Counting Homomorphisms to Trees Modulo a Prime

Andreas Göbel
Hasso Plattner Institute, University of Potsdam, Potsdam, Germany

J. A. Gregor Lagodzinski
Hasso Plattner Institute, University of Potsdam, Potsdam, Germany

Karen Seidel
Hasso Plattner Institute, University of Potsdam, Potsdam, Germany

Abstract

Many important graph theoretic notions can be encoded as counting graph homomorphism problems, such as partition functions in statistical physics, in particular independent sets and colourings. In this article we study the complexity of $\#_p\textsc{HomsTo}H$, the problem of counting graph homomorphisms from an input graph to a graph H modulo a prime number p. Dyer and Greenhill proved a dichotomy stating that the tractability of non-modular counting graph homomorphisms depends on the structure of the target graph. Many intractable cases in non-modular counting become tractable in modular counting due to the common phenomenon of cancellation. In subsequent studies on counting modulo 2, however, the influence of the structure of H on the tractability was shown to persist, which yields similar dichotomies.

Our main result states that for every tree H and every prime p the problem $\#_p\textsc{HomsTo}H$ is either polynomial time computable or $\#_p\mathsf{P}$-complete. This relates to the conjecture of Faben and Jerrum stating that this dichotomy holds for every graph H when counting modulo 2. In contrast to previous results on modular counting, the tractable cases of $\#_p\textsc{HomsTo}H$ are essentially the same for all values of the modulo when H is a tree. To prove this result, we study the structural properties of a homomorphism. As an important interim result, our study yields a dichotomy for the problem of counting weighted independent sets in a bipartite graph modulo some prime p. These results are the first suggesting that such dichotomies hold not only for the one-bit functions of the modulo 2 case but also for the modular counting functions of all primes p.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness, Mathematics of computing → Graph theory, Mathematics of computing → Combinatorics

Keywords and phrases Algorithms, Theory, Graph Homomorphisms, Counting Modulo a Prime, Complexity Dichotomy

Digital Object Identifier 10.4230/LIPIcs.MFCS.2018.49

Related Version A full version of the paper is available at `https://arxiv.org/abs/1802.06103`.

Acknowledgements The first author would like to thank Leslie Ann Goldberg and David Richerby for fruitful discussions during the early stages of this work.

1 Introduction

Edge preserving functions between the vertices of two graphs, known as *graph homomorphisms*, generate a powerful language expressing important notions; examples include constraint satisfaction problems and partition functions in statistical physics. As such, the computational

43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018).
Editors: Igor Potapov, Paul Spirakis, and James Worrell; Article No. 49; pp. 49:1–49:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

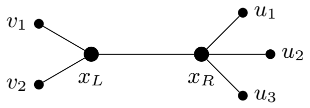

Figure 1 The graph H will be our recurring example and the labelling of the vertices is justified later in the introduction.

complexity of graph homomorphism problems has been studied extensively from a wide range of views. Early results include that of Hell and Nešetřil [14], who study the complexity of HomsToH, the problem of deciding if there exists a homomorphism from an input graph G to a fixed graph H. They show the following dichotomy: if H is bipartite or has a loop, the problem is in P and in every other case HomsToH is NP-complete. In particular, this is of interest since a result of Ladner [15] shows that if P $\neq$ NP, then there exist problems that are neither in P nor NP-hard.

Dyer and Greenhill [5] show a dichotomy for the problem #HomsToH, the problem of counting the homomorphisms from an input graph G to H. Their theorem states that #HomsToH is tractable if H is a complete bipartite graph or a complete graph with loops on all vertices; otherwise #HomsToH is #P-complete. This dichotomy was progressively extended to weighted sums of homomorphisms with integer weights, by Bulatov and Gohe [1]; with real weights, by Goldberg et al. [11]; finally, with complex weights, by Cai, Chen and Lu [2].

We study the complexity of counting homomorphisms modulo a prime p. The set of homomorphisms from the input graph G to the target graph H is denoted by $\mathrm{Hom}\,(G \to H)$. For each pair of fixed parameters p and H, we study the computational problem $\#_p$HomsToH, that is the problem of computing $|\mathrm{Hom}\,(G \to H)\,|$ modulo p. The value of p and the structure of the target graph H influence the complexity of $\#_p$HomsToH. Consider the graph H in Figure 1. Our results show that $\#_p$HomsToH is computable in polynomial time when $p = 2, 3$ while it is hard for any other prime p.

Our main goal is to fully characterise the complexity of $\#_p$HomsToH in a dichotomy theorem. In this manner we aim to determine for which pair of parameters (H, p) the problem is tractable and show that for every other pair of parameters the problem is hard. As the theorem of Ladner [15] extends to the modular counting problems, it is not obvious that there are no instances of $\#_p$HomsToH with an intermediate complexity.

The first study of graph homomorphisms under the setting of modular counting has been conducted by Faben and Jerrum [7]. Their work is briefly described in the following and we assume the reader to be familiar with the notion of an automorphism and its order. We provide the formal introduction in the full version. Given a graph H and an automorphism ϱ of H, H^{ϱ} denotes the subgraph of H induced by the fixpoints of ϱ. We write $H \Rightarrow_k H'$ if there is an automorphism ϱ of order k of H such that $H^{\varrho} = H'$ and we write $H \Rightarrow_k^* H'$ if either H is isomorphic to H' (written $H \cong H'$) or, for some positive integer t, there are graphs $H_1, \ldots, H_t$ such that $H \cong H_1$, $H_1 \Rightarrow_k \cdots \Rightarrow_k H_t$, and $H_t \cong H'$.

Faben and Jerrum showed [7, Lemma 3.3] that if the order of ϱ is a prime p, then $|\mathrm{Hom}\,(G \to H)\,|$ is equivalent to $|\mathrm{Hom}\,(G \to H^{\varrho})\,|$ modulo p. Furthermore, they showed [7, Theorem 3.7] that there is (up to isomorphism) exactly one graph H^{*p} without automorphisms of order p, such that $H \Rightarrow_p^* H^{*p}$. This graph H^{*p} is called the *order p reduced form* of H (see Figure 4 of the full version). If H^{*p} falls into the polynomial computable cases of the theorem of Dyer and Greenhill, then $\#_p$HomsToH is computable in polynomial time as well. For $p = 2$, Faben and Jerrum conjectured that these are the only instances computable in polynomial time and that in every other case $\#_2$HomsToH is $\#_2\,\mathsf{P}$-complete, where $\#_k\,\mathsf{P}$ is the "canonical" hardness class for modular counting problems (see Section 1.1).

▶ **Conjecture 1.1** (Faben and Jerrum [7]). Let H be a graph. If its order 2 reduced form H^{*2} has at most one vertex, then $\#_2$HomsToH is in FP; otherwise, $\#_2$HomsToH is $\#_2\,\mathsf{P}$-complete.

Faben and Jerrum [7, Theorem 3.8] underlined their conjecture by proving it for the case in which H is a tree. In subsequent works this proof was extended to cactus graphs in [9] and to square-free graphs in [10], by Göbel, Goldberg and Richerby.

The present work follows a direction orthogonal to the aforementioned. Instead of proving the conjecture for richer classes of graphs, we show a dichotomy for all primes, starting again by studying trees.

▶ **Theorem 1.2.** *Let p be a prime and let H be a graph, such that its order p reduced form H^{*p} is a tree. If H^{*p} is a star, then $\#_p$HomsToH is computable in polynomial time; otherwise, $\#_p$HomsToH is $\#_p\,\mathsf{P}$-complete.*

Our results are the first to suggest that the conjecture of Faben and Jerrum might apply to counting graph homomorphisms modulo every prime p instead of counting modulo 2. This suggestion, however, remains hypothetical. Borrowing the words of Dyer, Frieze and Jerrum [4]: "One might even rashly conjecture" it "(though we shall not do so)".

To justify our title we give the following corollary, stating a dichotomy for all trees H.

▶ **Corollary 1.3.** *Let p be a prime and let H be a tree. If the order p reduced form H^{*p} of H is a star, then $\#_p$HomsToH is computable in polynomial time; otherwise, $\#_p$HomsToH is $\#_p\,\mathsf{P}$-complete.*

We illustrate Theorem 1.2 using the following discussion on Figure 1. The order 2 and the order 3 reduced form of H both are the graph with one vertex, whereas for any other prime the graph stays as such.

The polynomial computable cases follow directly from the results of Faben and Jerrum. Thus, to prove Theorem 1.2 it suffices to show that $\#_p$HomsToH is $\#_p\,\mathsf{P}$-complete for every tree H that is not a star and has no automorphism of order p. The reductions in [7, 9, 10] show hard instances of $\#_2$HomsToH by starting from $\#_2$IS, the problem of computing $|\mathcal{I}(G)| \pmod 2$, where $\mathcal{I}(G)$ is the set of independent sets of G. $\#_2$IS was shown to be $\#_2\,\mathsf{P}$ complete by Valiant [18]. Later on, Faben [6] extended this result by proving $\#_k$IS to be $\#_k\,\mathsf{P}$-complete for all integers k. For reasons to be explained in Section 1.3 we do not use this problem as a starting point for our reductions.

We turn our attention to $\#_p$BIS, the problem of counting the independent sets of a bipartite graph modulo p. In the same work Faben [6] includes a construction to show hardness for $\#_p$BIS. We employ the weighted version $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$ as a starting point for our reduction extending the research on #BIS.

▶ **Problem 1.4.** $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$

Parameter.	p prime and $\lambda_\ell, \lambda_r \in \mathbb{Z}_p$.
Input.	Bipartite graph $G = (V_L, V_R, E)$.
Output.	$Z_{\lambda_\ell,\lambda_r}(G) = \sum_{I \in \mathcal{I}(G)} \lambda_\ell^{\lvert V_L \cap I\rvert} \lambda_r^{\lvert V_R \cap I\rvert} \pmod p$.

In fact, we obtain the following dichotomy.

▶ **Theorem 1.5.** *Let p be a prime and let λ_ℓ, $\lambda_r \in \mathbb{Z}_p$. If $\lambda_\ell \equiv 0 \pmod p$ or $\lambda_r \equiv 0 \pmod p$, then $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$ is computable in polynomial time. Otherwise, $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$ is $\#_p\,\mathsf{P}$-complete.*

In order to prove hardness for $\#_p\textsc{HomsTo}H$ we employ a reduction in three phases: (i) we reduce the "canonical" $\#_p\,\mathsf{P}$-complete problem $\#_p\mathrm{SAT}$ to $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$; (ii) we reduce $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$ to $\#_p\textsc{PartLabHomsTo}H$, a restricted version of $\#_p\textsc{HomsTo}H$ which we define in Section 1.3; (iii) we reduce $\#_p\textsc{PartLabHomsTo}H$ to $\#_p\textsc{HomsTo}H$.

Section 1.1 provides background knowledge on modular counting. In Section 1.2 we will discuss some related work. A high level proof of our three way reduction is provided in Section 1.3. There we also explain the technical obstacles arising from values of the modulo $p > 2$ and how we overcome them by generalising the techniques used for the case $p = 2$. First, we explain step (i), the reduction from $\#_p\mathrm{SAT}$ to $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$. Afterwards, we describe step (iii), the reduction from $\#_p\textsc{PartLabHomsTo}H$ to $\#_p\textsc{HomsTo}H$ establishing the required notation for the subsequent illustration of step (ii), the reduction from $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$ to $\#_p\textsc{PartLabHomsTo}H$. In Section 1.4 we discuss the limits of our techniques, which do not yield a dichotomy modulo any integer k.

1.1 Modular counting

Modular counting was originally studied from the decision problem's point of view. Here, the objective is to determine if the number of solutions is non-zero modulo k. The complexity class $\oplus\,\mathsf{P}$ was first studied by Papadimitriou and Zachos [16] and by Goldschlager and Parberry [12]. $\oplus\,\mathsf{P}$ consists of all problems of the form "is $f(x)$ odd or even?", where $f(x)$ is a function in $\#\,\mathsf{P}$. A result of Toda [17] states that every problem in the polynomial time hierarchy reduces in polynomial time to some problem in $\oplus\,\mathsf{P}$. This result suggests that $\oplus\,\mathsf{P}$-completeness represents strong evidence for intractability.

For an integer k the complexity class $\#_k\,\mathsf{P}$ consists of all problems of the form "compute $f(x)$ modulo k", where $f(x)$ is a function in $\#\,\mathsf{P}$. In the special case of $k = 2$, $\#_2\,\mathsf{P} = \oplus\,\mathsf{P}$, as the instances of $\#_2\,\mathsf{P}$ require a one bit answer. Throughout this paper though, instead of the more traditional notation $\oplus\,\mathsf{P}$, we will use $\#_2\,\mathsf{P}$ to emphasise our interest in computing functions.

If a counting problem can be solved in polynomial time, the corresponding decision and modular counting problems can also be solved in polynomial time. The converse, though, does not necessarily hold. The reason is that efficient counting algorithms rely usually on an exponential number of cancellations that occur in the problem, e.g. compute the determinant of a non-negative matrix. The modulo operator introduces a natural setting for such cancellations to occur.

For instance, consider the $\#\,\mathsf{P}$-complete problem of counting proper 3-colourings of a graph G in the modulo 3 (or even modulo 6) setting. 3-colourings of a graph assigning all three colours can be grouped in sets of size 6, since there are $3! = 6$ permutations of the colours. Thus, the answer to these instances is always a multiple of 6, and therefore "cancels out". It remains to compute the number of 3-colourings assigning less than 3 colours. For the case of using exactly 2 colours we distinguish the following two cases: G is not bipartite and there are no such colourings; G is bipartite and the number of 3-colourings of G that use exactly 2 colours is $3(2^c)$, where c is the number of components of G. Finally, computing the number of proper 3-colourings of G that use exactly one colour is an easy task. Either G has an edge and there are no such colourings, or G has no edges and for every vertex there are 3 colours to choose from.

Valiant [18] observed a surprising phenomenon in the tractability of modular counting problems. He showed that for a restricted version of 3-SAT computing the number of solutions modulo 7 is in FP, but computing this number modulo 2 is $\#_2$ P-complete. This mysterious number 7 was later explained by Cai and Lu [3], who showed that the k-SAT version of Valiant's problem is tractable modulo any prime factor of $2^k - 1$.

1.2 Related work

We have already mentioned earlier work on counting graph homomorphisms. In this section we highlight the work of Faben [6] and the work of Guo et al. [13] on the complexity of the modular counting variant of the constraint satisfaction problem.

▶ **Problem 1.6.** $\#_k\mathrm{CSP}(\mathcal{F})$

Parameter. $k \in \mathbb{Z}_{>0}$ and a set of functions $\mathcal{F} = \{f_1, \dots, f_m\}$, where for each $j \in [m]$, $f_j : \{0,1\}^{r_j} \to \mathbb{Z}_p$ and $r_j \in \mathbb{Z}_{>0}$.

Input. Finite set of constraints over Boolean variables $x_1, \dots, x_n$ of the form $f_{j_l}(x_{i_{l,1}}, x_{i_{l,2}}, \dots, x_{i_{l,r_{j_l}}})$.

Output. $\sum_{x_1,\dots,x_n \in \{0,1\}} \prod_l f_{j_l}(x_{i_{l,1}}, x_{i_{l,2}}, \dots, x_{i_{l,r_{j_l}}}) \pmod k$.

Faben showed a dichotomy theorem [6, Theorem 4.11] when the functions in $\mathcal{F}$ have Boolean domain and Boolean range, i.e. $f : \{0,1\} \to \{0,1\}$. Guo et al. extended this dichotomy [13, Theorem 4.1] to $\#_k\mathrm{CSP}$, when the functions in $\mathcal{F}$ have Boolean domain $\{0,1\}$ but range in $\mathbb{Z}_k$.

Constraint satisfaction problems generalise graph homomorphism problems, when the domain of the constraint functions is arbitrarily large. In order to illustrate that $\#_k\mathrm{CSP}$ is a generalisation of $\#_k$HomsToH, let G be an input for $\#_k$HomsToH, for which we describe an equivalent $\#_k\mathrm{CSP}$ instance. The domain of the constraint satisfaction problem is $D = V(H)$ and $\mathcal{F}$ contains a single binary relation R_H, with $R_H(u,v) = 1$ whenever $(u,v) \in E(H)$ and $R_H(u,v) = 0$ otherwise. Thus, $\#_k$HomsToH is an instance of $\#_k\mathrm{CSP}(\{R_H\})$. The input of $\#_k\mathrm{CSP}(\{R_H\})$ contains a variable x_v for every vertex $v \in V(G)$ and a constraint $R_H(x_u, x_v)$ for every edge $(u,v) \in E(G)$. As can be observed from the construction, every valid homomorphism $\sigma : V(G) \to V(H)$ corresponds to an assignment of the variables $\{x_v\}_{v \in V(G)}$ satisfying every constraint in the CSP.

These results of Faben and of Guo et al. are incomparable to ours. We consider prime values of the modulo and a single binary relation, however the domain of our relations is arbitrarily large. Furthermore, the results of Faben [6, Theorem 4.11] show that the constraint language $\mathcal{F}$ for which $\#_2\mathrm{CSP}$ is tractable is richer than the constraint language for which $\#_k\mathrm{CSP}$ is tractable, where $k > 2$. In contrast, our results show that the dichotomy criterion of $\#_p$HomsToH remains the same for all primes p, when H is a tree.

1.3 Beyond one-bit functions

Weighted bipartite independent sets

To explain how we prove Theorem 1.5, consider a bipartite graph $G = (V_L, V_R, E)$ and let $\lambda_\ell = 0$ (the case $\lambda_r = 0$ is symmetric). We observe that every independent set I which contributes a non-zero summand to $Z_{\lambda_\ell,\lambda_r}(G)$ can only contain vertices in V_R ($Z_{\lambda_\ell,\lambda_r}(G)$ is defined in Problem 1.4). This yields the closed form $Z_{\lambda_\ell,\lambda_r}(G) = (\lambda_r + 1)^{|V_R|}$, which is computable in polynomial time. Regarding the case $\lambda_\ell, \lambda_r \not\equiv 0 \pmod{p}$, we employ a generalisation of a reduction used by Faben. In [6, Theorem 3.7] Faben reduces $\#_p\mathrm{SAT}$ to $\#_p\mathrm{BIS}_{1,1}$, the problem of counting independent sets of a bipartite graph.

We have to generalise this reduction for the weighted setting, in particular allowing different vertex weights for the vertices of each partition. Furthermore, during the construction we have to keep track of the assignment of vertices to their corresponding part, V_L or V_R. For this purpose we need to show the existence of bipartite graphs B, where $Z_{\lambda_\ell,\lambda_r}(B)$ takes specific values. These graphs are then used as gadgets in our reduction. In the unweighted setting $\#_p\mathrm{BIS}_{1,1}$ the graphs B are complete bipartite graphs. However, in the weighted setting $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$ complete bipartite graphs are not sufficient. Therefore, we prove the existence of the necessary bipartite gadgets B constructively. Key results appear in Section 2 and the technical proofs appear in Section 3 of the full version.

Pinning

Similar to the existing hardness proofs on modular counting graph homomorphisms we deploy a "pinning" technique. A partial function from a set X to a set Y is a function $f : X' \to Y$ for some $X' \subseteq X$. For any graph H, a *partially H-labelled graph* $J = (G, \tau)$ consists of an *underlying graph* G and a *pinning function* τ, which is a partial function from $V(G)$ to $V(H)$. A homomorphism from a partially labelled graph $J = (G, \tau)$ to H is a homomorphism $\sigma\colon G \to H$ such that, for all vertices $v \in \mathrm{dom}(\tau)$, $\sigma(v) = \tau(v)$. The resulting problem is denoted by $\#_p\textsc{PartLabHomsTo}H$, that is, given a prime p and graph H, compute $|\mathrm{Hom}\,(J \to H)| \pmod{p}$. In Section 3, we show that $\#_p\textsc{PartLabHomsTo}H$ reduces to $\#_p\textsc{HomsTo}H$. This allows us to establish hardness for $\#_p\textsc{HomsTo}H$, by proving hardness for $\#_p\textsc{PartLabHomsTo}H$. The reduction generalises the pinning reduction of Göbel, Goldberg and Richerby [10] from $\#_2\textsc{PartLabHomsTo}H$ to $\#_2\textsc{HomsTo}H$.

We explain how to prove pinning when we restrict the value of the modulo to 2 and the pinning function $\tau(J) = \{u \mapsto v\}$ to "pin" a single vertex. Given two graphs with distinguished vertices (G, u) and (H, v), let $\mathrm{Hom}\,((G, u) \to (H, v))$ be the set of homomorphisms from G to H mapping u to v. Given a graph with a distinguished vertex (G, u) and a graph H, we define $\mathbf{w}_H(G)$ to be the $\{0, 1\}$-vector containing the entries $|\mathrm{Hom}\,((G, u) \to (H, v))| \pmod{2}$ for each vertex $v \in V(H)$. Observe that for two vertices $v_1, v_2 \in V(H)$, such that $(H, v_1) \cong (H, v_2)$, and any graph G the relevant entries in $\mathbf{w}_G(H)$ will always be equal. Therefore, we can contract all such entries to obtain the *orbit vectors* $\mathbf{v}_H(G)$. Suppose that there exists a graph with a distinguished vertex (Θ, u_Θ), such that $\mathbf{v}_H(\Theta) = 0 \dots 010 \dots 0$, where the 1-entry corresponds to the vertex v of H. Given our input J for $\#_2\textsc{PartLabHomsTo}H$ we can now define an input G for $\#_2\textsc{HomsTo}H$, such that $|\mathrm{Hom}\,(J \to H)| \equiv |\mathrm{Hom}\,((G(J), u) \to (H, v))| \equiv |\mathrm{Hom}\,(G \to H)| \pmod{2}$. G contains a disjoint copy of $G(J)$ and Θ, where the vertices u and u_Θ are identified (recall that u is the vertex of J mapped by $\tau(J)$). Due to the value of $\mathbf{v}_H(\Theta)$ and the structure of G there is an even number of homomorphisms mapping u to any vertex $v' \neq v$, which establishes the claim.

Such a graph Θ, however, is not guaranteed to exist. Instead, we can define a set of operations on the vectors $\mathbf{v}_H$ corresponding to graph operations and show that for any vector in $\{0, 1\}^{|V(H)|}$ there exist a sequence of graphs with distinguished vertices $(\Theta_1, u_1), \dots, (\Theta_t, u_t)$ that "generate" this vector. Thus, there exists a set of graphs that "generate" $\mathbf{v} = 0 \dots 010 \dots 0$, which yields the desired reduction. This technique of [10] exploits the value of the modulo to be 2. Applying this technique to counting modulo any prime p directly, one can establish pinning for asymmetric graphs, that is graphs whose automorphism group contains only the identity. A dichotomy for $\#_p\textsc{HomsTo}H$, when H is an asymmetric tree appears in the first author's doctoral thesis [8].

In order to go beyond asymmetric graphs, one has to observe that information redundant only in the modulo 2 case is lost from the contraction of the vectors $\mathbf{w}_H$ to the vectors $\mathbf{v}_H$. This works on asymmetric graphs, since then these two vectors are identical. For general graphs we are able to restore pinning for counting homomorphisms modulo any prime p by utilising the non-contracted vectors $\mathbf{w}_H$.

▶ **Theorem 1.7.** *Let p be a prime and let H be a graph. Then $\#_p$*PartLabHomsTo*H reduces to $\#_p$*HomsTo*H via polynomial time Turing reduction.*

To obtain hardness for $\#_p$HomsToH we only need to pin two vertices when H is a tree, i.e. the domain of the pinning function τ has size two. For a study of a more general class of target graphs H (see [10]), the size of the domain has to be larger. As our pinning theorem applies to all primes p, all graphs H and pinning functions of arbitrary domain size, it can potentially be used to show hardness for $\#_p$HomsToH for all primes and any class of target graphs H. The key lemmas are presented in Section 3 and the formal proofs in Section 5 of the full version.

Gadgets

Gadgets are structures appearing in the target graph H that allow to reduce $\#_2$IS to $\#_2$PartLabHomsToH (the hardness of $\#_2$HomsToH is then immediate from Theorem 1.7). For illustrative purposes we simplify the definitions appearing in [10]. $\#_2$HomsToH–gadgets consist of two partially labelled graphs with distinguished vertices (J_1, y), (J_2, y, z) along with two "special" vertices $i, o \in V(H)$. Given the input G for $\#_2$IS, we construct an input G' for $\#_2$PartLabHomsToH as follows. We attach a copy of J_1 to every vertex u of G (identifying u with y) and replace every edge (u, v) of G with a copy of J_2 (identifying u with y and v with z). The properties of J_1 ensure that there is an odd number of homomorphisms from G' to H where the original vertices of G are mapped to i or o, while the number of the remaining homomorphisms cancels out. The properties of J_2 ensure that there is an even number of homomorphisms from G' to H when two adjacent vertices of G are both mapped to i, and an odd number of homomorphisms in every other case. We can now observe that $|\mathcal{I}(G)| \equiv |\mathrm{Hom}\,(G' \to H)\,| \pmod 2$, as the set of homomorphisms that does not cancel out must map every vertex of G to i or o and no pair of adjacent vertices both to i. Every vertex of G that is in an independent set must be mapped to i, and every vertex that is out of the independent set must be mapped to o.

Generalising the described approach to modulo any prime $p > 2$ one would end up reducing from a restricted $\#_p$CSP instance, containing a binary relation and a unary weight that must be applied to every variable of the instance (this is known as *external field* in statistical physics). Similar to the modulo 2 case the edge interaction is captured by the binary relation and size of the set of "special" vertices by the unary weights. Since for primes $p > 2$ there are more non-zero values than 1 (odd) a study of the external field is no longer trivial in this case. Instead we choose a different approach and reduce from $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$. This seems to capture the structure that produces hardness in $\#_p$HomsToH in a more natural way.

We formally present our reduction in Section 4. In the following we sketch our proof method and focus our attention on the example graph H in Figure 1. Let $G = (V_L, V_R, E)$ be a bipartite graph. Homomorphisms from G to H must respect the partition of G, i.e. the vertices in V_L can only be mapped to the vertices in $\{x_L, u_1, u_2, u_3\}$ and the vertices in V_R can only be mapped to the vertices in $\{x_R, v_1, v_2\}$, or vice versa. Any homomorphism σ from G to H, which maps the vertex $w \in V(G)$ to any vertex in $\{u_1, u_2, u_3\}$, must map every neighbour of w to x_R. Similarly, any homomorphism σ from G to H, which maps the vertex $w \in V(G)$ to any vertex in $\{v_1, v_2\}$, must map every neighbour of w to x_L. Thus, homomorphisms from G to H express independent sets of G: $\{u_1, u_2, u_3\}$ represent the vertices of V_L in the independent set and $\{v_1, v_2\}$ represent the vertices of V_R in the independent set, or vice versa. We construct a partially labelled graph J from G to fix the choice of V_L and V_R in the set of homomorphisms from G to H. $G(J)$ contains a copy of G, where every vertex in V_L is attached to the new vertex $\hat{u}$ and every vertex in V_R is attached to the new vertex $\hat{v}$. In addition, $\tau(J) = \{\hat{u} \mapsto x_R, \hat{v} \mapsto x_L\}$ is the pinning function. We observe that the vertices in V_L can only be mapped to vertices in $\{x_L, u_1, u_2, u_3\}$ and vertices in V_R can only be mapped to vertices in $\{x_R, v_1, v_2\}$. This observation yields that the number of homomorphisms from J to H is equivalent to $\sum_{I \in \mathcal{I}(G)} 3^{|V_L \cap I|} 2^{|V_R \cap I|} \pmod{p}$. Furthermore, the cardinality of the sets $\{u_1, u_2, u_3\}$ and $\{v_1, v_2\}$ introduces weights in a natural way.

For the reduction above we need the following property easily observable in H: there exist two adjacent vertices of degree $a = \lambda_\ell + 1 \not\equiv 1 \pmod{p}$ and $b = \lambda_r + 1 \not\equiv 1 \pmod{p}$. Recall that in order to obtain hardness for $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$ Theorem 1.5 requires $\lambda_\ell, \lambda_r \not\equiv 0 \pmod{p}$. In fact, as we will show in Section 4, these vertices need not be adjacent. During the construction of J we can replace the edges of G with paths of appropriate length. We call such a structure in H an (a, b, p)-path. In Lemma 4.4 we formally prove that if H has an (a, b, p)-path, then $\#_p\textsc{HomsTo}H$ is $\#_p$ P-hard. In particular, observe that stars cannot contain (a, b, p)-paths. Finally, we show that every non-star tree H contains an (a, b, p)-path, which yields our main result on $\#_p\textsc{HomsTo}H$ (Lemma 4.2).

1.4 Composites

We outline the obstacles occurring when extending the dichotomy for $\#_k\textsc{HomsTo}H$ to any integer k. Let H be a graph and let $k = \prod_{i=1}^{m} k_i$, where $k_i = p_i^{r_i}$ is an integer with its prime factorisation. Assuming $\#_k\textsc{HomsTo}H$ can be solved in polynomial time, then for each $i \in [m]$, $\#_{k_i}\textsc{HomsTo}H$ can also be solved in polynomial time. The reason is that k_i is a factor of k and we can apply the modulo k_i operator to the answer for the $\#_k\textsc{HomsTo}H$ instance. The Chinese remainder theorem shows that the converse is also true: if for each $i \in [m]$ we can solve $\#_{k_i}\textsc{HomsTo}H$ in polynomial time, then we can also solve $\#_k\textsc{HomsTo}H$ in polynomial time. By the previous observations we can now focus on powers of primes $k = p^r$. Assuming $\#_k\textsc{HomsTo}H$ is computable in polynomial time yields again that $\#_p\textsc{HomsTo}H$ is also computable in polynomial time. However, the converse is not always true.

Guo et al. [13] were able to obtain this reverse implication for the constraint satisfaction problem. They showed [13, Lemma 4.1 and Lemma 4.3] that when p is a prime $\#_{p^r}\mathrm{CSP}$ is computable in polynomial time if $\#_p\mathrm{CSP}$ is computable in polynomial time. In the full version we show that their technique cannot be transferred to the $\#_k\textsc{HomsTo}H$ setting. We show that there is a graph (P_4) such that $\#_2\textsc{HomsTo}P_4$ is computable in polynomial time, while $\#_4\textsc{HomsTo}P_4$ is $\#_2$ P-hard.

2 Weighted bipartite independent set

We study the complexity of Problem 1.4, the problem of computing the weighted sum over independent sets in a bipartite graph. We begin by identifying the tractable instances.

▶ **Proposition 2.1.** *If $\lambda_\ell \equiv 0 \pmod p$ or $\lambda_r \equiv 0 \pmod p$ then $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$ is computable in polynomial time.*

To prove the hardness of the remaining cases, we reduce from $\#_p\mathrm{SAT}$. Our reduction starts with a Boolean formula φ and constructs, in two stages, a graph G_φ, such that $Z_{\lambda_\ell,\lambda_r}(G_\varphi) \equiv K|\operatorname{sat}(\varphi)| \pmod p$, where $\operatorname{sat}(\varphi)$ denotes the set of satisfying assignments of φ and K is a constant depending on the values of the weights λ_ℓ and λ_r. In the first stage we define the graph G'_φ. For every variable x_i in φ, G'_φ contains three vertices u_i, $\bar{u}_i$ and w_i to the left vertex set $V_L(G'_\varphi)$ as well as three vertices v_i, $\bar{v}_i$ and z_i to the right vertex set $V_R(G'_\varphi)$. For every clause c_j of φ, G'_φ further contains a vertex y_j in the right vertex set $V_R(G'_\varphi)$. We further introduce the edges forming the cycle $u_i v_i w_i \bar{v}_i \bar{u}_i z_i u_i$ to $E(G'_\varphi)$ for every variable x_i in φ. Additionally for all $i \in [n]$, if x_i appears as a literal in clause c_j of φ, we introduce the edge (u_i, y_j) in G'_φ and if $\bar{x}_i$ appears as a literal in clause c_j, we introduce the edge $(\bar{u}_i, y_j)$ in G'_φ. The left part of Figure 2 illustrates an example of this construction.

The second stage uses copies of a bipartite graph B, which is obtained by the following key lemma.

▶ **Lemma 2.2.** *Let p be a prime and $\lambda_\ell, \lambda_r \in \mathbb{Z}_p^*$. There exists a bipartite graph $B = (V_L, V_R, E)$ with distinguished vertices $u_L \in V_L$ and $v_R \in V_R$, that satisfies*

1. $Z_{\lambda_\ell,\lambda_r}(B) \equiv 0 \pmod p$,
2. $Z_{\lambda_\ell,\lambda_r}(B - u_L) \not\equiv 0 \pmod p$,
3. $Z_{\lambda_\ell,\lambda_r}(B - v_R) \not\equiv 0 \pmod p$.

In the second and final stage, we construct the graph G_φ. Let (B, u_L, v_R) be the graph obtained from Lemma 2.2. G_φ is a copy of G'_φ together with two copies of B for every variable of φ and one copy of B for every clause. The first n copies $B^1, \dots, B^n$ are connected to G'_φ by identifying the distinguished vertex u_L^i in the left component with $w_i \in V_L(G'_\varphi)$ for all $i \in [n]$. The second n copies $B^{n+1}, \dots, B^{2n}$ are connected to G'_φ by identifying the distinguished vertex v_R^{n+i} in their right components with $z_i \in V_R(G'_\varphi)$ for all $i \in [n]$. The remaining m copies $B^{2n+1}, \dots, B^{2n+m}$ of B are connected to G'_φ by identifying the distinguished vertex v_R^{2n+j} in their right components with $y_j \in V_R(G'_\varphi)$ for all $j \in [m]$. For an example see the right part of Figure 2.

From this construction we obtain the desired result.

▶ **Theorem 1.5.** *Let p be a prime and let λ_ℓ, $\lambda_r \in \mathbb{Z}_p$. If $\lambda_\ell \equiv 0 \pmod p$ or $\lambda_r \equiv 0 \pmod p$ then $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$ is computable in polynomial time. Otherwise, $\#_p\mathrm{BIS}_{\lambda_\ell,\lambda_r}$ is $\#_p\mathsf{P}$-complete.*

3 Homomorphisms of partially labelled graphs

We study the following problem.

▶ **Problem 3.1.** $\#_p$PartLabHomsToH.

Parameter.	Graph H and prime p.
Input.	Partially H-labelled graph $J = (G, \tau)$.
Output.	$\lvert\mathrm{Hom}(J \to H)\rvert \pmod p$.

We observe the following theorem for all primes p.

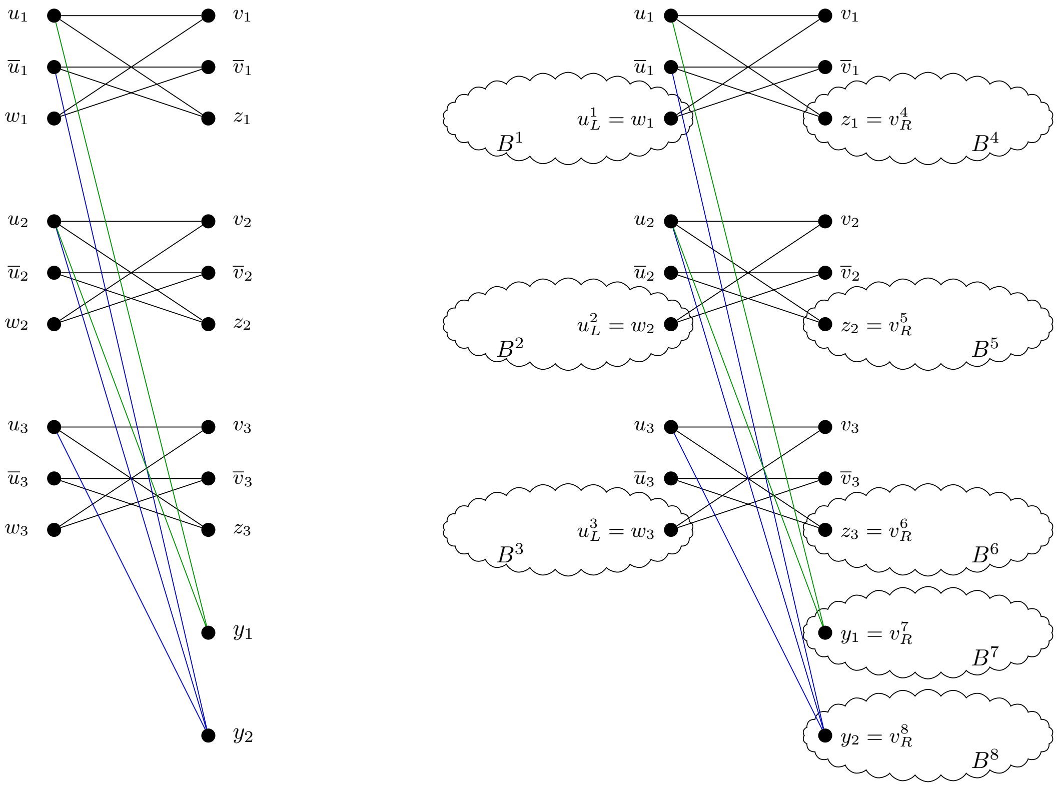

Figure 2 The graphs G'_φ and G_φ for $\varphi = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_2 \vee x_3)$.

▶ **Lemma 3.2** (Göbel, Goldberg and Richerby [10]). *Let p be a prime and let $(H, \bar{v})$ and $(H', \bar{v}')$ be graphs having no automorphism of order p, each with r distinguished vertices. Then $(H, \bar{v}) \cong (H', \bar{v}')$ if and only if, for all (not necessarily connected) graphs $(G, \bar{u})$ with r distinguished vertices,*

$$|\operatorname{Hom}((G, \bar{u}) \to (H, \bar{v}))| \equiv |\operatorname{Hom}((G, \bar{u}) \to (H', \bar{v}'))| \pmod{p}.$$

Instead of orbit vectors, which are used in the pinning proof of [10], we employ tuple vectors. The tuple vectors include the sizes of the orbits $\mathrm{Orb}_H(\bar{v})$, for all $v \in V(H)^r$, and this information is vital for the proof of our pinning theorem.

▶ **Definition 3.3.** Let H be a graph with no automorphism of order p, $r \in \mathbb{Z}_{>0}$ and let $\bar{w}_1, \ldots, \bar{w}_\nu$ be an enumeration of $(V(H))^r$, i.e., $\nu = |V(H)|^r$. Let $(G, \bar{u})$ be a graph with r distinguished vertices. We define the *tuple vector* $\mathbf{w}_H(G, \bar{u}) \in (\mathbb{Z}_p)^\nu$ where, for each $j \in [\nu]$, the j-th component of $\mathbf{w}_H(G, \bar{u})$ is given by

$$\left(\mathbf{w}_H(G, \bar{u})\right)_j \equiv |\operatorname{Hom}((G, \bar{u}) \to (H, \bar{w}_j))| \pmod{p}.$$

We say that $(G, \bar{u})$ *implements* this vector.

Not all tuple vectors in $(\mathbb{Z}_p)^\nu$ are implementable. We only require the following set to be implementable.

▶ **Definition 3.4.** Let H be a graph with no automorphism of order p, $r \in \mathbb{Z}_{>0}$ and let $\bar{w}_1, \ldots, \bar{w}_\nu$ be an enumeration of $(V(H))^r$, i.e., $\nu = |V(H)|^r$. Denote by $F(H, r) \subseteq (\mathbb{Z}_p)^\nu$ the set of vectors $\mathbf{w}$, such that, for all $i, j \in [\nu]$ with $(H, \bar{w}_i) \cong (H, \bar{w}_j)$, we have $(\mathbf{w})_i = (\mathbf{w})_j$.

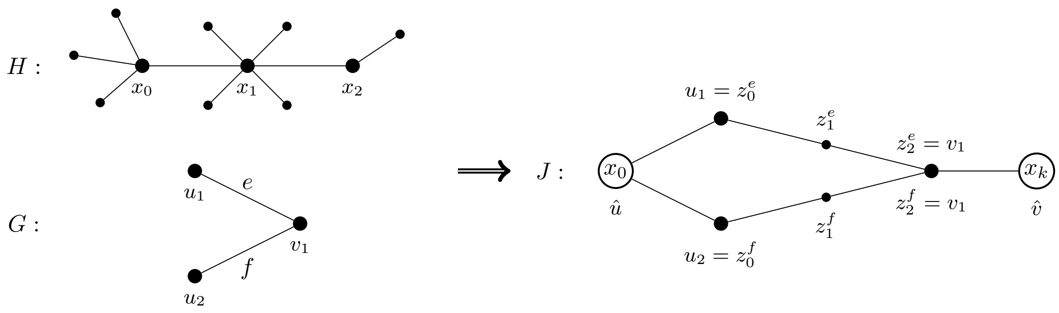

Figure 3 An instance for $p = 5$ of $\#_p\text{BIS}_{3,1}$ reducing to $\#_p$PartLabHomsToH. H contains the $(4, 2, 5)$-path $x_0x_1x_2$. G is transformed to the partially labelled graph J, where the mappings of $\tau(J)$ are shown as vertices encircling the target of the mapping.

▶ **Lemma 3.5.** *Let H be a graph with no automorphism of order p, $r \in \mathbb{Z}_{>0}$ and $\bar{w}_1, \ldots, \bar{w}_\nu$ an enumeration of $(V(H))^r$, i.e., $\nu = |V(H)|^r$. Then every $\mathbf{w} \in F(H, r)$ is H-implementable.*

Using the above in the full version we prove the following theorem.

▶ **Theorem 1.7.** *Let p be a prime and let H be a graph. Then $\#_p$PartLabHomsToH reduces to $\#_p$HomsToH via polynomial-time Turing reduction.*

4 Hardness for trees

The structure in H that yields hardness for $\#_p$HomsToH is formally defined as follows.

▶ **Definition 4.1.** Let H be a graph, p be a prime and $a, b \in \mathbb{Z}_p \setminus \{1\}$. Assume H contains a path $P = x_0 \ldots x_k$ for $k > 0$, such that the following hold

1. P is the unique path between x_0 and x_k in H.
2. $\deg_H(x_0) \equiv a \pmod{p}$ and $\deg_H(x_k) \equiv b \pmod{p}$.
3. For all $0 < i < k$, $\deg_H(x_i) \equiv 1 \pmod{p}$.

Then, we will call P an *(a, b, p)-path in H* and denote it Q_H.

▶ **Lemma 4.2.** *Let H be a tree that has no automorphism of order p. Then, either H is a star or there are $a, b \in \mathbb{Z}_p \setminus \{1\}$ such that H contains an (a, b, p)-path.*

In order to show that $\#_p$HomsToH is $\#_p$P-hard we are going to establish a reduction from $\#_p\text{BIS}_{\lambda_\ell,\lambda_r}$ to $\#_p$PartLabHomsToH. Let p be a prime and let H be a tree, target graph in $\#_p$PartLabHomsToH. Given a graph $G = (V_L, V_R, E)$ input for $\#_p\text{BIS}_{\lambda_\ell,\lambda_r}$, we construct a partially labelled graph J, input for $\#_p$PartLabHomsToH, such that $Z_{\lambda_\ell,\lambda_r}(G) \equiv |\operatorname{Hom}(J \to H)| \pmod{p}$. Assume H contains an (a, b, p)-path $Q = x_0 \ldots x_k$ and let $P_k = z_0 \ldots z_k$ be the k-path of length k. For every edge $e \in E$, we take a copy of P_k denoted $P_k^e = z_0^e \ldots z_k^e$. Then, J is constructed starting with G by adding two vertices $\hat{u}$ and $\hat{v}$ and connecting them to every vertex in V_L and V_R, respectively. Subsequently, every edge $e \in E$ is substituted with a the path P_k^e. Finally, the pinning function of J maps $\hat{u}$ to x_0 as well as $\hat{v}$ to x_k. See Figure 3 for an example.

The following lemma gives the key properties of J, which establish the reduction.

▶ **Lemma 4.3.** *Let p be a prime, $G = (V_L, V_R, E)$ a bipartite graph and H be a tree. Assume there are $a, b \in \mathbb{Z}_p \setminus \{1\}$ such that H contains an (a, b, p)-path $Q_H = x_0 \ldots x_k$. We denote the diminished neighbourhoods of x_0 and x_k by $W_L = \Gamma_H(x_0) - x_1$ and $W_R = \Gamma_H(x_k) - x_{k-1}$,* *respectively. Additionally, let J be the partially labelled graph described above. Then, for every homomorphism σ from J to H the following hold.*

1. *Let $u \in V_L$ and $v \in V_R$, then $\sigma(u) \in \Gamma_H(x_0)$ and $\sigma(v) \in \Gamma_H(x_k)$, respectively;*
2. *Let $\mathfrak{O}_\sigma = \{u \in V_L \mid \sigma(u) = x_1\} \cup \{v \in V_R \mid \sigma(v) = x_{k-1}\}$ and $\mathfrak{I}_\sigma = (V_L \cup V_R) \setminus \mathfrak{O}_\sigma$. Given another homomorphism σ' from J to H, the relation $\sigma \sim_{\mathfrak{I}} \sigma'$ if $\mathfrak{I}_\sigma = \mathfrak{I}_{\sigma'}$ is an equivalence relation with equivalence class denoted $[\![\cdot]\!]_{\mathfrak{I}}$;*
3. *Let $\sigma_1, \ldots, \sigma_\mu$ be representatives from each $\sim_{\mathfrak{I}}$-equivalence class. Then, the set $\mathcal{I}(G)$ of independent sets of G is exactly the set $\{\, \mathfrak{I}_{\sigma_i} \mid i \in [\mu] \,\}$.*
4. *For the diminished neighbourhoods holds $|[\![\sigma]\!]_{\mathfrak{I}}| \equiv |W_L|^{|\mathfrak{I}_\sigma \cap V_L|} |W_R|^{|\mathfrak{I}_\sigma \cap V_R|} \pmod{p}$.*

▶ **Lemma 4.4.** *Let p be a prime and let H be a graph with no automorphism of order p. If there are $a, b \in \mathbb{Z}_p \setminus \{1\}$ such that H has an (a, b, p)-path Q_H then $\#_p$*HomsTo*H is $\#_p\,\mathsf{P}$-hard under Turing reductions.*

5 Dichotomy theorems

The results of Faben and Jerrum [7] combined with Lemma 4.4 give the following dichotomy theorem.

▶ **Theorem 1.2.** *Let p be a prime and let H be a graph, such that its order p reduced form H^{*p} is a tree. If H^{*p} is a star, then $\#_p$*HomsTo*H is computable in polynomial time; otherwise, $\#_p$*HomsTo*H is $\#_p\,\mathsf{P}$-complete.*

To justify our title, we use the following proposition showing that our dichotomy theorem holds for all trees. In [7, Section 5.3] this was stated as an obvious fact, however for the sake of completeness we provide a formal proof.

▶ **Proposition 5.1.** *Let H be a tree and ϱ an automorphism of H. Then the subgraph H^ϱ of H induced by the fixed points of ϱ is also a tree.*

The claim implies that if H is a tree, then its order p reduced form H^{*p} is also a tree. This yields the following corollary.

▶ **Corollary 1.3.** *Let p be a prime and let H be a tree. If the order p reduced form H^{*p} of H is a star, then $\#_p$*HomsTo*H is computable in polynomial time; otherwise, $\#_p$*HomsTo*H is $\#_p\,\mathsf{P}$-complete.*

To deal with disconnected graphs, Faben and Jerrum [7, Theorem 6.1] show the following theorem.

▶ **Theorem 5.2** (Faben and Jerrum). *Let H be a graph that has no automorphism of order 2. If H' is a connected component of H and $\#_2$*HomsTo*H' is $\#_2\,\mathsf{P}$-hard, then $\#_2$*HomsTo*H is $\#_2\,\mathsf{P}$-hard.*

The only part where the value 2 of the modulo is required, is the application of their pinning theorem [7, Theorem 4.7]. Since we have already shown the more general Theorem 1.7, we conclude that the theorem holds in the following form.

▶ **Theorem 5.3.** *Let p be a prime and let H be a graph that has no automorphism of order p. If H_1 is a connected component of H and $\#_p$*HomsTo*H_1 $\#_p\,\mathsf{P}$-hard, then $\#_p$*HomsTo*H is $\#_p\,\mathsf{P}$-hard.*

The latter strengthens Theorem 1.2 to the following version.

▶ **Theorem 5.4.** *Let H be a graph whose order p reduced form H^{*p} is a forest. If every component of H^{*p} is a star, $\#_p$HomsToH is computable in polynomial time, otherwise $\#_p$HomsToH is $\#_p$P-complete.*

A discussion of our results was already conducted in the introduction. Again we refer the curious reader to the full version of the paper, available at `https://arxiv.org/abs/1802.06103`.

References

1 A. A. Bulatov and M. Grohe. The complexity of partition functions. *Theoretical Computer Science*, 348(2-3):148–186, 2005. doi:10.1016/j.tcs.2005.09.011.

2 J.-Y. Cai, X. Chen, and P. Lu. Graph homomorphisms with complex values: A dichotomy theorem. *SIAM Journal on Computing*, 42(3):924–1029, 2013. doi:10.1137/110840194.

3 J.-Y. Cai and P. Lu. Holographic algorithms: From art to science. *Journal of Computer and System Sciences*, 77(1):41–61, 2011.

4 M. E. Dyer, A. M. Frieze, and M. Jerrum. On counting independent sets in sparse graphs. *SIAM Journal on Computing*, 31(5):1527–1541, 2002. doi:10.1137/S0097539701383844.

5 M. E. Dyer and C. S. Greenhill. The complexity of counting graph homomorphisms. *Random Structures and Algorithms*, 17(3-4):260–289, 2000.

6 J. Faben. The complexity of counting solutions to generalised satisfiability problems modulo k. *arXiv*, abs/0809.1836, 2008.

7 J. Faben and M. Jerrum. The complexity of parity graph homomorphism: an initial investigation. *Theory of Computing*, 11:35–57, 2015.

8 A. Göbel (A. Gkompel-Magkakis). *Counting, Modular Counting and Graph Homomorphisms*. PhD thesis, University of Oxford, 2016.

9 A. Göbel, L. A. Goldberg, and D. Richerby. The complexity of counting homomorphisms to cactus graphs modulo 2. *ACM Transactions on Computation Theory*, 6(4):17:1–17:29, 2014.

10 A. Göbel, L. A. Goldberg, and D. Richerby. Counting homomorphisms to square-free graphs, modulo 2. *ACM Transactions on Computation Theory,*, 8(3):12:1–12:29, 2016.

11 L. A. Goldberg, M. Grohe, M. Jerrum, and M. Thurley. A complexity dichotomy for partition functions with mixed signs. *SIAM Journal on Computing*, 39(7):3336–3402, 2010.

12 L. M. Goldschlager and I. Parberry. On the construction of parallel computers from various bases of Boolean functions. *Theoretical Computer Science*, 43:43–58, 1986.

13 H. Guo, S. Huang, P. Lu, and M. Xia. The complexity of weighted boolean #CSP modulo k. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 249–260, 2011.

14 P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990.

15 R. E. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, 22(1):155–171, 1975.

16 C. H. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Proceedings of the GI-Conference on Theoretical Computer Science*, pages 269–276, 1982.

17 S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.

18 L. G. Valiant. Accidental algorthims. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 509–517, 2006.