THEORY OF COMPUTING, Volume 11 (2), 2015, pp. 35–57
www.theoryofcomputing.org

The Complexity of Parity Graph Homomorphism: An Initial Investigation

John Faben* Mark Jerrum†

Received November 28, 2013; Revised March 2, 2015; Published March 14, 2015

Abstract: Given a graph G, we investigate the problem of determining the parity of the number of homomorphisms from G to some other fixed graph H. We conjecture that this problem exhibits a complexity dichotomy, such that all parity graph homomorphism problems are either polynomial-time solvable or $\oplus$P–complete, and provide a conjectured characterisation of the easy cases.

We show that the conjecture is true for the restricted case in which the graph H is a tree, and provide some tools that may be useful in further investigation into the parity graph homomorphism problem, and the problem of counting homomorphisms for other moduli.

ACM Classification: F.1.3, G.2.2

AMS Classification: 05C30, 05C60, 68Q17, 68Q25

Key words and phrases: complexity theory, graph homomorphisms, modular counting, dichotomy theorem

1 Graph homomorphism

Graph homomorphism is a natural generalisation of graph colouring, in which the restrictions on adjacencies between colours can be more general than in the usual graph colouring problem. A homomorphism from a graph G to a graph H is an edge-preserving map between the vertices (see Definition 1.1). It is sometimes referred to as an H-colouring (where the target graph for the homomorphism is H). Ordinary graph colouring is the special case of homomorphisms into the complete graph.

*Supported by EPSRC grant EP/E064906/1 “The Complexity of Counting in Constraint Satisfaction Problems”.
†Supported by EPSRC grant EP/I011935/1 “Computational Counting”.

 DOI: 10.4086/toc.2015.v011a002

Definition 1.1. A homomorphism from a graph G into another graph H is a map $\varphi : V(G) \to V(H)$ having the property that $(\varphi(u),\varphi(v)) \in E(H)$ whenever $(u,v) \in E(G)$. The set of homomorphisms from G to H is denoted by $\mathrm{Hom}(G,H)$, and the number of homomorphisms by $\hom(G,H)$.

Example 1.2. A homomorphism from a graph G to the complete graph K_n is a (proper, vertex) n-colouring of G.

Example 1.3. Let H_1 be the graph with vertex set $\{a,b\}$, an edge joining a and b, and a loop at b. A homomorphism from a graph G to H_1 can be considered as an independent set of G. The vertices mapped to vertex a form an independent set (as none of them can be pairwise adjacent) and, conversely, given an independent set, it is possible to map the vertices of the independent set to a and the vertices of its complement to b. So there is a natural one-to-one correspondence between homomorphisms to H_1 and independent sets.

For the purposes of this paper, both G and H are allowed to have loops on their vertices, but not multiple edges. To reduce the potential for confusion, we will usually refer to the vertices of H as "colours", reserving the word "vertex" for vertices of G.

Fix a target graph H. There are a number of computational problems of the form: given an instance (graph) G return some information about $\mathrm{Hom}(G,H)$. The most basic one is the decision problem, which asks if $\mathrm{Hom}(G,H)$ is non-empty. Each H specifies a particular decision problem; for example, if H is the triangle, the problem is to decide if G is 3-colourable. The goal is then to classify the complexity of the computational problem in terms of the graph H. The ideal is to identify a dichotomy, i. e., a partition of graphs H into those that specify tractable problems and those that specify intractable ones.

The complexity of the decision version of the graph homomorphism problem was completely classified by Hell and Nešetřil in [14]. For a given graph H, deciding whether an arbitrary graph has a homomorphism to H can be done in polynomial time if H has a loop or is bipartite. Hell and Nešetřil showed that this decision problem is NP-complete in all other cases.

It is also natural to consider the counting problem, which asks for the cardinality $\hom(G,H)$ of $\mathrm{Hom}(G,H)$. The problem of exactly counting the homomorphisms to a fixed graph H was considered by Dyer and Greenhill [6], who gave a complete characterisation, again with a dichotomy theorem: the counting problem is polynomial-time solvable if H is either a complete graph with loops everywhere or a complete bipartite graph without loops, and it is #P-complete otherwise.

The result of Dyer and Greenhill has been extended in many different directions by various authors. One possibility is to specify weights $w : E(H) \to \mathbb{C}$ for the edges of H; this edge-weighting naturally induces a weighing of homomorphisms φ from G to H, by taking a product of weights $w(\varphi(u),\varphi(v))$ over edges $\{u,v\}$ of G. In the weighted setting, one can express partition functions of models in statistical physics. Note that the unweighted form of the problem can be recovered by restricting weights to be $\{0,1\}$. Bulatov and Grohe [2] exhibited a dichotomy for non-negative real weights, which was extended to arbitrary real weights by Goldberg, Grohe, Jerrum and Thurley [12], and then on to complex weights by Cai, Chen and Lu [4]. The massive further generalisation to Constraint Satisfaction Problems (CSPs) was undertaken by several authors (e. g., Bulatov [1] and Dyer and Richerby [7]), culminating in the complex weighted case by Cai and Chen [3]. See Chen's survey for more details [5].

In this paper, we shall mostly be concerned with the problem of determining the cardinality of $\mathrm{Hom}(G,H)$ modulo k, for a positive integer k, with a special emphasis on $k = 2$, i. e., determining whether the number of H-colourings is odd or even. For $k \geq 2$ and n an integer, denote by $[n]_k$ the residue class of n *modulo* k. We can of course identify these classes with the integers $\{0, 1, \ldots, k-1\}$. Formally, our computational problem is the following.

Name. $\#_k H$-COLOURING.

Instance. An undirected graph G.

Output. $[\hom(G,H)]_k$, i. e., the number of H-colourings of G *modulo* k.

Since the case $k = 2$ is of special significance, we introduce $\oplus H$-COLOURING as a synonym for $\#_2 H$-COLOURING.

We give a dichotomy theorem for $\oplus H$-COLOURING in the case where H is a tree: either $\oplus H$-COLOURING is $\oplus\mathsf{P}$-complete or it can be solved in polynomial time. (See Theorem 3.8.) Informally, $\oplus\mathsf{P}$ is the class of problems that can be expressed in terms of deciding the parity of the number of accepting computations of a non-deterministic Turing machine; see Section 2 for a precise definition. The proof of the dichotomy is based on a reduction system which transforms H to a "reduced form" of equivalent complexity. Since it is easy to decide the complexity of $\oplus H$-COLOURING for reduced forms, we obtain not only the dichotomy result, but also an effective procedure for deciding the dichotomy. We conjecture that the same reduction system decribes a complexity dichotomy for general graphs. Although this conjecture remains open in general, Göbel, Goldberg and Richerby have extended our result by showing that the conjecture holds for cactus graphs [10] and square-free graphs [11].

Finally we draw attention to some existing work in the general area of modular counting. The complexity of modular counting problems has been studied for at least three decades, early contributions being made by Valiant [21] and Papadimitriou and Zachos [19]. One of the more striking results, is that of Valiant [22], who provides an example of a counting problem that is unexpectedly easy *modulo* 7, though hard *modulo* 2. It is worth noting that modular CSPs have been studied, e. g., by Faben [8] and Guo, Huang, Lu and Xia [13]. This work is both more general, in the sense of being set within the wider context of CSPs, but also more restrictive, in that it relates to the two-element (Boolean) domain only.

A preliminary version of the results presented here appeared in the first author's Ph. D. thesis [9].

2 Modular counting complexity

2.1 The classes $\#_k\mathsf{P}$

In this section, we formally define the counting classes that we will use in this paper.

A classical counting problem can be considered as a function taking a problem instance to the number of solutions associated with that instance. When counting is done *modulo* some number $k \geq 2$, it is possible to view the problem from two somewhat different standpoints. On the one hand there is the decision or language view, where the task is to determine whether the number of solutions is different from 0, *modulo* k. On the other is the function view, where the task is to compute the residue, modulo k, of the number of solutions. Both views have been taken in earlier work, and the distinctions between them have been examined by Faben [9].

In the current context, the function view seems more natural. We work within a class $\#_k\mathsf{P}$ of computational problems which is the modular analogue of the classical class $\#\mathsf{P}$ of counting problems.

Informally, $\#_k\mathsf{P}$ contains functions that can be expressed as the residue, *modulo k*, of the number of accepting computations of a nondeterministic polynomial-time Turing machine.

Let Σ be a finite alphabet over which we agree to encode problem instances, and M a non-deterministic Turing Machine with input alphabet Σ. Denote by $\#\mathrm{acc}_M(x)$ the number of accepting paths of the machine M on the input $x \in \Sigma^*$.

Definition 2.1. The class $\#\mathsf{P}$ consists of all functions $f : \Sigma^* \to \mathbb{N}$ that can be expressed as $f(x) = \#\mathrm{acc}_M(x)$ for some non-deterministic polynomial-time Turing Machine M. The class $\#_k\mathsf{P}$ consists of all functions $f : \Sigma^* \to \{0,1,\dots,k-1\}$ that can be expressed as $f(x) = [\#\mathrm{acc}_M(x)]_k$.

In this paper, we are concerned particularly with the case $k = 2$, and we follow other authors in using $\oplus\mathsf{P}$ as a synonym for $\#_2\mathsf{P}$ [19].

Given a counting problem in $\#\mathsf{P}$, say $\#A$, we write $\#_kA$ for the $\#_k\mathsf{P}$ problem of determining the number of solutions to A *modulo k*. So while $\#A : \Sigma^* \to \mathbb{N}$ is a function defined from strings to the natural numbers, $\#_kA : \Sigma^* \to \{0,\dots,k-1\}$ is the function from strings to the integers *modulo k* defined by $\#_kA(x) \equiv \#A(x) \pmod{k}$. As an example, $\#_k$SAT is the problem of determining the number of satisfying assignments to a CNF Boolean formula, *modulo k*. Naturally, $\oplus$SAT is the special case $k = 2$ of this problem.

2.2 Completeness

Again, in an analogy with $\#\mathsf{P}$-completeness, we define the notion of $\#_k\mathsf{P}$-completeness with respect to polynomial-time Turing reducibility (also known as Cook reducibility). Essentially, a problem A is $\#_k\mathsf{P}$-hard if every problem in $\#_k\mathsf{P}$ can be solved in polynomial time given an oracle for A.

Definition 2.2. We say that a problem B is *polynomial-time Turing reducible* to a problem A if problem B can be solved in polynomial time using an oracle for problem A. We write $B \leq_\mathrm{p}^\mathrm{T} A$.

Definition 2.3. A counting problem A is $\#_k\mathsf{P}$-*hard* if, for every problem B in $\#_k\mathsf{P}$, it is the case that $B \leq_\mathrm{p}^\mathrm{T} A$. It is $\#_k\mathsf{P}$-*complete* if, in addition, A is in $\#_k\mathsf{P}$.

As one might expect, the modular counting versions of SAT, namely $\#_k$SAT for $k \geq 2$, are examples of $\#_k\mathsf{P}$-complete problems for all k. This can easily be seen, as the usual reduction in Cook's Theorem, showing that SAT is NP-complete, is parsimonious (i. e., preserves the number of solutions), and so certainly preserves the number of solutions *modulo k* for all k.

As mentioned above, the complexity of exactly counting the homomorphisms to a given graph H was characterised by Dyer and Greenhill. They proved the following theorem.

Theorem 2.4 (Dyer and Greenhill [6])**.** *If a graph H is a complete bipartite graph with no loops or a complete graph with loops everywhere, then exactly counting H-colourings can be done in polynomial time. Otherwise, the problem is $\#\mathsf{P}$-complete.*

Clearly, if the number of homomorphisms to a graph H can be counted exactly in polynomial time, then the parity can be determined in polynomial time. We will show that there are some cases in which symmetries of H can make the related modular counting problem easy, even when the exact counting problem is $\#\mathsf{P}$-hard.

3 A confluent reduction system

As hinted at earlier, our approach is based on a reduction system for graphs H that preserves the complexity of the problem $\#_pH$-COLOURING. The reductions are defined in terms of the automorphisms of H.

3.1 Reduction by automorphisms

Definition 3.1. An *automorphism* of a graph G is an injective homomorphism from G to itself. In other words, an automorphism of a graph G is a permutation σ of the vertices of G such that

$$\{\sigma(u),\sigma(v)\} \in E(G) \iff \{u,v\} \in E(G)\,.$$

If σ has order 2, i. e., σ is not the identity but $\sigma \circ \sigma$ is, then we say that σ is an *involution* of G.

Definition 3.2. Let H be a graph, and σ an automorphism of H. We denote by H^σ the subgraph of H induced by the fixed points of σ.

Lemma 3.3. *If H is a graph, and σ an involution of H, the number of H-colourings of any graph G is congruent modulo 2 to the number of H^σ-colourings of G.*

Proof. We will in fact show that the number of H-colourings of G that are not H^σ-colourings is even, which is equivalent to saying that the number of H-colourings that use at least one colour in $V(H)\setminus V(H^\sigma)$ is even.

To see this, we partition the set of such colourings into subsets of size two. The basic idea here is that to each colouring that uses at least one colour in $V(H)\setminus V(H^\sigma)$ we can associate the colouring gained by first applying σ to H and then colouring G. Formally, given any colouring $\varphi : V(G) \to V(H)$, consider the alternative colouring $\sigma \circ \varphi$. This is still an H-colouring of G, as both σ and φ are edge-preserving. It is different from φ as there is some vertex $v \in G$ such that $\varphi(v) \in V(H)\setminus V(H^\sigma)$, and so $\sigma(\varphi(v)) \neq \varphi(v)$. On the other hand $\sigma \circ \sigma \circ \varphi$ is just φ, as σ is an involution. So σ acts as an involution on the set of H-colourings of G that use at least one colour from $V(H)\setminus V(H^\sigma)$. Since this involution has no fixed points, the size of this set must be even. □

Note that the above argument does not rely on any special properties of the modulus 2 beyond the fact that it is prime.

Theorem 3.4. *For any prime p, if H is a graph, and σ an automorphism of H of order p, the number of H-colourings of any graph G is congruent modulo p to the number of H^σ-colourings of G.*

It is not just the proof that fails for a composite modulus k. The complete graph K_5 on five vertices has an automorphism of order 6 that moves all the vertices, but it is not true that for every graph G the number of 5-colourings of G is divisible by 6.

We define the following reduction system on the set of unlabelled graphs.

Definition 3.5. The binary relation $\to_k$ on graphs is defined as follows. For graphs H and K, the relation $H \to_k K$ holds iff there exists an automorphism σ of H, of order k, such that $H^\sigma = K$. If there exists a sequence of graphs $H_1, H_2, \ldots, H_\ell$ such that $H \to_k H_1 \to_k H_2 \to_k \cdots \to_k H_\ell = K$, we write $H \to_k^* K$

and say that H *reduces to K by automorphisms of order k*. (If $k=2$, we say that H *reduces to K by involutions*.) If K has no automorphisms of order k we say that K is a *reduced form* associated with the graph H.

Example 3.6. In Figure 1 we give an example of a graph H, along with two ways of reducing H by involutions. On the right-hand side we reduce H by using the involution σ that swaps each of the pairs of vertices a and e, b and f, c and d, leaving behind only the involution-free graph on the vertices g and h. On the left-hand side, we begin with the involution τ that swaps e and f, and have to reduce the resulting graph by involutions twice more before we get to the involution-free graph $((H^\tau)^\upsilon)^\eta$, which is isomorphic to the graph H^σ. This is not a coincidence. We will see in Theorem 3.7 that reduced forms are unique.

To make further progress, we need to assume $k=p$ is prime. Eventually, we will further restrict attention to the case $p=2$. However, we state and prove some intermediate results for a general prime p, as they may be of use in further explorations of modular counting problems.

Theorem 3.4 says that in classifying the complexity of $\#_pH$-COLOURING problems, it is enough to restrict attention to graphs H that are reduced forms, i. e., that do not have any automorphisms of order p. This is enough for the proof of the main dichotomy result, but it is an interesting fact that reduced forms are unique. In any case, the concepts used in the proof of uniqueness of the reduced form will be needed later.

Theorem 3.7. *Given a graph G, and a prime p there is (up to isomorphism) exactly one graph G^* such that G^* has no automorphisms of order p and $G \to_p^* G^*$.*

The proof is deferred to the next section. We can now state main result.

Theorem 3.8. *If H is a tree, then $\oplus H$-COLOURING is $\oplus$P-complete if the reduced form obtained by reducing H by involutions is non-trivial, i. e., has more than one vertex. Otherwise it is solvable in polynomial time.*

We conjecture that this result holds for graphs in general. The conjecture is unresolved, though Göbel, Goldberg and Richerby [10] recently extended our result from trees to cactus graphs. One could extend the conjecture to $\#_pH$-COLOURING, for primes $p>2$. Specifically, one might conjecture that, for each p, the set of reduced forms H corresponding to polynomial-time cases of $\#_pH$-COLOURING is finite (and that all other reduced forms correspond to $\#_p$P-complete cases). However, we do not go that far here.

3.2 The Lovász vector of a graph

We need a modular version of the Lovász vector [14, §2.3] of a graph.

Definition 3.9. Let p be a prime, and $G_1, G_2, \ldots$ be a fixed enumeration of all pairwise non-isomorphic graphs. (Thus every graph is isomorphic to exactly one graph in the sequence.) The *mod-p Lovász vector* of a graph H is the sequence $([\hom(G_i,H)]_p : i \geq 1)$.

This is a modular version of a concept introduced by Lozász, from which the terminology derives. We show that the mod-p Lovász vector determines a graph, provided the graph has no automorphisms

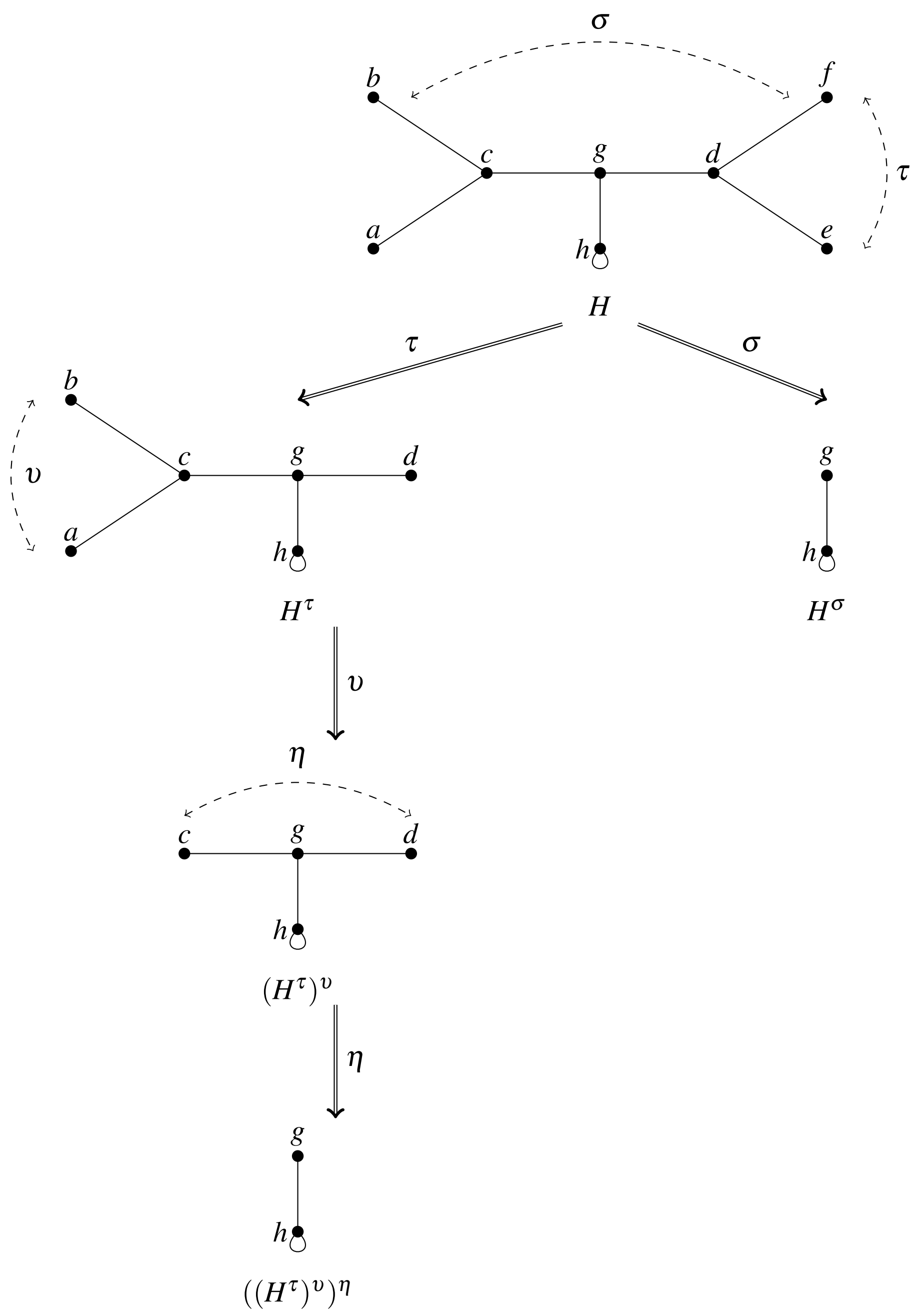

Figure 1: An example of a graph H with the sequence of reductions we get from H if we start with each of the involutions σ and τ.

of order p. For the statement and proof of the classical (non-modular) version of Lemma 3.10 below, see [18, Problem 15.20(b)]. Note that the ideas were generalised by Lovász to a much wider setting [17].

First recall the following fact about finite groups. For any prime p, a finite group G has an element of order p if and only if the order of G is divisible by p. (In the context of group theory, the "if" direction is Cauchy's theorem and the "only if" Lagrange's theorem.)

Lemma 3.10. *Suppose p is a prime, and H and H' are two graphs, neither of which has an automorphism of order p. Then H and H' are isomorphic if and only if they have the same mod-p Lovász vector.*

Proof. Clearly the condition is necessary: two isomorphic graphs have the same mod-p Lovász vector. Now we need to prove that it is sufficient. This proof is similar to the proof of Theorem 2.11 in Hell and Nešetřil's monograph [15].

So suppose H and H' have the same mod-p Lovász vector, that is,

$$\hom(G,H) \equiv \hom(G,H') \pmod{p}, \tag{3.1}$$

for all graphs G. We first observe that, in order to show that H and H' are isomorphic, it is sufficient to prove that for every graph G,

$$\mathrm{inj}(G,H) \equiv \mathrm{inj}(G,H') \pmod{p}, \tag{3.2}$$

where $\mathrm{inj}(G,H)$ denotes the number of injective homomorphisms from G to H. To see this, first take $G = H$ in the above congruence (3.2). The left hand side of the congruence is just the order of the automorphism group of H, which, since H does not have an automorphism of order p, is not congruent to 0 *modulo* p. Therefore, the right hand side, $\mathrm{inj}(H,H')$, is also different from from 0 *modulo* p and, in particular, there exists an injective homomorphism from H to H'. Similarly, if we take $G = H'$ we find an injective homomorphism the other way, and thus an isomorphism between H and H'.

We will prove that the system of congruences (3.1) implies the system (3.2), by induction on n, the number of vertices of G. Specifically, our induction hypothesis is that if congruence (3.1) holds for all graphs G with n or fewer vertices, then the same is true of congruence (3.2). If $n = 1$, then G has only one vertex and every homomorphism from G to any other graph is injective and (3.2) holds.

Now assume $n > 1$. For a partition $\Theta = \{S_i : i \in I\}$ of the vertex set $V(G)$ of a graph G, define the quotient graph G/Θ as follows. The vertex set of G/Θ is the index set I. There is an edge between $i, j \in I$ in G/Θ iff there is some edge joining a vertex in S_i to a vertex in S_j in Θ. (It may happen that $i = j$, in which case G/Θ has a loop at i.) A colouring of G with H induces a partition of G in the obvious way, with vertices which are given the same colour assigned to the same part of the partition. If we call this partition Θ, then any H-colouring of G can be considered as an injective H-colouring of G/Θ, since each vertex of G/Θ is associated with exactly one colour from H. Let ι be the partition consisting of a single block for each vertex (i. e., the partition associated with injective homomorphisms from G to H). Then we have both

$$\hom(G,H) = \mathrm{inj}(G,H) + \sum_{\Theta \neq \iota} \mathrm{inj}(G/\Theta,H)$$

and

$$\hom(G,H') = \mathrm{inj}(G,H') + \sum_{\Theta \neq \iota} \mathrm{inj}(G/\Theta,H').$$

Since G/Θ is necessarily smaller than G if $\Theta \neq \iota$, we know by the induction hypothesis that $\mathrm{inj}(G/\Theta, H) \equiv \mathrm{inj}(G/\Theta, H') \pmod{p}$, and since $\hom(G,H) \equiv \hom(G,H') \pmod{p}$ by assumption, we do indeed have $\mathrm{inj}(G,H) \equiv \mathrm{inj}(G,H') \pmod{p}$, as required.

Note that the largest graph G considered in the above inductive argument has the same number of vertices as H. So if H and H' are not isomorphic then there must be a graph G with at most as many vertices as H that distinguishes H and H', that is, $\hom(G,H) \not\equiv \hom(G,H') \pmod{p}$. □

Proof of Theorem 3.7. Suppose $G \to_p^* G^*$ and $G \to_p^* G^\dagger$, where G^* and $G^\dagger$ have no automorphisms of order p. Theorem 3.4 says the reduction operation $\to_p$ preserves the mod-p Lovász vector, so G^* and $G^\dagger$ have the same vector. On the other hand, Lemma 3.10 above says that the mod-p Lovász vector characterises (isomorphism classes of) graphs with no automorphisms of order p, so G^* and $G^\dagger$ are isomorphic. □

4 Pinning colours to vertices

We would like to be able to count the number of H-colourings of a given graph G in which certain vertices of G are forced to receive certain colours from H. This would allow us to isolate a suitable "hard" subgraph H' of H, and hence reduce the known hard H'-colouring problem to the particular H-colouring problem that interests us. We achieve this by building gadgets, which are graphs with a distinguished vertex, with the following property: effectively, only a certain set of colours can be applied to the distinguished vertex of a gadget. By attaching these gadgets to a vertex of G, we can restrict that vertex to be coloured with a particular set of colours.

4.1 Rooted graphs

Definition 4.1. A *rooted graph* is a pair (G,v) where G is a graph and $v \in V(G)$ is a distinguished vertex of G (referred to as the *root*).

In essence, we want to show that for any two distinct colours $h_1, h_2 \in V(H)$ in a given H, there exists some rooted graph (Γ,γ) such that the number of ways of H-colouring Γ with γ receiving h_1 is different, *modulo* 2, to the number of ways of H-colouring Γ with γ receiving h_2. (In fact, as we can see, we can find such a rooted graph Γ for all prime moduli.) Suppose G is an instance graph with distinguished root vertex v. We can then use rooted graphs such as (Γ,γ) to pick out the colourings of G in which vertex v receives a colour from some particular subset of the colours. Roughly, we do this by attaching a copy of Γ to G, identifying γ and v. Call the resulting graph G'. Suppose a colouring of G with vertex v receiving h_1 extends to a colouring of G' in (say) an odd number of ways. Then a colouring with v receiving h_2 will extend in an even number of ways. In this way we have effectively "cancelled" the colourings of G with v coloured h_2, while leaving untouched those with v coloured h_1.

The construction of the required gadgets rests on a rooted version of Lemma 3.10. Before we give the proof, we need to define rooted versions of a few concepts we have already encountered.

Definition 4.2. A *homomorphism* (repectively, *isomorphism*) between two rooted graphs (G,v) and (G',v') is a graph homomorphism (respectively, isomorphism) $\varphi : V(G) \to V(G')$ with $\varphi(v) = v'$. An *automorphism* of rooted graph (G,v) is an isomorphism of (G,v) to itself.

Definition 4.3. We denote the number of homomorphisms from rooted graph (G,g) to rooted graph (H,h) by $\hom^*((G,g),(H,h))$. If the roots are implied by the context we will sometimes suppress them in the above notation, and just write $\hom^*(G,H)$.

Similarly, we denote the number of injective homomorphisms from rooted graph (G,g) to (H,h) by $\mathrm{inj}^*((G,g),(H,h))$ and, again, we may suppress the specified vertices if they are implied by the context, instead writing $\mathrm{inj}^*(G,H)$.

Finally, we will use the concept of the Lovász vector of a rooted graph.

Definition 4.4. Let $G_1,G_2,\dots$ be a fixed enumeration of all pairwise non-isomorphic rooted graphs. Then the mod-p Lovász vector of a rooted graph H is the sequence $([\hom^*(G_i,H)]_p : i \geq 1)$.

We will use *parity Lovász vector* as an alternative name for mod-2 Lovász vector.

Lemma 4.5. *Suppose p is a prime, and H and H' are two rooted graphs neither of which has an automorphism of order p. Then H and H' are isomorphic if and only if they have the same mod-p Lovász vector.*

Proof. As for Lemma 3.10, but with $\hom^*$ and inj^* replacing hom and inj. In defining the quotient of a rooted graph (G,g) by a partition $\Theta = \{S_i : i \in I\}$, we define the root of $(G,g)/\Theta$ to be the vertex $i \in I$ such that $g \in S_i$. $\square$

As with Lemma 3.10, it can be seen that we need only finitely many terms of the mod-p Lovász vector to reconstruct (H,h).

4.2 Building gadgets

In the following we return to $\oplus H$-COLOURING, and are only interested in automorphisms of order two, or involutions. Note that many of the results in this section can be generalised to automorphisms of arbitrary prime order, but we only require the gadgets for the case $p = 2$ in Section 5, so only this case is presented here, for simplicity.

It will be useful to consider the case where H and H' have the same underlying graph but different roots (note that for H and H' to be non-isomorphic as rooted graphs, there can be no automorphism of H with takes h to h', i. e., that h and h' lie in different orbits of the automorphism group of H). Since we will no longer be able to use the previous naming convention for the specified vertices, we will refer to the two roots in H as x and y. In the following, we will be assuming that H (as an unrooted graph) is involution-free. As we saw in Section 3, it suffices to consider the complexity of $\oplus H$-COLOURING for involution-free H.

Lemma 4.5 allows us to construct the following useful gadgets: given an involution-free graph H and two colours x and y which are in different orbits of $\mathrm{Aut}(H)$, there is a rooted graph (Γ,γ) that distinguishes x and y.

Lemma 4.6. *Given an involution-free graph H and two vertices x and y which lie in different orbits of* $\mathrm{Aut}(H)$*, there exists a rooted graph (Γ,γ) such that* $\hom^*((\Gamma,\gamma),(H,x)) \not\equiv \hom^*((\Gamma,\gamma),(H,y)) \pmod 2$.

Proof. Since (H,x) and (H,y) are non-isomorphic as rooted graphs, they have different parity Lovász vectors by Lemma 4.5. Simply take (Γ,γ) to be the first rooted graph for which the corresponding entries of the parity Lovász vectors of (H,x) and (H,y) differ. $\square$

We will use rooted graphs such as those guaranteed by Lemma 4.6 as "gadgets" in a reduction from the problem of counting restricted H-colourings (in which a given vertex of the instance graph is forced to be coloured with colours from a specified orbit of $\mathrm{Aut}(H)$) to the problem of counting unrestricted H-colourings *modulo* 2.

Theorem 4.7. *Given an involution-free graph H, an orbit O of the automorphism group of H, and an oracle for $\oplus H$-COLOURING, it is possible to determine, in polynomial time, the parity of the number of H-colourings of a rooted graph G in which the root receives a colour from O.*

Note that this result would follow immediately if we were able to build a gadget (i. e., rooted graph) (Γ,γ) such that $\hom^*((\Gamma,\gamma),(H,x))$ is odd, while $\hom^*((\Gamma,\gamma),(H,y))$ is even for all $y \neq x$. Then we could just attach a copy of Γ at the vertex of G that we want to colour with x, identifying this vertex with γ, and then count H-colourings of the new graph. Unfortunately, Lemma 4.6 doesn't allow us to construct such a gadget, as it doesn't allow us to choose which colour is x and which is y. However, we can construct a series of gadgets that allow us to count colourings of G in which the root of G receives a colour from a given orbit of H, by developing a sort of algebra on the gadgets, as described below.

Definition 4.8. Suppose H is a graph, and $h_1,\dots,h_n$ is an enumeration of the vertices of H. With each gadget (Γ,γ) we associate a vector $v_H(\Gamma) \in \mathrm{GF}(2)^n$, indexed by $\{1,\dots,n\}$, such that the i^{th} component of the vector is 1 if there are an odd number of H-colourings of Γ that use colour h_i at γ, and 0 otherwise.

Note that if two colours (vertices of H) h_i and h_j are in the same orbit of the automorphism group of H then the i^{th} and j^{th} entries of $v_H(G)$ are the same for all rooted graphs G. So we may instead consider the vector $v^*_H(G)$ which is indexed by *orbits* of the automorphism group of H rather than individual vertices of H, the coordinate of $v^*_H(G)$ associated with a given orbit being the coordinate of $v_H(G)$ associated with any (and hence all) of the colours in that orbit. Note that $v_H(G)$ and $v^*_H(G)$ contain exactly the same information.

We define an operation that combines two rooted graphs by identifying their root vertices.

Definition 4.9. Given two rooted graphs Γ and Π, we define the the new rooted graph $\Gamma\cdot\Pi$ to be the graph obtained by identifying the roots of each. The root of $\Gamma\cdot\Pi$ is the vertex formed by identifying the roots of the other two graphs.

If we think of each gadget Γ and Π as enforcing a certain set of allowed colours at its root vertex, we can view this operation as forming a gadget that enforces the intersection of these sets. This is equivalent to saying that vector v_H associated with the new gadget is obtained by taking the coordinate-wise product of the vectors associated with the individual gadgets.

Definition 4.10. We define the operation $* : \mathrm{GF}(2)^n \times \mathrm{GF}(2)^n \to \mathrm{GF}(2)^n$ to be the coordinate-wise product of two vectors, so the i^{th} coordinate of $v * w$ is the i^{th} coordinate of v multiplied by the i^{th} coordinate of w.

Lemma 4.11. *Suppose Γ and Π are two rooted graphs, and H is graph. Then $v_H(\Gamma\cdot\Pi) = v_H(\Gamma) * v_H(\Pi)$.*

Proof. Fix a colour $h_i \in V(H)$. The number of colourings of $\Gamma\cdot\Pi$ with the root receiving colour h_i is just the product of the number of colourings Γ and Π with the roots in each case receiving colour h_i. Thus, if there is a zero in the i^{th} place of either of the vectors $v_H(\Gamma)$ or $v_H(\Pi)$, then there is a zero in the i^{th} place of $v_H(\Gamma\cdot\Pi)$; otherwise there is a one. □

We now introduce a formal sum of rooted graphs, with coefficients in GF(2), which preserves addition of these vectors. Note that since this sum has coefficients in GF(2) we have $\Gamma+\Gamma = 0$.

Definition 4.12. For a set of rooted graphs $\Gamma_1, \Gamma_2, \cdots, \Gamma_r$, we define $v_H(\Gamma_1+\Gamma_2+\cdots+\Gamma_r)$ to be $v_H(\Gamma_1)+v_H(\Gamma_2)+\cdots+v_H(\Gamma_r)$.

Definition 4.13. We will say that a vector $v \in \mathrm{GF}(2)^n$ is *implementable* for some n-vertex H if there is a set of rooted graphs $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_r\}$ such that v is equal to $v_H(\Gamma_1+\Gamma_2+\cdots+\Gamma_r)$.

Suppose v is the characteristic vector of a set of colours we wish to restrict to, as in the discussion following Theorem 4.7. We'll see presently that the gadgets $\Gamma_1, \ldots, \Gamma_r$ will enable us to effectively restrict to that colour set, justifying the term "implementable".

Lemma 4.14. *The set of vectors that are implementable for a given H is closed under the operations of vector addition and point-wise multiplication (or the operation $*$, as defined in Definition 4.10).*

Proof. Suppose $v = v_H(\Gamma_1+\Gamma_2+\cdots+\Gamma_r)$ and $v' = v_H(\Pi_1+\Pi_2+\cdots+\Pi_s)$ are any two implementable vectors. Then $v+v'$ is implementable, since

$$v+v' = v_H(\Gamma_1+\Gamma_2+\cdots+\Gamma_r+\Pi_1+\Pi_2+\cdots+\Pi_s)\,.$$

Furthermore, noting that $*$ distributes over $+$,

$$\begin{aligned}
v*v' &= v_H(\Gamma_1+\Gamma_2+\cdots+\Gamma_r)*v_H(\Pi_1+\Pi_2+\cdots+\Pi_s)\\
&= \big(v_H(\Gamma_1)+v_H(\Gamma_2)+\cdots+v_H(\Gamma_r)\big)*\big(v_H(\Pi_1)+v_H(\Pi_2)+\cdots+v_H(\Pi_s)\big)\\
&= v_H(\Gamma_1)*v_H(\Pi_1)+v_H(\Gamma_1)*v_H(\Pi_2)+\cdots+v_H(\Gamma_r)*v_H(\Pi_s)\\
&= v_H(\Gamma_1\cdot\Pi_1)+v_H(\Gamma_1\cdot\Pi_2)+\cdots+v_H(\Gamma_r\cdot\Pi_s) \qquad (4.1)\\
&= v_H(\Gamma_1\cdot\Pi_1+\cdots+\Gamma_r\cdot\Pi_s)\,,
\end{aligned}$$

where equality (4.1) follows from repeated application of Lemma 4.11. □

Lemma 4.15. *For any involution-free graph, H, the all-ones vector is implementable, and for any pair of distinct orbits in H there is at least one implementable vector that has a 1 at every vertex in one of the two orbits and a 0 at every vertex in the other orbit.*

Proof. The all-ones vector is implementable using the graph on one vertex. The rooted graphs whose vectors distinguish between distinct orbits of colours in H are obtained using Lemma 4.6. □

We'll now show that Lemmas 4.14 and 4.15 together imply that any vector that is constant on orbits is implementable.

Lemma 4.16. *Consider a set, S, of vectors in* $\mathrm{GF}(2)^n$ *which contains the all-ones vector* $(1,1,\ldots,1)$ *and has the property that for any two indices i and j there is some vector in the set whose* i^{th} *coordinate differs from its* j^{th} *coordinate. The closure of this set under the operations of coordinate-wise multiplication and coordinate-wise addition includes each of the vectors in the standard basis.*

Proof. We proceed by induction on n. If $n=1$ the lemma clearly holds, as the all-ones vector is the only vector in the standard basis. Now, assume that $n>1$; we shall attempt to construct the vectors in the standard basis in $\mathrm{GF}(2)^n$.

By induction, we can construct vectors that agree with the standard basis in the first $n-1$ places, without being able to control what happens in the n^{th} place (note that the restriction of the set of vectors S to the first $n-1$ places still satisfies the conditions of the lemma). That is, we can certainly obtain vectors of each of the following forms, where the x_i can be either 0 or 1

$$\begin{array}{llllllll}
(1 & 1 & 1 & 1 & \ldots & 1 & 1 & 1) \\
(1 & 0 & 0 & 0 & \ldots & 0 & 0 & x_1) \\
(0 & 1 & 0 & 0 & \ldots & 0 & 0 & x_2) \\
(0 & 0 & 1 & 0 & \ldots & 0 & 0 & x_3) \\
\vdots & & & & & & & \vdots \\
(0 & 0 & 0 & 0 & \ldots & 0 & 1 & x_n)
\end{array}$$

This leaves several cases:

Case 1. The x_i are all equal to zero. In this case, we already have the first $n-1$ vectors from the standard basis, and we can just take the sum of all $n-1$ vectors with the all-ones vector, which has a 1 in the last place and zeros everywhere else, to get the last one.

Case 2. There are at least two i,j such that $x_i,x_j=1$. But then the product of these two vectors is the vector $(0,0,\ldots,0,1)$. To obtain the remaining vectors from the standard basis, we just take the sum of this vector with any of those from the original list which had a 1 in the n^{th} place, i. e., e_i is the sum of this vector with the vector that had a 1 in the i^{th} place and a 1 in the n^{th} place.

Case 3. There is exactly one vector in the list, v with a 1 as the n^{th} coordinate. Say this vector has a 1 in the i^{th} and n^{th} places. By assumption, there is some vector in S which has different values in the n^{th} and i^{th} places. The product of this with v is a vector with exactly one 1, in either the i^{th} or the n^{th} place, and the sum of this basis vector with v is the other of e_i and e_n. □

Lemma 4.17. *For any involution-free graph H, and any orbit of O of* $\mathrm{Aut}(H)$*, the characteristic vector of O (which is 1 in coordinates indexed by O and 0 elsewhere) is implementable.*

Proof. For the purposes of this proof, it is convenient to think in terms to the abbreviated vectors $v^*_H(G)$ in place of the full vectors $v_H(G)$. (This is not an essential change; we are merely eliminating duplicated coordinates.) So, now, an implementable vector is one the form $v^*_H(\Gamma_1+\cdots+\Gamma_r)$, for some rooted graphs $\Gamma_1,\ldots,\Gamma_r$. By Lemma 4.14 the set of vectors we can implement is closed under the operations of addition and coordinate-wise multiplication, and by Lemma 4.15 we can implement the all ones vector and, for each pair of indices (orbits) i and j a vector v with $v_i \neq v_j$. Thus, by Lemma 4.16, every vector in the standard basis is implementable. □

We are now ready to return to Theorem 4.7. Let v be the characteristic vector of the orbit O. We know that v is implementable. So we now just have to show that our definition of "implementable" actually does what we want it to do. That is, it is possible to determine, in polynomial time using an oracle for unrestricted H-colourings, the parity of the number of H-colourings of a rooted graph G in which the root receives a colour from O.

Proof of Theorem 4.7. Let $v \in \mathrm{GF}(2)^n$ be the characteristic vector of the orbit O. By Lemma 4.17, the vector v is implementable, i. e., $v = v_H(\Gamma_1 + \Gamma_2 + \cdots + \Gamma_r)$ for some set of rooted graphs $\{\Gamma_1, \ldots, \Gamma_r\}$. Thus,

$$\begin{aligned} v_H(G) * v &= v_H(G) * v_H(\Gamma_1 + \Gamma_2 + \cdots + \Gamma_r) \\ &= v_H(G) * v_H(\Gamma_1) + \cdots + v_H(G) * v_H(\Gamma_r) \\ &= v_H(G \cdot \Gamma_1) + \cdots + v_H(G \cdot \Gamma_r)\,. \end{aligned}$$

Now take the sum of the coordinates of the vectors, *modulo* 2:

$$\sum_{i=1}^{n} (v_H(G) * v)_i = \sum_{i=1}^{n} v_H(G \cdot \Gamma_1)_i + \cdots + \sum_{i=1}^{n} v_H(G \cdot \Gamma_r)_i\,.$$

The left-hand side counts, *modulo* 2, H-colourings of G in which vertex x receives a colour from O; this is exactly the quantity we are interested in computing. The j^{th} term on the right hand side, counts, *modulo* 2, the number of (unrestricted) H-colourings of the graph $G \cdot \Gamma_j$. So the right-hand side can be evaluated using r calls to an oracle for $\oplus H$-COLOURING. □

Finally, we need an analogue of Theorem 4.7 which allows pinning of two vertices of G. (We thank the authors of [10] for pointing out a lacuna at this point in an earlier version of the proof.)

Corollary 4.18. *Suppose G is a graph with distinguished vertices x and y. Given an involution-free graph H, orbits O and O' of the automorphism group of H, and an oracle for $\oplus H$-COLOURING, it is possible to determine, in polynomial time, the parity of the number of H-colourings of G in which x (respectively y) receives a colour from O (respectively O').*

Proof. Define the matrix $A = (a_{ij}) \in \mathrm{GF}(2)^{n \times n}$ as follows. For all $1 \le i, j \le n$,

$$a_{ij} = \big[\text{number of colourings of } G \text{ with } x \text{ receiving colour } i \text{ and } y \text{ colour } j\big]_2\,.$$

Let u and v be the characteristic vectors of O and O'. By Lemma 4.17 we know that u and v are implementable, i. e., $u = v_H(\Gamma_1) + \cdots + v_H(\Gamma_r)$ and $v = v_H(\Gamma'_1) + \cdots + v_H(\Gamma'_s)$ for some rooted graphs $\Gamma_1, \ldots, \Gamma_r$ and $\Gamma'_1, \ldots, \Gamma'_s$. Thus

$$\begin{aligned} u^{\mathsf{T}} A v &\equiv \big(v_H(\Gamma_1) + \cdots + v_H(\Gamma_r)\big)^{\mathsf{T}} A \big(v_H(\Gamma'_1) + \cdots + v_H(\Gamma'_s)\big) \\ &\equiv \sum_{i=1}^{r} \sum_{j=1}^{s} v_H(\Gamma_i)^{\mathsf{T}} A\, v_H(\Gamma'_j) \pmod 2\,. \end{aligned}$$

Note that the left hand side is the quantity we are interested in, namely the number of restricted H-colourings of G. Finally note that the $(i, j)^{\text{th}}$ term in the last sum is equal, *modulo* 2, to the number of colourings of G with Γ_i attached to x and Γ'_j to y. So each term on the right hand side may be computed using an oracle for $\oplus H$-COLOURING. □

5 Trees

As we have seen, applying the reduction operations in Definition 3.5 preserves the parity of the number of H-colourings of any graph G. This allows us to concentrate on involution-free graphs. There are certain involution-free graphs H for which the H-colouring problem obviously lies in P:

- the null graph (the graph on no vertices),
- the graph on one vertex with no loop,
- the graph on one vertex with a loop, and
- the graph on two disconnected vertices, one with a loop and one without.

(5.1)

Lemma 5.1. *If H is one of the graphs in list 5.1, then H-colourings of an instance G can be counted in polynomial time.*

Proof. If H is the null graph then there is no H-colouring of G, so the counting problem is obviously trivial. If H is the graph on one vertex then G has exactly one H-colouring if and only if G has no edges, and zero otherwise, which can be determined in polynomial time. If H is the graph on one vertex with a loop, then there is exactly one H-colouring of G. If H is the graph on two vertices one with a loop and one without then there are exactly $2^{|\mathrm{Isol}(G)|}$ colourings of G, where $\mathrm{Isol}(G)$ is the set of isolated vertices of G. Each isolated vertex can be coloured with either the looped vertex or the unlooped vertex of H independently, and all the vertices which form part of a connected component of size greater than one must be coloured with the looped vertex. □

Corollary 5.2. *Let H_0 be the reduced form associated with H in the reduction system defined in Definition 3.5. If H_0 is one of the graphs in list 5.1, then $\oplus H$-COLOURING is in P.*

Proof. This follows directly from Lemma 5.1 and the fact that the reduction system preserves the parity of the number of H-colourings, as shown in Lemma 3.3 □

We conjecture that for general graphs, the criterion given in Corollary 5.2, that is, H reducing by involutions to one of the four trivial graphs, is the only way in which the $\oplus H$-COLOURING problem can fail to be $\oplus\mathsf{P}$-complete. Note that this criterion does encompass all of the easy cases identified by Dyer and Greenhill [6]. A complete graph with loops everywhere reduces to the null graph if it has an even number of vertices and the graph on one vertex with a loop if it has an odd number. On the other hand, a complete bipartite graph reduces to the graph on one vertex if there are an odd number of vertices in total, and the null graph otherwise.

In this section, we will prove that this conjecture is true for trees. In particular, if, in the reduction system of Definition 3.5, the reduced form associated with a given tree T has at most one vertex, then the associated $\oplus T$-COLOURING problem can be solved in polynomial time. Otherwise, it is $\oplus\mathsf{P}$-complete. Note that Göbel, Goldberg and Richerby have recently extended the known range of validity of the conjecture from trees to cactus graphs [10] and square-free graphs [11].

5.1 Involution-free trees

Involution-free trees have quite a lot of structure, and we will exploit this when we build gadgets for our reductions from $\oplus$INDSET (defined below) to $\oplus H$-COLOURING in the next section.

Lemma 5.3. *An involution-free tree on more than one vertex has two vertices of degree 2 which are adjacent to leaves.*

Proof. The argument given below is very similar to the standard argument given to show that any tree has at least two leaves.

The first observation to make is that any involution-free tree contains some path of length at least 3. If the maximum-length path in a tree is of length 1, then the tree consists of a single edge, and so has an involution. If it is of length 2, then the tree is a star, and exchanging any two of its leaves is an involution.

Consider a longest path in an involution-free tree, and label the vertices of this path $p_0, p_1, \dots, p_\ell$. Note that p_0 and p_ℓ are both leaves. Then we claim that both vertices p_1 and $p_{\ell-1}$ are degree 2. Note that p_1 and $p_{\ell-1}$ are in fact distinct vertices, as $\ell \geq 3$. Assume the degree of p_1 is greater than 2, and consider a vertex, v, adjacent to p_1, which is neither p_0 nor p_2. This vertex cannot have any neighbours which are not already in the path (as this would contradict maximality of the path). It also cannot have any neighbours which are in the path (as this would create a cycle, contradicting the fact that G is a tree). Therefore, it cannot have any neighbours other than p_1. But then exchanging this vertex with p_0 is an involution of G, so there is no such vertex, and p_1 is degree 2 as claimed. An analogous argument shows that $p_{\ell-1}$ must be degree 2. □

We will also require the following lemma.

Lemma 5.4. *An involution-free tree has trivial automorphism group.*

Proof. The automorphism group of a tree can be formed from symmetric groups using the operations of direct product and wreath product with a symmetric group (Pólya [20] after Jordan [16]). Since the symmetric groups S_n for $n > 1$ have even order, the automorphism group of a tree is either of even order or has order 1. If it has even order, then the tree has an involution. □

Finally, we require the following technical lemma concerning the number of walks (i. e., not necessarily simple paths) of various lengths between vertices in involution-free trees. Note that the vertices e_0 and e_ℓ mentioned in the statement of the lemma are guaranteed by Lemma 5.3.

Lemma 5.5. *Let H be an involution-free tree, let e_0 be a vertex of degree 2 that is adjacent to a leaf in H, and let e_ℓ be a vertex of even degree such that there are no vertices of even degree on the path joining e_0 and e_ℓ, where $\ell \geq 1$ is the length of the path joining e_0 and e_ℓ. We will name the vertices on this path $e_0, o_1, o_2, \dots, o_{\ell-1}, e_\ell$.*

Then there are an even number of vertices v such that both:

1. *v is a neighbour of the first vertex on this path other than e_0, i. e., v is a neighbour of e_1 in the case $\ell = 1$, and a neighbour of o_1 otherwise; and*

2. *the number of walks of length ℓ from v to e_ℓ in H is odd.*

Proof. We will refer in this proof to the vertices o_1 and o_2, which do not exist if $\ell = 1$ or $\ell = 2$, we deal with this at the end of this proof. For now, assume $\ell \geq 3$. We want to prove that there are an even number of neighbours of o_1 from which there are an odd number of walks of length ℓ to e_ℓ in H. There are an odd number of paths of length ℓ from e_ℓ to each of the neighbours of o_1 other than o_2: there is, in fact, one such walk, and it is the unique path connecting the neighbour to e_ℓ in the tree. We claim that there are an even number of walks of length ℓ from e_ℓ to o_2.

A walk of length ℓ from e_ℓ to o_2 traverses exactly 1 edge more than once, as there is a unique path of length $\ell - 2$ from e_ℓ to o_2. Two such walks which traverse the same edge more than once are identical. There is therefore a one-to-one correspondence between these walks and the edges which are traversed at least twice by at least one of them. We claim that the number of such edges is even.

Any edge which is adjacent to any of the vertices in $\{o_2, o_3, \ldots, e_\ell\}$, and only those edges, may be traversed more than once, so it suffices to show that there are an even number of such edges. To see this, note that the only edges in this set which are adjacent to more than one of the vertices in the set are: $\{(o_2, o_3), (o_3, o_4), \ldots, (o_{\ell-1}, e_\ell)\}$, there are the same number of edges in this set as the number of vertices of odd degree in $\{o_2, \ldots, e_\ell\}$. The total number of edges is then just the sum of the vertex degrees minus the number of edges which are adjacent to more than one of the vertices; but the sum of the vertex degrees is $\ell - 2 \pmod 2$ (as there are $\ell - 2$ vertices of odd degree) and the number of repeated edges is $\ell - 2$, so the parity of the total number of edges is $(\ell - 2) - (\ell - 2) \equiv 0 \pmod 2$.

As noted above, if $\ell = 1$ or if $\ell = 2$ the vertices o_1 or o_2 may not exist. However, the theorem still holds.

In particular, if $\ell = 1$ then we actually have two adjacent vertices of even degree and the first vertex on the path which is not e_0 is in fact e_1, which is of even degree. Clearly there are an even number of vertices adjacent to e_1 with an odd number of length 1 walks to e_1, these being exactly the neighbours of e_1.

If $\ell = 2$, then again the vertex whose neighbours we are interested in is of odd degree, call it o_1, and there are an odd number of walks of length 2 from e_2 to each of the neighbours of o_1 other than itself: in fact, there is exactly one such walk, the path joining the two vertices. On the other hand, e_2 is of even degree, so there are an even number of walks of length 2 from e_2 to itself. Since o_1 has an odd number of neighbours, this leaves an even number of neighbours of o_1 which have an odd number of length 2 walks to e_2, as claimed. $\square$

5.2 The reduction

Our starting point is the following problem, which was shown by Valiant [22] (in the guise of "Mon 2-CNF") to be $\oplus$P-complete; see also Faben [8, Thm. 3.5]).

Name. $\oplus$INDSET.

Instance. An undirected graph G.

Output. The parity of the number of independent sets in G.

Theorem 5.6. *Given an involution-free tree H with more than one vertex, $\oplus H$-COLOURING is $\oplus$P-complete. In fact, there is a polynomial-time reduction from $\oplus$INDSET to $\oplus H$-COLOURING.*

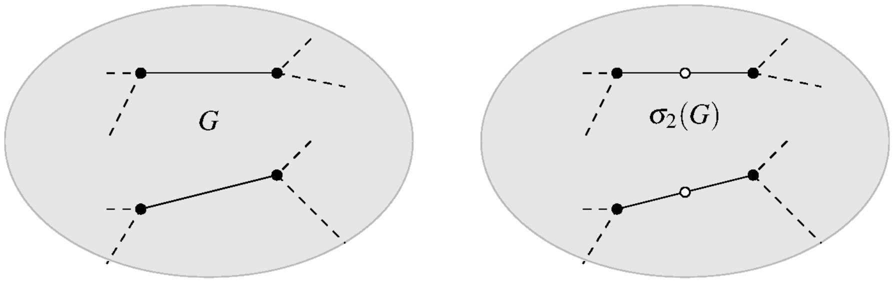

Figure 2: The 2-stretch of G.

Definition 5.7. Given a graph G, we call $\sigma_2(G)$ the graph obtained by replacing every edge in G with a path of length 2. We refer to the newly introduced vertices as *stretch vertices*, and the original vertices of G as *G-vertices*. The construction is illustrated in Figure 2.

The graph defined above, $\sigma_2(G)$, is usually referred to as the *2-stretch* of G, and it is an established result that counting H-colourings of $\sigma_2(G)$ is equivalent to counting H^2-colourings of G, where H^2 is the multigraph whose adjacency matrix is the square of the adjacency matrix of H (see, e. g.,[6]). We will use a variant of this stretch operation in which we count only those colourings of $\sigma_2(G)$ in which both the stretch vertices and the G-vertices are coloured with specific subsets of the colours in H. This is achieved using gadgetry based on the principles established in Section 4.

We now detail the reduction from $\oplus$INDSET. First, given any graph G, we will construct a certain graph G^*. We then claim that the number of H-colourings of G^*, with certain vertices restricted to receive certain colours from H, is congruent *modulo* 2 to the number of independent sets in G.

For a given involution-free tree H, pick a vertex of degree 2, e_0, adjacent to a leaf, and a vertex of even degree, e_k such that the unique path of length k in H from e_0 to e_k does not contain any vertex of even degree (exactly as in the statement of Lemma 5.5). Note that, as H is involution-free, there are two vertices of even degree, and at least one vertex of degree two which is adjacent to a leaf in H by Lemma 5.3, and we can choose e_0 and e_k with the above properties.

Now, given a graph G, first create $\sigma_2(G)$, then add two new vertices R and B. Add an edge between each of the original vertices of G (G-vertices) and R, and a path of length k from every one of the new vertices (stretch vertices) of $\sigma_2(G)$ to B. We call this new graph G^*, and the construction is illustrated in Figure 3.

Now, using the technology described in Corollary 4.18, and the fact that the orbit of a vertex in an involution-free tree is trivial by Lemma 5.4, we can determine the parity of the number of H-colourings of G^*, in which R is restricted to be coloured with e_0 and B is restricted to be coloured with e_k, using only a $\oplus H$-COLOURING oracle. We claim that this number is congruent (*modulo* 2) to the number of independent sets in G. We will use what we know about the number of walks of length k between the colours e_0 and e_k from Lemma 5.5.

Lemma 5.8. *Suppose H is an involution-free tree, and let e_0 be a vertex of degree 2 adjacent to a leaf, and e_k a vertex of even degree at distance $k \geq 1$ from e_0 such that there are no vertices of even degree on*

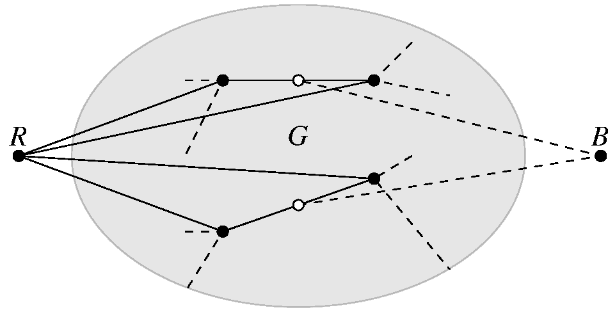

Figure 3: The construction of G^*.

the path of length k joining them. Suppose G is a graph and let G^* be constructed from G as described above.

Then the number of H-colourings of G^* in which R receives e_0 and B receives e_k is congruent modulo 2 to the number of independent sets in G.

Proof. First consider the G-vertices in G. They are all neighbours of a vertex which is coloured with e_0, so they must therefore receive colours that are adjacent to e_0 in H. But e_0 was chosen to be one of the vertices of degree 2 adjacent to a leaf in H, so G-vertices can only be coloured with either the leaf adjacent to e_0 (which we will call l) or with the first vertex on the path linking e_0 and e_k, which we will call v_1 in the remainder of this proof. This vertex is o_1, except in the case $k = 1$ where it is e_1.

Now, consider the stretch vertices. These are connected to a vertex which is coloured e_k by a path of length k. So, consider the colour used at a given stretch vertex, s. If there are an even number of walks of length k from e_k to this colour in H, then there are an even number of colourings of G^* which use that colour at s, as there are an even number of ways of colouring the path joining s and B, and the total number of colourings is the product of the number of ways of colouring this path with the number of ways of colouring the rest of the graph.

We therefore need to count colourings of G^* in which the colours used at the stretch vertices are such that there are an odd number of paths of length k between them and e_k in H. Note that these colours must also be adjacent to either v_1 or l in H (as the G-vertices are all coloured with either v_1 or l, and every stretch vertex is adjacent to a G-vertex), and therefore, in fact, must be adjacent to v_1, as the only neighbour of l is e_0, which is also a neighbour of v_1.

Now, we are reduced to considering colourings of G^* in which the following conditions hold. The G-vertices are coloured either l or v_1, while the stretch vertices are coloured with one of the neighbours of v_1 which has an odd number of length k walks from itself to e_k. We claim that the parity of the number of such colourings is equal to the parity of the number of ways of colouring G with the two colours l and v_1 such that no two vertices coloured with v_1 are adjacent.

Consider a colouring of G with the colours v_1 and l. If there are two vertices of G which are adjacent in G and both coloured with v_1 then there are an even number of extensions of this colouring to an H-colouring of G^*: the stretch vertex between the two G-vertices in G^* can be coloured with any one of the neighbours of v_1 which are at distance k from e_k in H, and there are an even number of such vertices by Lemma 5.5.

On the other hand, if there are no two such vertices, there is exactly one extension of the given colouring of G to an H-colouring of G^*: every one of the stretch vertices is adjacent to a vertex which is coloured l, so the stretch vertices must all be coloured e_0, and as there is only one path of length k from e_0 to e_k in H, this determines the colouring of the vertices on the paths linking the stretch vertices to B.

So the number of colourings of G^* with H such that R is coloured e_0 and B is coloured e_k is congruent *modulo* 2 to the number of colourings of G in which each vertex is either coloured with l or v_1 and adjacent vertices may not both be coloured with v_1. But these are exactly the independent sets of G: vertices coloured v_1 are "in" the independent set and vertices coloured l are "out". □

Proof of Theorem 5.6. By Theorem 4.18 and Lemma 5.4 we can count H-colourings of G^* in which R is coloured e_0 and B is coloured e_k in polynomial time if equipped with an H-colouring oracle. But we know that the number of such colourings is congruent *modulo* 2 to the number of independent sets in G. Since clearly G^* can be constructed from G in polynomial time, this gives us a polynomial-time Turing reduction from $\oplus$INDSET to $\oplus H$-COLOURING. □

5.3 A dichotomy for trees

The main result now follows easily.

Proof of Theorem 3.8. By Lemma 3.3, the number of H-colourings of a graph G is congruent *modulo* 2 to the number of H'-colourings, where H' is any graph obtained from H by reducing H by any of its involutions. Also, if H is a tree then any graph H' which can be reached from H by reduction by involutions is also a tree. It therefore suffices to consider involution-free trees.

If H is an involution-free tree, and H contains more than one vertex, then Theorem 5.6 shows that $\oplus H$-COLOURING is $\oplus$P-complete. On the other hand, if H contains either 0 or 1 vertices then #H-COLOURING (and hence $\oplus H$-COLOURING) is polynomial-time solvable by Lemma 5.1. □

Note that the dichotomy described by Theorem 3.8 is decidable in polynomial time.

6 Other graphs

As noted earlier, we conjecture not only that there is a dichotomy for the complexity of $\oplus H$-COLOURING for general H, but that this dichotomy arises in the same way as it does for trees. In other words, that the only way in which a $\oplus H$-COLOURING problem can be polynomial-time solvable is if H reduces by involutions to one of the four trivial graphs. If this is the correct characterisation, then the dichotomy is certainly decidable, but it is not clear whether it can be decided in polynomial time. On the face of it, finding the reduced form associated with a graph H requires finding an involution of H, and no polynomial-time algorithm is known for this problem.

We finish by showing that in uncovering a dichotomy for general graphs it is enough to consider connected H. That is, if an involution-free graph H has any connected component H_1 for which $\oplus H_1$-COLOURING is $\oplus$P-hard, then the parity colouring problem associated with H is itself $\oplus$P-hard.

Theorem 6.1. *Let H be an involution-free graph. If H_1 is a connected component of H and $\oplus H_1$-*COLOURING *is $\oplus$P-hard, then $\oplus H$-*COLOURING *is $\oplus$P-hard.*

Proof. Take any graph G, and assume that G is connected (since the number of H-colourings of G is just the product of the number of H-colourings of each of its connected components). We can use an oracle for H-colouring to determine the parity of the number of colourings of G in which only colours from H_1 are used in the following way: let $v \in V(G)$ be any vertex of G. For each colour $h_i \in V(H_1)$, we can count the colourings of G in which v is coloured h_i using Theorem 4.7. Notice that the size of the orbit of h_i in $\mathrm{Aut}(H)$ is odd, as H has no involutions, so the parity of the number of colourings of G with h_i at v is the same as the parity of the number of colourings of G which use any of the vertices in the orbit of h_i at v.

But we can do this for every vertex in H_1, and since G is connected, any colouring which uses a vertex from H_1 at v can use only colours from H_1 anywhere in G. Conversely, any colouring of G which uses only colours from H_1 must use some colour from H_1 at v, so this does indeed allow us to count all such colourings of G. □

Note that this actually allows us to strengthen Theorem 3.8: the H-colouring problem associated with any *forest* H is polynomial-time solvable if the reduced form associated with the forest in the reduction system described in Section 3 is the null graph or the graph on one vertex, and $\oplus$P-complete otherwise.

References

[1] ANDREI A. BULATOV: The complexity of the counting constraint satisfaction problem. *J. ACM*, 60(5):34, 2013. Preliminary version in ICALP'08, also available at ECCC. [doi:10.1145/2528400] 36

[2] ANDREI A. BULATOV AND MARTIN GROHE: The complexity of partition functions. *Theoret. Comput. Sci.*, 348(2-3):148–186, 2005. Preliminary version in ICALP'04. [doi:10.1016/j.tcs.2005.09.011] 36

[3] JIN-YI CAI AND XI CHEN: Complexity of counting CSP with complex weights. In *Proc. 44th STOC*, pp. 909–920, New York, 2012. ACM Press. [doi:10.1145/2213977.2214059, arXiv:1111.2384] 36

[4] JIN-YI CAI, XI CHEN, AND PINYAN LU: Graph homomorphisms with complex values: A dichotomy theorem. *SIAM J. Comput.*, 42(3):924–1029, 2013. Preliminary versions in ICALP'10 and arXiv. [doi:10.1137/110840194] 36

[5] XI CHEN: Guest column: Complexity dichotomies of counting problems. *SIGACT News*, 42(4):54–76, 2011. [doi:10.1145/2078162.2078177] 36

[6] MARTIN DYER AND CATHERINE GREENHILL: The complexity of counting graph homomorphisms. *Random Structures Algorithms*, 17(3-4):260–289, 2000. Extended abstract in SODA'00, Corrigendum in Random Struct. Algorithms. [doi:10.1002/1098-2418(200010/12)17:3/4<260::AID-RSA5>3.0.CO;2-W] 36, 38, 49, 52

[7] MARTIN DYER AND DAVID RICHERBY: The #CSP dichotomy is decidable. In *Proc. 28th Symp. Theoretical Aspects of Comp. Sci. (STACS'11)*, volume 9 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 261–272. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2011. [doi:10.4230/LIPIcs.STACS.2011.261] 36

[8] JOHN FABEN: The complexity of counting solutions to generalised satisfiability problems modulo k. In *CoRR*, 2008. [arXiv:0809.1836] 37, 51

[9] JOHN FABEN: *The Complexity of Modular Counting in Constraint Satisfaction Problems*. Ph. D. thesis, School of Mathematics, Queen Mary, University of London, 2012. 37

[10] ANDREAS GÖBEL, LESLIE ANN GOLDBERG, AND DAVID RICHERBY: The complexity of counting homomorphisms to cactus graphs modulo 2. *ACM Trans. Comput. Theory*, 6(4):17:1–17:29, 2014. Preliminary version in STACS'14. [doi:10.1145/2635825, arXiv:1307.0556] 37, 40, 48, 49

[11] ANDREAS GÖBEL, LESLIE ANN GOLDBERG, AND DAVID RICHERBY: Counting homomorphisms to square-free graphs, modulo 2. Jan 2015. [arXiv:1501.07539] 37, 49

[12] LESLIE ANN GOLDBERG, MARTIN GROHE, MARK JERRUM, AND MARC THURLEY: A complexity dichotomy for partition functions with mixed signs. *SIAM J. Comput.*, 39(7):3336–3402, 2010. Preliminary versions in STACS'09 and arXiv. [doi:10.1137/090757496] 36

[13] HENG GUO, SANGXIA HUANG, PINYAN LU, AND MINGJI XIA: The complexity of weighted boolean #CSP modulo k. In *Proc. 28th Symp. Theoretical Aspects of Comp. Sci. (STACS'11)*, volume 9 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 249–260. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2011. [doi:10.4230/LIPIcs.STACS.2011.249] 37

[14] PAVOL HELL AND JAROSLAV NEŠETŘIL: On the complexity of H-coloring. *J. Comb. Theory Ser. B*, 48(1):92–110, 1990. [doi:10.1016/0095-8956(90)90132-J] 36, 40

[15] PAVOL HELL AND JAROSLAV NEŠETŘIL: *Graphs and Homomorphisms*. Oxford Univ. Press, 2004. Available from Oxford University Press. 42

[16] CAMILLE JORDAN: Sur les assemblages de lignes. *J. Reine Angew. Math.*, 1869(70):185–190, 1869. [doi:10.1515/crll.1869.70.185] 50

[17] LÁSZLÓ LOVÁSZ: On the cancellation law among finite relational structures. *Period. Math. Hungar.*, 1(2):145–156, 1971. [doi:10.1007/BF02029172] 42

[18] LÁSZLÓ LOVÁSZ: *Combinatorial Problems and Exercises*. North-Holland Publishing Co., Amsterdam-New York, 1979. 42

[19] CHRISTOS H. PAPADIMITRIOU AND STATHIS K. ZACHOS: Two remarks on the power of counting. In *Theoret. Comput. Sci.*, volume 145, pp. 269–276, London, UK, 1982. Springer. [doi:10.1007/BFb0036487] 37, 38

[20] GEORGE PÓLYA: Kombinatorische Anzahlbestimmung für Gruppen, Graphen und chemische Verbindungen. *Acta Math.*, 68(1):145–254, 1937. [doi:10.1007/BF02546665] 50

[21] LESLIE G. VALIANT: The complexity of computing the permanent. *Theoret. Comput. Sci.*, 8(2):189–201, 1979. [doi:10.1016/0304-3975(79)90044-6] 37

[22] LESLIE G. VALIANT: Accidental algorithms. In *Proc. 47th FOCS*, pp. 509–517, Washington, DC, USA, 2006. IEEE Comp. Soc. Press. [doi:10.1109/FOCS.2006.7] 37, 51

AUTHORS

John Faben
School of Mathematical Sciences
Queen Mary, University of London
jdfaben@gmail.com
`http://www.johnfaben.com`

Mark Jerrum
School of Mathematical Sciences
Queen Mary, University of London
m.jerrum@maths.qmul.ac.uk
`http://www.maths.qmul.ac.uk/~mj`

ABOUT THE AUTHORS

JOHN FABEN studied Mathematics at the University of Birmingham. He then spent a year in Edinburgh doing a Masters in Operational Research before going on to do a Ph. D. with Mark Jerrum at Queen Mary in London. The focus of his research there was the complexity of modular counting, particularly in Constraint Satisfaction Problems, but he also enjoyed the existence of the Combinatorics Study Group. He doesn't do research level mathematics any more, but has moved back to Scotland and works for Barclays Bank in Glasgow. He plays both bridge and water polo, and has yet to meet anyone else who can say the same.

MARK JERRUM graduated from Edinburgh University in 1981, where his advisor was Leslie Valiant. He remained at Edinburgh until 2007, when he moved to Queen Mary, University of London. He has a long-term interest in the computational complexity of counting problems, and in randomised algorithms, particularly those based on Markov chain Monte Carlo.