The failure set of the 3rd gadget.

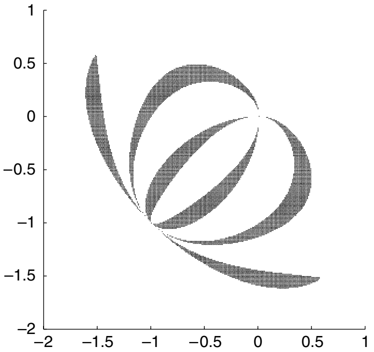


Fig. 15. The intersection of the first 3 failure sets.

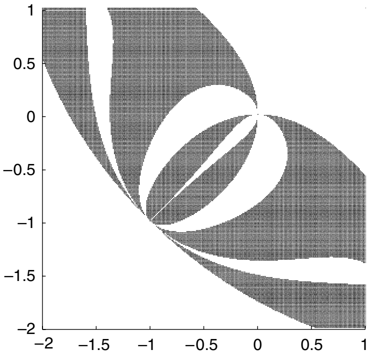


Fig. 16. The failure set of the 4th gadget.

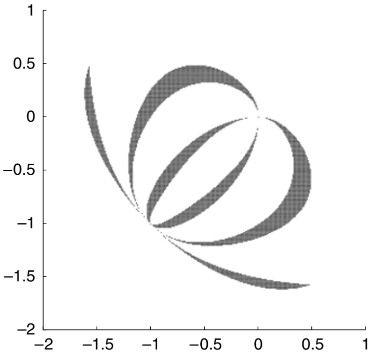


Fig. 17. The intersection of the first 4 failure sets.

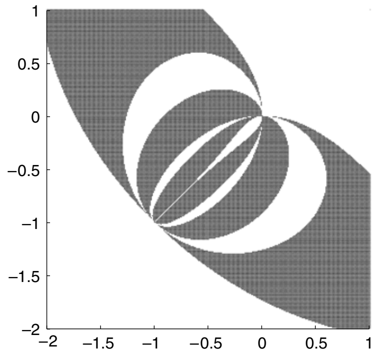


Fig. 18. The failure set of the 5th gadget.

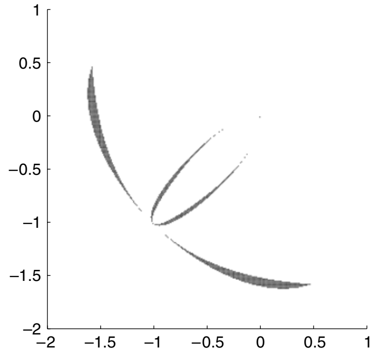


Fig. 19. The intersection of the first 5 failure sets.

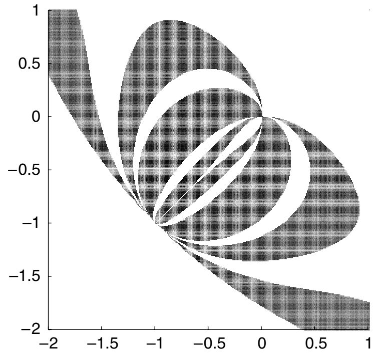


Fig. 20. The failure set of the 6th gadget.

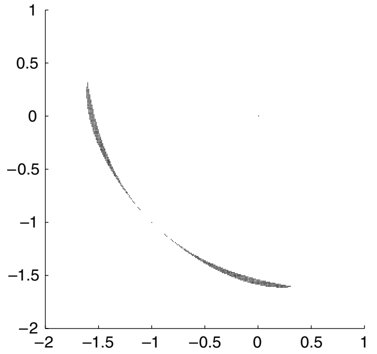


Fig. 21. The intersection of the first 6 failure sets.

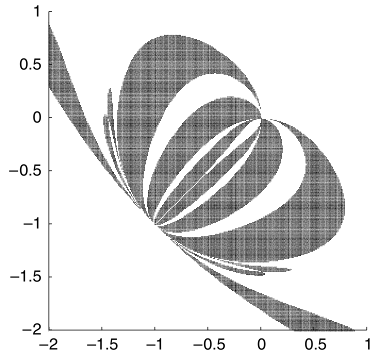


Fig. 22. The failure set of the 7th gadget.

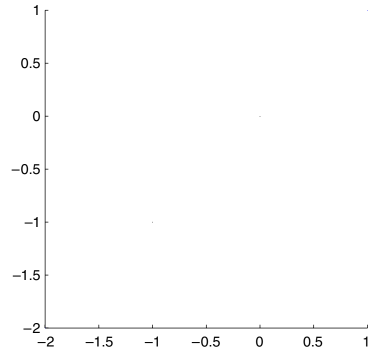


Fig. 23. The intersection of the first 7 failure sets.

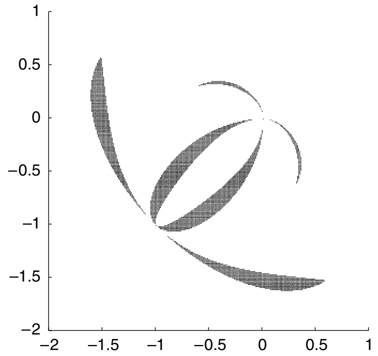


Fig. 24. The failure set of all gadgets with $m \leq 4$.

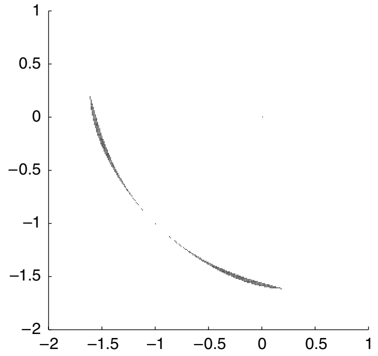


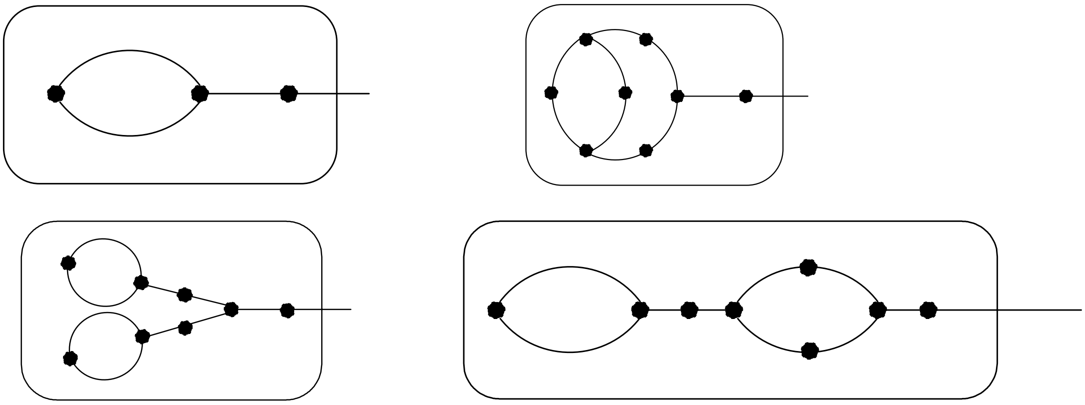
Fig. 25. The Failure set of all 170 plus gadgets with $m \leq 6$.

Appendix B. Logical steps of the proof

In this section, we give the logical steps of the proof of Lemma 4, by proving (8). We assume that the software Mathematica™ correctly implemented the decision algorithm for semi-algebraic sets.

We will use four initial gadgets g_j ($j \in [4]$) and seven iteration gadgets A_i ($i \in [7]$).

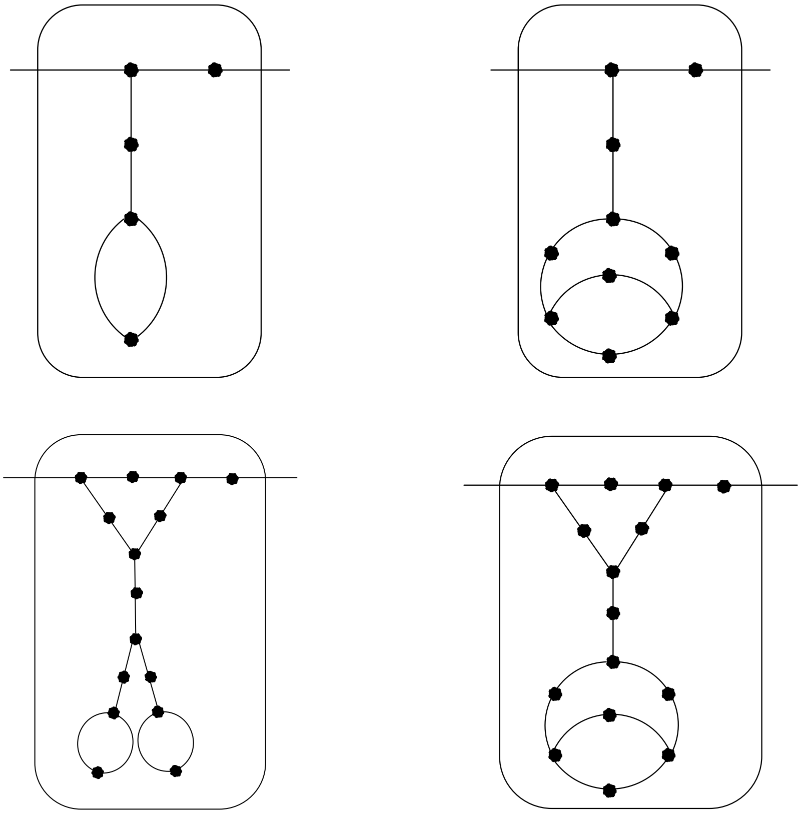
Here are the 4 initial gadgets g_1–g_4:

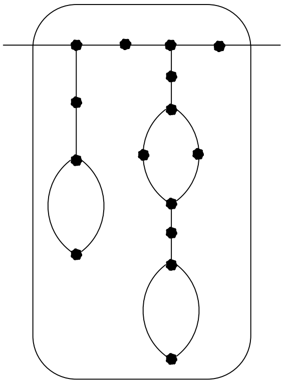


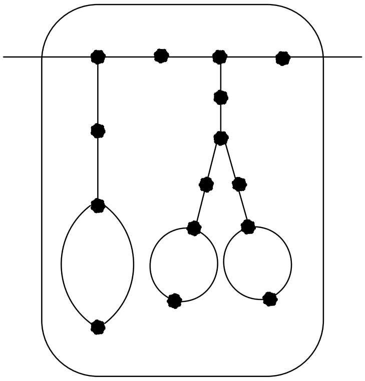
The signatures of these four gadgets are listed as follows:

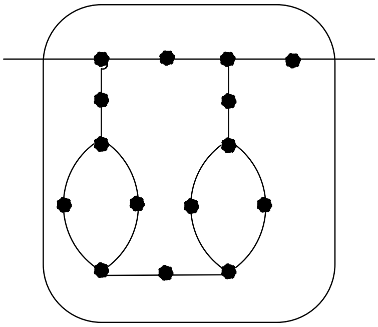
$$\begin{aligned}
g_1 &= [a^2 + b, a + b^2];\\
g_2 &= [a^5 + 3a^2 + ab^2 + 2b + b^4, b^5 + 3b^2 + ba^2 + 2a + a^4];\\
g_3 &= [a^5 + 2ba^3 + 3ab^2 + a^2 + b^4, a^4 + 3ba^2 + b^2 + 2ab^3 + b^5];\\
g_4 &= [a^5 + ba^3 + 2a^2 + 2ab^2 + b + b^4, a^4 + 2ba^2 + a + 2b^2 + ab^3 + b^5].
\end{aligned}$$

Next we list the seven iteration gadgets A_1 to A_7 and their signatures. The first two gadgets are









$$A_1 = \begin{bmatrix} a(a^2+b) & a+b^2 \\ a^2+b & b(a+b^2) \end{bmatrix};$$

$$A_2 = \begin{bmatrix} a(a^5+3a^2+ab^2+2b+b^4) & b^5+3b^2+ba^2+2a+a^4 \\ a^5+3a^2+ab^2+2b+b^4 & b(b^5+3b^2+ba^2+2a+a^4) \end{bmatrix}.$$

The remaining 5 gadgets of the seven gadgets are depicted with their signatures A_3–A_7. For A_3–A_7, in matrix form, the first two listed entries make up the first column, and the remaining two entries make up the second column.

$A_3 = [a^9 + 2ba^7 + 3a^5b^2 + 3a^6 + a^4b^4 + 6a^4b + 7a^2b^2 + 2a^3b^3 + a^2b^5 + a^3 + 3ab^4 + b^3 + b^6, a^8 + 3ba^6 + 6a^4b^2 + 2a^5 + a^3b^4 + 4ba^3 + ab^2 + 8a^2b^3 + 4ab^5 + b^4 + b^7, a^7 + 4ba^5 + 8a^3b^2 + a^4 + 6a^2b^4 + 4ab^3 + 3ab^6 + ba^2 + 2b^5 + a^4b^3 + b^8, a^6 + 3a^4b + 7a^2b^2 + a^3 + 6ab^4 + b^3 + 3b^6 + a^5b^2 + 2a^3b^3 + a^4b^4 + 3a^2b^5 + 2ab^7 + b^9]$;

$A_4 = [a^9 + 5a^6 + a^5b^2 + 4a^4b + a^4b^4 + 5a^3 + 5a^2b^2 + a^2b^5 + 4ab + ab^4 + 3b^3 + b^6, a^8 + 4a^5 + 2a^4b^2 + 6ba^3 + a^3b^4 + 2a^2 + 7ab^2 + 2ab^5 + ba^6 + 2a^2b^3 + 3b^4 + b^7, a^7 + 3a^4 + 2a^3b^2 + 7ba^2 + 2a^2b^4 + 2ba^5 + 6ab^3 + ab^6 + 2b^2 + 4b^5 + a^4b^3 + b^8, a^6 + 3a^3 + 5a^2b^2 + 4ab + 4ab^4 + a^4b + 5b^3 + 5b^6 + a^5b^2 + a^4b^4 + a^2b^5 + b^9]$;

$A_5 = [a^9 + 2ba^7 + 3a^5b^2 + 3a^6 + a^4b^4 + 6a^4b + 5a^2b^2 + 3a^3b^3 + 2a^2b^5 + a^3 + ab + 2b^3 + ab^4 + b^6, a^8 + 3ba^6 + 6a^4b^2 + 2a^5 + 2a^3b^4 + 4ba^3 + 2ab^2 + 6a^2b^3 + 2ab^5 + a^2b^6 + 2b^4 + b^7, a^7 + 2ba^5 + 2a^4 + 6a^3b^2 + 2ba^2 + 6a^2b^4 + a^6b^2 + 2a^4b^3 + 4ab^3 + 3ab^6 + 2b^5 + b^8, a^6 + a^4b + 2a^3 + 5a^2b^2 + ab + 6ab^4 + 2a^5b^2 + 3a^3b^3 + b^3 + 3b^6 + a^4b^4 + 3a^2b^5 + 2ab^7 + b^9]$;

$A_6 = [a^9 + 3ba^7 + 5a^5b^2 + 2a^6 + a^4b^4 + 5a^3b^3 + 5a^4b + 2a^2b^5 + 4a^2b^2 + b^3 + 2ab^4 + b^6, a^8 + 4ba^6 + 9a^4b^2 + a^5 + 3a^3b^4 + 7a^2b^3 + ba^3 + 3ab^5 + a^2b^6 + b^4 + b^7, a^7 + 3ba^5 + 7a^3b^2 + a^4 + 9a^2b^4 + a^6b^2 + 3a^4b^3 + 4ab^6 + ab^3 + b^5 + b^8, a^6 + 2a^4b + 4a^2b^2 + a^3 + 5ab^4 + 2a^5b^2 + 5a^3b^3 + 2b^6 + 5a^2b^5 + 3ab^7 + a^4b^4 + b^9]$;

$A_7 = [a^9 + 5a^6 + 2a^4b + 2a^5b^2 + 6a^3 + 3a^3b^3 + 3a^2b^2 + a^2b^5 + 3ab + ab^4 + 1 + 3b^3 + b^6, a^8 + 4a^5 + 4ba^3 + 3a^4b^2 + 3a^2 + 3a^2b^3 + 5ab^2 + 2ab^5 + ba^6 + a^3b^4 + b + 3b^4 + b^7, a^7 + 3a^4 + 5ba^2 + 2ba^5 + a^4b^3 + 3a^2b^4 + a + 3a^3b^2 + 4ab^3 + ab^6 + 3b^2 + 4b^5 + b^8, a^6 + 3a^3 + 3ab + a^4b + 3a^3b^3 + 2ab^4 + 1 + 3a^2b^2 + 6b^3 + 5b^6 + a^5b^2 + 2a^2b^5 + b^9]$.

So, overall we have 4 × 7 = 28 recursive constructions. As described in Section 5, there will be some exceptional points for the intersection of the failure sets of these gadgets. These exceptional points come in three categories, (1) a small region, (2) some finitely many curve segments, and (3) some finitely many points. Points in categories (1) and (2) will be bounded by boxes which are defined by simpler curves. We will denote by B_1 and B_2 some suitable bounding boxes, and B_3 will denote some finitely many points. These will be specified later. We want to prove

$$\bigcap_{i,j} \mathcal{F}(A_i, g_j) \subseteq E \cup B_1 \cup B_2 \cup B_3.$$

First we do some logical transformation to decrease the complexity of the symbolic verification. This uses the fact that the main failure sets only depend on A_i and not g_j.

$$\begin{aligned}\bigcap_{i,j}\mathcal{F}(A_i,g_j) &= \bigcap_{i,j}[[\det(A_i)=0]\cup[\operatorname{tr}A_i=0]\cup[(\operatorname{tr}A_i)^2-4\det(A_i)\le 0]\cup[\det[A_ig_j,g_j]=0]]\\ &= \bigcap_i\left[([\det(A_i)=0]\cup[\operatorname{tr}A_i=0]\cup[(\operatorname{tr}A_i)^2-4\det(A_i)\le 0])\cup\left(\bigcap_j[\det[A_ig_j,g_j]=0]\right)\right].\end{aligned}$$

For $i = 1, 2, 3, 5, 6$, we can verify using Mathematica™ that

$$\bigcap_j[\det[A_ig_j,g_j]=0]\subset E.$$

For $i = 4$, Mathematica™ tells us that there are two more points $(-\sqrt[3]{4},\sqrt[3]{2})$ and $(\sqrt[3]{2},-\sqrt[3]{4})$. And for $j = 7$, there are also two more points $(a, b) = (r_1, r_2)$ and $(a, b) = (r_2, r_1)$, where r_1 and r_2 are the two real roots of $Z^6 + 2Z^3 - 1 = 0$. We will put these four exceptional points to B_3. Then we will not worry about the initial value part and focus on the remaining, which only depends on A_i, $1 \le i \le 7$.

So we will prove

$$\bigcap_i\left([\det(A_i)=0]\cup[\operatorname{tr}A_i=0]\cup[(\operatorname{tr}A_i)^2-4\det(A_i)\le 0]\right)\subset E\cup B_1\cup B_2\cup B_3.$$

The complexity of this verification is still too high for Mathematica™. We observe that all the sets involved above are symmetric for a and b. So we use the transformation $a = -\sqrt{x} + y - 1$, $b = \sqrt{x} + y - 1$. The symmetry of a and b implies that this set is also a semi-algebraic set of x, y. This transformation reduced the degree of the polynomials and as a result the complexity of the verification.

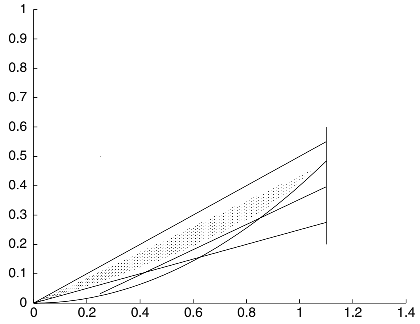
In the plane of (x, y), we define

$$\begin{aligned}B_1 &= \{(x,y)\in\mathbb{R}^2 \mid [x>0]\,\&\,[x<11/10]\,\&\,[y\le x/2]\,\&\,[y\ge x/4]\,\&\,[y\ge 2x^2/5]\,\&\,[y\ge 3x/7-3/40]\};\\ B_2 &= \{(x,y)\in\mathbb{R}^2 \mid 0<x<0.14\,\&\,0\le y\le 1\}.\end{aligned}$$

These two boxes are suggested by numerical computations. The following figure shows a shaded region which is the main failure set and the set B_1 which contains it. After this reduction of degrees, Mathematica™ is able to verify that, with nine extra points and the two pairs of exceptional points mentioned above, the intersection of failure sets is contained in $E \cup B_1 \cup B_2$. All these exceptional points constitute the set B_3. We remark that the actual failure set is very complicated, whose description by Mathematica™ requires *several pages* of equations.

The remaining task is to prove #P-hardness for all the points in B_1, B_2 and B_3. Any point outside $E \cup B_1 \cup B_2 \cup B_3$ has already been proved to be #P-hard.

Using Mathematica™ we can verify that all points in B_3 will go out of $E \cup B_1 \cup B_2 \cup B_3$ after at most three iterative mappings using the mappings f_1 and f_2 mentioned in Section 5.

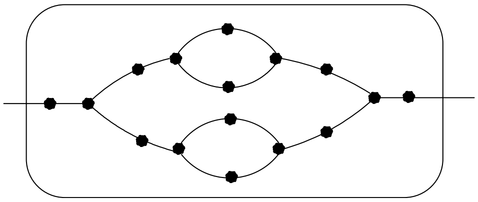


$$\begin{aligned}f_1 &: (a,b)\mapsto\left(\frac{a^2(a^2+b)+a+b^2}{a^3+2ab+b^3},\frac{a^2+b+b^2(a+b^2)}{a^3+2ab+b^3}\right);\\ f_2 &: (a,b)\mapsto\left(\frac{a(a^3+1)+a+b^2}{a^3+ab+1+b^3},\frac{a^2+b+b(1+b^3)}{a^3+ab+1+b^3}\right).\end{aligned}$$

In xy coordinates, these maps take the form

$$f_1\ :\ (x,y)\mapsto\left(\frac{4x(5y-6y^2+2xy+2y^3-2x)^2}{(6xy-7x+2y^3-5y^2+4y)^2},\frac{2y^2+y^4-2y^3+6xy^2-6xy+x^2}{6xy-7x+2y^3-5y^2+4y}\right);$$
$$f_2\ :\ (x,y)\mapsto\left(\frac{x(8y+4y^3-11y^2-5x+4xy)^2}{4(3xy-4x+y^3-2y^2+y)^2},\frac{y^4-y^3+6xy^2-7xy+x^2}{2(3xy-4x+y^3-2y^2+y)}\right).$$

In fact for one particular point on the xy plane (this corresponds to two points on the ab plane, $(a,b)=(0,-1)$ and $(a,b)=(-1,0)$), we have to use the following third gadget, with its more complicated mapping f_3.



Now we deal with the boxes B_1 and B_2. After one iteration using the maps f_1 and f_2 above, we define $\tilde{B}$ to be the subset of points (x,y) in $B_1\cup B_2$ such that neither f_1 nor f_2 can map the point (x,y) outside of $B_1\cup B_2$. We can re-bound the subset $\tilde{B}$ by the following two new boxes:

$$B_1'=(y\geq 1/10x\ \&\ y\geq 3x/7-1/10\ \&\ y\leq 23x/70\ \&\ x\leq 7/10);$$
$$B_2'=(y\geq 43x/30\ \&\ y\leq 4x\ \&\ y\geq 4x-8/25\ \&\ y\leq 2/5).$$

The next iteration we can bound the images by only one box

$$B''=(y\leq 24x/70\ \&\ y\geq 13x/70\ \&\ x\leq 7/10).$$

After several iterations we can bound the subset where #P-hardness is still unsettled by a single box:

$$B'''=(y\leq 3x/14\ \&\ y\geq x/5\ \&\ x\leq 3/10).$$

Then we directly verify using Mathematica$^{\text{TM}}$ that the set B''' is in fact already killed by the seven gadgets listed at the beginning of this section.

This completes the proof. □

References

[1] S. Basu, R. Pollack, M.-F. Roy, A new algorithm to find a point in every cell defined by a family of polynomials, in: B. Caviness, J. Johnson (Eds.), Quantifier Elimination and Cylindrical Algebraic Decomposition, in: Texts and Mongraphs in Symbolic Computation, Springer, 1998, pp. 341–350.
[2] M. Ben-Or, D. Kozen, J. Reif, The complexity of elementary algebra and geometry, J. Comput. Syst. Sci. (1986) 457–464.
[3] L. Blum, F. Cucker, M. Shub, S. Smale, Complexity and Real Computation, Springer, 1998.
[4] A.A. Bulatov, V. Dalmau, Towards a dichotomy theorem for the counting constraint satisfaction problem, Inf. Comput. 205 (5) (2007) 651–678.
[5] A.A. Bulatov, M. Grohe, The complexity of partition functions, Theor. Comput. Sci. 348 (2–3) (2005) 148–186.
[6] J-Y. Cai, X. Chen, P. Lu, Graph homomorphisms with complex values: a Dichotomy theorem, ICALP (1) (2010) 275–286.
[7] J-Y. Cai, X. Chen, P. Lu, Graph homomorphisms with complex values: a Dichotomy Theorem CoRR abs/0903.4728: (2009).
[8] J-Y. Cai, P. Lu, Holographic algorithms: from art to science, in: The Proceedings of STOC 2007, pp 401–410.
[9] J-Y. Cai, P. Lu, M. Xia, Holographic algorithms by fibonacci gates and holographic reductions for hardness, in: FOCS 2008, pp. 644–653.
[10] J-Y. Cai, P. Lu, M. Xia, A computational proof of complexity of some restricted counting problems, in: TAMC 2009, pp. 138–149.
[11] J-Y. Cai, P. Lu, M. Xia, Holant problems and counting CSP, in: STOC 2009, pp. 715–724.
[12] J. Canny, Some algebraic and geometric computations in PSPACE, in: Proc. ACM Symp. Theory of Computing, 1988, pp. 460–469.
[13] G. Collins, Quantifier elimination for real closed fields by cylindric algebraic decomposition, in: 2nd GI Conference on Automata Theory and Formal Languages, 1975, pp. 134–183.
[14] N. Creignou, M. Hermann, Complexity of generalized satisfiability counting problems, Inf. Comput. 125 (1) (1996) 1–12.
[15] N. Creignou, S. Khanna, M. Sudan, Complexity classifications of boolean constraint satisfaction problems, in: SIAM Monographs on Discrete Mathematics and Applications, 2001.
[16] M.E. Dyer, L.A. Goldberg, M. Jerrum, The Complexity of Weighted Boolean #CSP CoRR abs/0704.3683: (2007).
[17] M.E. Dyer, L.A. Goldberg, M. Paterson, On counting homomorphisms to directed acyclic graphs, J. ACM 54 (6) (2007).
[18] M.E. Dyer, C.S. Greenhill, The complexity of counting graph homomorphisms, Random Struct. Algorithm 17 (3–4) (2000) 260–289.
[19] L.A. Goldberg, M. Grohe, M. Jerrum, M. Thurley, A complexity dichotomy for partition functions with mixed signs. CoRR abs/0804.1932: (2008).
[20] P. Hell, J. Nešetřil, On the complexity of H-coloring, J. Combin. Theory Ser. B 48 (1990) 92–110.
[21] A. Tarski, A decision method for elementary algebra and geometry, Univ. of Calif., 1951.
[22] K. Ko, Computational Complexity of Real Functions, Birkhauser Boston, Boston, MA, 1991.
[23] S.P. Vadhan, The complexity of counting in sparse, regular, and planar graphs, SIAM J. Comput. 31 (2) (2001) 398–427.
[24] L.G. Valiant, Quantum circuits that can be simulated classically in polynomial time, SIAM J. Comput. 31 (4) (2002) 1229–1254.
[25] L.G. Valiant, Holographic algorithms (extended abstract), in: Proc. 45th IEEE Symposium on Foundations of Computer Science, 2004, pp. 306–315. A more detailed version appeared in ECCC Report TR05-099.