UNIVERSITY OF LIVERPOOL

Computational complexity of approximation of partition functions

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Colin McQuillan

September 2013

Contents

Preface

Chapter 2 is joint work with Leslie Ann Goldberg and Mark Jerrum [63]. Chapter 6 is the full version of joint work with Xi Chen, Martin Dyer, Leslie Ann Goldberg, Mark Jerrum, Pinyan Lu, and David Richerby [34]. Chapter 7 is based on joint work with Andrei Bulatov, Martin Dyer, Leslie Ann Goldberg and Mark Jerrum [22]. Some of the reductions in Chapter 4 were shown to the author by Leslie Ann Goldberg and David Richerby. Everything else is my work, written for my PhD supervised by Leslie Ann Goldberg. Chapter 3 is based on the manuscript [84]. Chapter 5 is based on the manuscript [83]. Those citations are:

[22] Andrei A. Bulatov, Martin E. Dyer, Leslie Ann Goldberg, Mark Jerrum, and Colin McQuillan, *The expressibility of functions on the Boolean domain, with applications to Counting CSPs*, J. ACM (2013), To appear; preprint at arXiv:1108.5288 [cs.CC].

[34] Xi Chen, Martin Dyer, Leslie Ann Goldberg, Mark Jerrum, Pinyan Lu, and Colin McQuillan David Richerby, *The complexity of approximating conservative counting CSPs*, STACS, 2013, full version at arXiv:1208.1783 [cs.CC], pp. 148–159.

[63] Leslie Ann Goldberg, Mark Jerrum, and Colin McQuillan, *Approximating the partition function of planar two-state spin systems*, arXiv:1208.4987 [cs.CC], submitted.

[83] Colin McQuillan, *Degree two approximate Boolean #CSPs with variable weights*, arXiv:1204.5714 [cs.CC], submitted, 2012.

[84] ———, *Approximating Holant problems by winding*, arXiv:1301.2880 [cs.CC], submitted, 2013.

My PhD research was funded by an EPSRC doctoral training grant, via the University of Liverpool's Computer Science Department. [22, 34, 63] were supported by an EPSRC Research Grant "Computational Counting". Xi Chen was also supported by NSF Grant CCF-1139915 and Columbia University start-up funds. The work in [22] was also supported by an NSERC Discovery Grant, and was partially supported by a visit to the Isaac Newton Institute for Mathematical Sciences, under the programme "Discrete Analysis".

Abstract

This thesis studies the computational complexity of approximately evaluating partition functions. For various classes of partition functions, we investigate whether there is an FPRAS: a fully polynomial randomised approximation scheme. In many of these settings we also study "expressibility", a simple notion of defining a constraint by combining other constraints, and we show that the results cannot be extended by expressibility reductions alone. The main contributions are:

- We show that there is no FPRAS for evaluating

$$\sum_{\text{independent sets } I} 312^{|I|}$$

  on planar graphs, unless RP = NP.

- We generalise an argument of Jerrum and Sinclair to give FPRASes for a large class of degree-two Boolean #CSPs.

- We initiate the classification of degree-two Boolean #CSPs where the constraint language consists of a single arity 3 relation.

- We show that the complexity of approximately counting downsets in directed acyclic graphs is not affected by restricting to graphs of maximum degree three.

- We classify the complexity of degree-two #CSPs with Boolean relations and weights on variables.

- We classify the complexity of the problem $\#\mathsf{CSP}(\mathcal{F})$ for arbitrary finite domains when enough non-negative-valued arity 1 functions are in the constraint language.

- We show that not all log-supermodular functions can be expressed by binary log-supermodular functions in the context of #CSPs.

Chapter 1

Introduction

1.1 Gibbs measures and partition functions

We first briefly discuss how the problems considered in this thesis can be seen as a generalisation of computational problems arising from statistical physics. The book [75] discusses this connection in more detail.

Iron behaves as a magnet below a certain temperature. This is a statistical effect of interactions between particles, and the Ising model is a simple model of this effect. Given a finite graph (V, E), the Ising model Hamiltonian (with "interaction" or "coupling constant" 1) is the function $H\colon \{-1,+1\}^V \to \mathbb{R}$ defined by $H(\sigma) = -\sum_{ij\in E} \sigma_i\sigma_j$. The value $H(\sigma)$ is the energy of the configuration σ. The vertices model particles, and adjacent particles get a penalty for being in opposite states.

A Hamiltonian can then be converted to a probability measure. For any value $\beta > 0$ and function $H\colon \Omega \to \mathbb{R}$ where Ω is a finite set, the Gibbs measure is given by the probability mass function p on Ω defined by $p(\sigma) = \frac{1}{Z}e^{-\beta H(\sigma)}$. Here Z is a normalising constant $\sum_{\sigma\in\Omega} e^{-\beta H(\sigma)}$.

Gibbs measures are important for various related reasons. Most directly, they model the behaviour of a physical system in a heat bath - see [68]. Here the value of β is inversely proportional to temperature, and the Ising model on suitable graphs displays ferromagnetism below a certain temperature [75]. Less directly, p is the unique maximiser of the entropy $\sum_{\sigma\in\Omega} -p(\sigma)\log p(\sigma)$ subject to a given mean energy $\overline{H} = \sum_{\sigma\in\Omega} p(\sigma)H(\sigma)$; the value of β depends on $\overline{H}$. According to the principle of maximum entropy (again, see [68]), the Gibbs measure is the correct guess for the distribution of a system with a specified mean energy.

The normalising constant Z is called the partition function. As we will mention in Section 1.5.2, the computational problem of computing Z is often related to the problem of approximately sampling from the Gibbs measure. The behaviour of Z also determines thermodynamic quantities such as $\overline{H} = -\frac{\partial Z}{\partial \beta}$.

For the problems considered in this thesis, the important features of the Ising model are that the configuration σ is a vector of local configurations in some finite domain, and that the energy is a sum of energies with a simple form. We will nearly always work with the weights $\exp(-\beta H(\sigma))$ rather than energies $H(\sigma)$. This eliminates β, simplifying the description of computational problems, and also allows zero weights without having to talk about infinite energies. So for our purposes, a partition function is an expression of the form

$$\sum_{\sigma\colon V \to D} \prod_k F_k(\sigma).$$

For example, this expression specialises to the Ising model partition function described above, as follows: set $D = \{-1, 1\}$, take the product to run over all edges $k = ij$, and set $F_{ij}(\sigma) = e^{-\beta\sigma_i\sigma_j}$.

1.2 Computational complexity

We can consider the partition function for the Ising model as a computational problem $f\colon \Sigma^* \to \mathbb{R}_{\geq 0}$, taking a string encoding a graph to the partition function of the Ising model on that graph (fixing $\beta > 0$, for simplicity). Jerrum and Sinclair [71] gave a type of approximation algorithm for this problem, called a fully polynomial randomised approximation scheme (FPRAS).

A **randomised approximation scheme** for $f\colon \Sigma^* \to \mathbb{R}_{\geq 0}$ is a randomised algorithm that when given an input w and error parameter $\varepsilon > 0$, outputs a random value q satisfying

$$\Pr(e^{-\varepsilon} f(w) \leq q \leq e^{\varepsilon} f(w)) \geq \frac{3}{4}.$$

A randomised approximation scheme is a **fully polynomial randomised approximation scheme (FPRAS)** if it runs in time polynomial in ε^{-1} and the length $|w|$ of w. To represent the error parameter and outputs, ε^{-1} can be specified as a positive integer in binary, and the output can be specified as a ratio of binary integers.

For other problems, such as the anti-ferromagnetic Ising model, it is known that there is an FPRAS if and only if the complexity classes RP and NP are equal. A goal of this thesis is to classify which problems have an FPRAS under various assumptions.

The notion of an FPRAS is quite robust. The number 3/4 can be amplified to $1 - \delta$ using $O(\log \delta^{-1})$ trials and taking the median [73, Lemma 6.1]. For typical problems the dependence on ε is also not important because partition functions are multiplicative: taking k disjoint copies of the input raises the partition function to the power of k, and given q satisfying $e^{-\varepsilon} Z^k \leq q \leq e^{\varepsilon} Z^k$ we know $q^{1/k}$ satisfies $e^{-\varepsilon/k} Z \leq q^{1/k} \leq e^{\varepsilon/k} Z$. However, as we discuss in Chapter 2, this argument fails for logarithms of partition functions.

Following [46], we will study computational problems by trying to find approximation-preserving reductions (AP-reductions) between them. The justification for studying these reductions is that if there is an AP-reduction from f to g, and g has an FPRAS, then f also has an FPRAS. The following definition is a modest generalisation of the definition given in [46], which was restricted to problems in $\#\mathsf{P}$.

Let $f, g\colon \Sigma^* \to \mathbb{R}_{\geq 0}$. An **AP-reduction** $\mathcal{A}$ from f to g is a probabilistic oracle Turing machine [1] satisfying the following requirements. (i) $\mathcal{A}$ takes inputs (w, ε) where $w \in \Sigma^*$ and $\varepsilon > 0$. The run-time of $\mathcal{A}$ is polynomial in $|w|$ and ε^{-1} and the bit-size of the values returned by the oracle (this avoids requiring the oracle to give concise responses). (ii) The oracle calls made by $\mathcal{A}$ are of the form (v, δ), where v is an instance of g and $\delta > 0$ is an error parameter, such that $|v|$ and δ^{-1} are bounded by a polynomial in $|w|$ and ε^{-1} (depending only on $\mathcal{A}$). (iii) If the oracle's outputs meet the specification of a randomised approximation scheme for g, then $\mathcal{A}$ is a randomised approximation scheme for f.

If there is an AP-reduction from f to g we write $f \leq_{\mathrm{AP}} g$. We will say that f *is AP-equivalent to* g, written $f =_{\mathrm{AP}} g$, whenever $f \leq_{\mathrm{AP}} g$ and $g \leq_{\mathrm{AP}} f$. (In [46] this relation is called AP-interreducibility.)

When discussing FPRASes for a problem, it makes sense to also discuss exact evaluation. For exact evaluation, the challenge is to classify each problem as either being in FP or being #P-hard. For example, [71] showed that a version of the problem of evaluating the Ising partition function is #P-hard.

The notion of #P-hardness was introduced in [97]. In this thesis, FP means the set of functions that can be computed by a polynomial-time algorithm. #P is the set of functions $g\colon \Sigma^* \to \mathbb{N}$ such that there is a non-deterministic polynomial-time Turing machine that has exactly $g(w)$ accepting paths when given an input w. A problem f is #P-hard if for each $g \in$ #P there is a polynomial-time Turing reduction from g to f. A problem is #P-complete if it is in #P and is #P-hard. Apart from in Chapter 4, the problems we consider usually take non-integer values, so are not in #P.

1.2.1 #BIS, #PM, #SAT

Our results will often be stated in terms of AP-reductions to or from one of the following problems.

Name. #BIS
Instance. A bipartite graph G.
Output. The number of independent sets in G.

Name. #PM
Instance. A graph G.
Output. The number of perfect matchings in G. (A perfect matching of a graph is a set M of edges, such that every vertex is incident to exactly one edge in M.)

Name. #SAT
Instance. A Boolean formula φ in conjunctive normal form.
Output. The number of satisfying assignments of φ.

[1] A probabilistic oracle Turing machine (POTM) is a randomised algorithm that can make oracle calls. There is some freedom in the type of oracles allowed; for concreteness we can use restricted RAS oracles [8] but allow rational outputs.

All these problems are #P-complete: see [85] for #BIS, and [97] for #PM and #SAT. It is known that #SAT has an FPRAS if and only if RP = NP: see for example the remarks in [46] and [69]. There are easy AP-reductions #$\mathsf{BIS} \leq_{\mathrm{AP}}$ #SAT and #$\mathsf{PM} \leq_{\mathrm{AP}}$ #SAT, and no other reduction between these problems is known. However, there are many reductions involving other problems.

Example 1.1. A downset in a partially ordered set $(P, \preceq)$ is a set $X \subseteq P$ such that for all $x, y \in P$ with $x \leq y$ and $y \in X$, we have $x \in X$. #$\mathsf{Downsets}$ is the problem of counting downsets in a partially ordered set, specified explicitly by a set P and a set $(\preceq) \subseteq P \times P$. Then #$\mathsf{Downsets} =_{\mathrm{AP}}$ #BIS [46]. ⋄

Example 1.2. In [59] it is shown that #PM is AP-equivalent to computing the partition function of the Ising model with negative weights in a certain sense: for any fixed rational $\lambda \in (-1, 0)$, #PM is AP-equivalent to the problem of computing

$$\sum_{\sigma\colon V \to \{0,1\}} \prod_{ij \in E} \lambda^{|\sigma(i) - \sigma(j)|}$$

given a graph (V, E) as input. (This sum happens to always take non-negative values.) ⋄

1.3 Counting constraint satisfaction problems

A #CSP (counting constraint satisfaction problem) is the computational problem of evaluating certain partition functions. For any finite set D, a **weight-function on** D is a function $f\colon D^k \to \mathbb{R}_{\geq 0}$ for some arity $k \geq 0$. A (weighted) **constraint language** $\mathcal{F}$ is a set of weight-functions on a finite domain D. A **constraint** over $\mathcal{F}$ on a finite set V is a pair $\langle (v_1, \ldots, v_k), F \rangle$ where $v_1, \ldots, v_k \in V$ and F is an arity-k function in $\mathcal{F}$. For any finite constraint language $\mathcal{F}$, define a problem

Name. #$\mathsf{CSP}(\mathcal{F})$
Instance. A set of variables V and a list C of constraints over $\mathcal{F}$ on V.
Output. The partition function

$$Z_{V,C} = \sum_{\sigma\colon V \to D} \mathrm{wt}_{V,C}(\sigma)$$

where

$$\mathrm{wt}_{V,C}(\sigma) = \prod_{\langle (v_1,\ldots,v_k), F \rangle \in C} F(\sigma(v_1), \ldots, \sigma(v_k)).$$

To neaten the notation we will drop curly braces and use commas to denote unions of constraint languages, so #$\mathsf{CSP}(\mathcal{F} \cup \{F_1, \ldots, F_k\})$ can be written #$\mathsf{CSP}(\mathcal{F}, F_1, \ldots, F_k)$. Note that we only define $\mathcal{F}$ for finite constraint languages[2], so there is no ambiguity about how elements of $\mathcal{F}$ are represented as input.

[2]We do not use the convention that the phrase "constraint language" means a finite set.

To capture the problem of computing the partition function of the Ising model on multigraphs (for a fixed $\beta > 0$), we take $D = \{-1, 1\}$ and let $\mathcal{F} = \{F\}$ where $F\colon D^2 \to \mathbb{R}$ is defined by $F(x, y) = e^{-\beta xy}$. Given a multigraph (V, E), we can construct a $\#\mathsf{CSP}(\mathcal{F})$ instance with variables V and a constraint $\langle (i, j), F\rangle$ for each edge ij. Conversely, given a $\#\mathsf{CSP}(\mathcal{F})$ instance (V, C) we can construct a multigraph with vertices V and an edge ij for each $\langle (i, j), F\rangle$. In both cases the partition function of the Ising model matches the partition function of the #CSP.

An important special case is when the constraint language consists of relations $R \subseteq D^k$, which we can consider as functions $D^k \to \{0, 1\}$ by taking the characteristic function. A set of relations is called an **unweighted constraint language**. For finite unweighted constraint languages Γ, the problem $\#\mathsf{CSP}(\Gamma)$ can be described as the problem of counting the number of configurations $\sigma\colon V \to D$ that are **satisfying assignments**, that is:

$$(\sigma(v_1), \ldots, \sigma(v_k)) \in R \text{ for all } \langle (v_1, \ldots, v_k), R\rangle \in C.$$

For example, letting $\mathrm{NAND} = \{(0, 0), (0, 1), (1, 0)\}$, the problem $\#\mathsf{CSP}(\mathrm{NAND})$ corresponds to counting independent sets in multigraphs, using the same translations as for the Ising model: the satisfying assignments are the characteristic functions of independent sets.

Apart from Chapter 6, in this thesis we will use the **Boolean domain** $D = \{0, 1\}$. We will study #CSPs as defined above, but we also use #CSPs as a convenient framework for describing computational problems. In particular we will focus on **degree-two #CSPs**: the restriction of a problem of the form $\#\mathsf{CSP}(\mathcal{F})$ to instances where every variable is used at most twice. Certain constraints have the effect of breaking the degree bound; see for example Lemma 4.17. This means that restricting the degree of instances gives a more general class of problems, in a sense.

The restriction to read-twice #CSPs, where every variable appears exactly twice, gives the class of **Holant problems**. [3] These have useful algebraic properties that allow reductions called holographic reductions [32, 99]. Holant problems are often described in terms of the dual constraint graph: the multigraph whose vertices are constraints and with an edge cc' for each variable whose two occurrences are in the constraints c and c'. In this way Holant problems are similar to problems such as $\#\mathsf{PM}$, where edges can be seen as variables, and each vertex enforces a constraint (that exactly one incident edge is in the matching).

1.4 Expressibility

Expressibility is a limited notion of reduction between computational problems like #CSPs. The idea is that constraints can be combined to simulate a new weight-function

[3] The author learnt this CSP definition of Holant problems from [101, Proposition 9.2].

$F\colon D^U \to \mathbb{R}_{\geq 0}$ of the form

$$F(\sigma) = \sum_{\sigma'} \prod_{\langle (v_1,\dots,v_k),F\rangle \in C} F(\sigma'(v_1),\dots,\sigma'(v_k))$$

where C is a list of constraints on a variable set $V \supseteq U$, and σ is a configuration $U \to D$, and the sum ranges over extensions of σ to $\sigma' \in D^V$.

If C only uses functions from a set $\mathcal{F}$, we say F is *pps-definable over* $\mathcal{F}$. In this case there is an AP-reduction $\#\mathsf{CSP}(\mathcal{F} \cup \{F\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}(\mathcal{F})$ [22, Lemma 17]. This gives a reduction, which we will call an "expressibility reduction". Throughout this thesis we will try to show that our results cannot be extended by a reduction of this type.

There are variants of pps-definability allowing certain limits [22]. To study degree-two #CSPs we need a notion of expressibility with degree restrictions. In each case the important property is that we get an expressibility reduction between relevant #CSP-like problems.

1.5 Background

To place the results of this thesis in context, we will briefly describe relevant previous results.

1.5.1 Spin systems

Two-state spin systems are a generalisation of the Ising model to an arbitrary vertex function $\varphi\colon \{0,1\} \to \mathbb{R}_{\geq 0}$ and edge function $\overline{\varphi}\colon \{0,1\}^2 \to \mathbb{R}_{\geq 0}$ satisfying $\overline{\varphi}(0,1) = \overline{\varphi}(1,0)$. (These can be normalised by setting $\varphi(0) = \overline{\varphi}(0,1) = 1$.) The partition function on a graph (V,E) is

$$Z = \sum_{\sigma\colon V \to \{0,1\}} \left(\prod_{v \in V} \varphi(\sigma(v)) \right) \left(\prod_{uv \in E} \overline{\varphi}(\sigma(u),\sigma(v)) \right).$$

For fixed $(\varphi,\overline{\varphi})$ we can then study the problem of evaluating Z. There is an almost complete classification of the complexity of approximating the partition function of antiferromagnetic ($\overline{\varphi}(0,0)\overline{\varphi}(1,1) \leq \overline{\varphi}(0,1)\overline{\varphi}(1,0)$) two-state spin systems on general and bounded-degree graphs. We review these results in more detail in Chapter 2. An important special case is that there is an FPRAS for counting independent sets in graphs of maximum degree five due to Weitz [100] (in fact, a deterministic FPRAS), but no FPRAS for counting independent sets in regular graphs of degree six unless RP = NP [92]. Note that we get the number of independent sets by setting $\varphi(0) = \varphi(1) = \overline{\varphi}(0,0) = \overline{\varphi}(0,1) = \overline{\varphi}(1,0) = 1$ and $\overline{\varphi}(1,1) = 0$.

In a different direction, the Potts model generalises the Ising model to larger sets of states than just $\{-1,1\}$, while the Tutte polynomial generalises the Potts model to a two-variable polynomial. The computational complexity of evaluating these partition functions has been studied [59, 61, 62]. In particular Goldberg and Jerrum [62] determined the computational complexity of evaluating some points of the Tutte polynomial on planar graphs.

1.5.2 Sampling and Markov chains

In [73] it is shown that for the class of "self-reducible" problems, counting and sampling are related: there is an FPRAS for a problem if and only if there is an algorithm that samples from the Gibbs distribution, to within exponentially small error.

The FPRAS for the ferromagnetic Ising model in [71] mentioned earlier uses this approach, together with a rapidly converging Markov chain for sampling. (Though because they used a transformation called the high-temperature expansion, they did not sample from configurations $\sigma\colon V \to \{-1, 1\}$ but from spanning subgraphs $X \subseteq E$.) They then bounded the mixing time using an analysis we will refer to as the cycle-unwinding canonical paths argument.

Another important type of Markov chain analysis is the path coupling technique (which we use for Theorem 5.16). There are also examples of FPRASes that do not use Markov chain techniques: see Karp and Luby's algorithm for #DNFSAT [74], and Weitz's algorithm mentioned above.

1.5.3 #CSP and Holant classifications

There have been several results on the problem $\#\mathsf{CSP}(\mathcal{F})$. For exact evaluation there is a dichotomy into FP and $\#\mathsf{P}$-hard even for arbitrary finite domains, even if we allow the functions in $\mathcal{F}$ to take complex algebraic values [23]. We will say a bit more about the history of exact evaluation of #CSPs in Chapter 6, where we study the approximation evaluation of #CSPs with no extra restrictions.

For approximate evaluation, there is a trichotomy for all (finite) unweighted constraint languages Γ on the Boolean domain: the problem $\#\mathsf{CSP}(\Gamma)$ can either be evaluated exactly in polynomial time, or is AP-equivalent to $\#\mathsf{BIS}$, or is AP-equivalent to $\#\mathsf{SAT}$ [48]. A similar trichotomy was given for degree-six #CSPs in [51] under the assumption $\{(0)\}, \{(1)\} \in \Gamma$, with partial results for degree-three #CSPs.

There are several results on exact and approximate evaluation of Holant problems. Holant problems are often studied allowing complex-valued weight-functions (called signatures in this context).

There is a dichotomy for the complexity of exactly evaluating $\mathsf{Holant}^*(\mathcal{F})$, which is $\mathsf{Holant}(\mathcal{F})$ where all arity 1 complex-weighted constraints are allowed [27]. (Similarly to the conservative CSPs mentioned below.) Other work on exact evaluation of Holant problems has focussed on symmetric functions, so $F(x_1, \dots, x_k)$ only depends on $x_1 + \cdots + x_k$; see [31, 66] for recent results.

Yamakami [103] studied the approximation complexity of $\mathsf{Holant}^*(F)$ (referring to it as $\#\mathsf{CSP}_2^*$) where F is in a certain set of arity 3 functions. The conclusion is that each problem can either be evaluated exactly in polynomial time or is #P-hard to approximate. In the same setting there are results for higher degree bounds [101, 102].

1.5.4 Decision and optimisation

Given a set Γ of relations on a finite domain D, CSP(Γ) is the problem of recognising which instances of #CSP(Γ) have a satisfying assignment. It is an open problem to classify the complexity of CSP(Γ), but there is a dichotomy into polynomial-time and NP-complete problems when $|D|$ is three [18]. Bounded-degree Boolean CSPs were studied by Feder [52] and Dalmau and Ford [41]. We will use Feder's ideas in Chapter 5.

Given a set Φ of functions $f\colon D^k \to \mathbb{Q} \cup \{+\infty\}$, the problem VCSP($\Phi$) ("valued CSP") is the optimisation problem of minimising

$$\sum_{\langle (v_1,\ldots,v_k),F\rangle} f(\sigma(v_1),\ldots,\sigma(v_k))$$

over functions $\sigma\colon V \to D$, given a set of variables V and a list of constraints C using Φ. This gives a generalisation of CSPs.

This is similar to a #CSP but with addition replaced by minimisation and multiplication replaced by addition; a valued CSP can be thought of as the problem of finding the configuration of maximum weight, or minimum energy. We will use this connection in Chapter 6.

The set of functions that can be expressed by a CSP or valued CSP has a useful dual description in terms of inequalities called weighted polymorphisms [37]. No such duality is known to exist for #CSPs. Analogous inequalities do however turn out to be useful for studying #CSPs in some cases — we use this connection in Chapter 6.

1.5.5 Conservative constraint languages

For both CSPs and VCSPs, the *conservative* case has been completely classified. In a conservative CSP, all arity 1 relations are allowed. In a conservative valued CSP or #CSP, all non-negative arity 1 functions are allowed. (Formalising this with finite constraint languages requires a bit of circumlocution.)

Bulatov *et al.* [22] gave a classification of Boolean conservative #CSPs. An interesting case is when every function $F\colon \{0,1\}^n \to \mathbb{R}_{\geq 0}$ in the constraint language $\mathcal{F}$ is **log-supermodular**:

$$F(\mathbf{x} \vee \mathbf{y})F(\mathbf{x} \wedge \mathbf{y}) \geq F(\mathbf{x})F(\mathbf{y}) \text{ for all } \mathbf{x},\mathbf{y} \in \{0,1\}^n$$

where $\mathbf{x}\vee\mathbf{y}$ and $\mathbf{x}\wedge\mathbf{y}$ are the joint and meet respectively, defined by $(\mathbf{x}\vee\mathbf{y})_i = \max(x_i, y_i)$ and $(\mathbf{x} \wedge \mathbf{y})_i = \min(x_i, y_i)$ $(1 \leq i \leq n)$.

In this case, either #CSP($\mathcal{F}$) is in FP or there is a finite set S of non-negative arity 1 functions such that #BIS $\leq_{\mathrm{AP}}$ #CSP($\mathcal{F} \cup S$) (we import this result as Proposition 5.8).

But it is not known whether $\#\mathsf{BIS} =_{\mathrm{AP}} \#\mathsf{CSP}(\mathcal{F} \cup S)$. It is known that any function that can be pps-defined by log-supermodular functions is log-supermodular.

Bulatov introduced the name "conservative" and gave a dichotomy for conservative CSPs [17]. Kolmogorov and Živný established a dichotomy for conservative valued CSPs [77]. (Because the definitions of conservativity are different, the conservative valued CSP dichotomy does not generalise the conservative CSP dichotomy.) We will discuss the VCSP result more in Chapter 6.

1.6 Summary of results

This section gives a brief summary of the thesis. We will focus here on stating the most important consequences with as few preliminaries as possible rather than giving the most powerful or general statements. In particular, this means we do not state explicit criteria in classifications. We will also mention only the most relevant previous work for context, leaving a fuller discussion for the introductions to each chapter.

Approximating the partition function of planar two-state spin systems

In Chapter 2 we show there is no FPRAS for evaluating

$$\sum_{\text{independent sets } I \subseteq V} 312^{|I|}$$

on planar graphs, unless RP = NP. The result extends to other two-state spin systems. We use a "contour" argument, which is a classic argument in statistical physics but is used here to analyse a gadget. There is relevant previous work on spin systems on not-necessarily-planar graphs by Sly and Sun [93], and on the Tutte polynomial on planar graphs [62], but we study the case of spin systems on planar graphs.

We also show a striking feature of these spin systems on planar graphs: there is a polynomial-time randomised approximation scheme for the logarithm of the partition function.

Approximating Holant problems by winding

In Chapter 3 we generalise the canonical paths cycle-unwinding argument of Jerrum and Sinclair [71] to Holant problems using "windable" functions, and show that this gives an FPRAS for functions that are additionally "strictly terraced". Both of these classes are closed under a natural notion of expressibility.

A consequence is that there is an FPRAS for the following problem.

Name. $\#\mathsf{ParityNAE}$
Instance. A multigraph G in which each vertex is labelled Even, Odd, or NAE
Output. The number of subsets $F \subseteq E(G)$ such that:

- each Even vertex has an even number of incident edges in F

- each Odd vertex has an odd number of incident edges in F
- each NAE vertex has at least one incident edge in F and at least one incident edge in $E(G) \setminus F$

The most relevant previous work is that of Jerrum and Sinclair [71].

We then ask whether windability is the same as expressibility by "matchings circuits", a natural notion of expressibility for #PM, and give a positive answer for functions of arity three.

Holant problems with arity three relations, and counting downsets

In Chapter 4 we initiate the classification of Boolean degree-two #CSPs (and Holant problems) where the constraint language consists of a single arity three relation. We relate many problems to #PM, #BIS and #SAT. The most relevant previous work is the study of bounded-degree #CSPs in [51], which looked at degrees greater than two.

Even the complexity of exact evaluation does not seem to have been previously published. We give a dichotomy in FP and #P, for degree-two #CSPs and Holant problems with finite unweighted constraint languages using relations of arity at most three. The proof is an application of a dichotomy for real-weighted symmetric constraints due to Huang and Lu [67].

We also show that counting downsets in directed acyclic graphs of maximum degree three is AP-equivalent to #BIS.

Read-twice #CSPs with variable weights

In Chapter 5 we study the complexity of degree-two #CSPs with a finite unweighted constraint language Γ and "variable weights", a #CSP analogue of edge weights for perfect matchings:

Name. $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma)$
Instance. A tuple (V, C, w) where:

- V is a finite set of variables.
- C is a finite set of constraints over Γ on V such that each variable $v \in V$ is used at most twice in C
- w is a function from V to $\mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$, where rationals are specified as ratios of binary integers.

Output. The sum of $\prod_{v \in V} w(v)_{\mathbf{x}(v)}$ over satisfying assignments $\mathbf{x}\colon V \to \{0,1\}$ of C, where we index the pair $w(v)$ from zero: $w(v)_{\mathbf{x}(v)}$ denotes the first element of $w(v)$ when $\mathbf{x}(v) = 0$, and the second element when $\mathbf{x}(v) = 1$.

For example the problem $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathrm{PM}_3)$ with $\mathrm{PM}_3 = \{(0,0,1), (0,1,0), (1,0,0)\}$ corresponds to counting perfect matchings in cubic multigraphs with non-negative rational edge weights specified in binary (except that we allow "edges" of order one).

The main result is that for each Γ, the problem $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma)$ is either in FP, or $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma)$, or $\#\mathsf{BIS} =_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma)$. So assuming neither $\#\mathsf{BIS}$ nor $\#\mathsf{PM}$ have an FPRAS, there is no FPRAS for any problem $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma)$, except when there is an exact algorithm.

Related previous results, mentioned in Section 1.5.3, are the trichotomy for degree-six #CSPs, the dichotomy for *exact* evaluation of Holant^* problems, and partial results for approximate evaluation of Holant^* problems.

We also give partial results for weighted constraint languages, leaving open the possibility that some degree-two #CSPs with variable weights for constraint languages using "terraced basically binary" functions could have an FPRAS even if $\#\mathsf{PM}$ and $\#\mathsf{BIS}$ do not. We show that this class is closed under a natural notion of expressibility, which shows that the classification cannot be extended by expressibility reductions.

The complexity of approximating conservative counting CSPs

In Chapter 6 we give a classification for conservative #CSPs with an arbitrary finite domain. As mentioned previously, conservative #CSPs are awkward to formalise. So the following description of results is slightly imprecise, with formal statements given in Chapter 6.

We show that each problem is either in FP or admits an AP-reduction from $\#\mathsf{BIS}$. We further sub-divide the $\#\mathsf{BIS}$-hard case, by showing that each problem either AP-reduces to a Boolean log-supermodular #CSP, or is AP-equivalent to $\#\mathsf{SAT}$. Finally, we give a full trichotomy for the arity-2 case, where each problem is either in FP, or is AP-equivalent to $\#\mathsf{BIS}$, or is AP-equivalent to $\#\mathsf{SAT}$.

The most relevant previous works are the classification of (approximating) conservative Boolean #CSPs by Bulatov et al. [22], and the dichotomies for exact evaluation of #CSPs which we will discuss in Chapter 6, and Kolmogorov and Živný's classification of conservative VCSPs [77].

LSM is not generated by binary functions

In Chapter 7 we show that binary log-supermodular functions cannot pps-define all arity four log-supermodular functions in the setting of #CSPs. Živný, Cohen, and Jeavons studied the expressibility of binary submodular functions in the setting of valued CSPs. In particular they showed that binary submodular functions do not express all arity four submodular functions in that setting.

1.7 Layout and notation

The chapters can be read independently for the most part, with a few dependencies:

- the study of arity three relations in Chapter 4 uses results from Chapter 3: the definition of windable functions, Theorem 3.4, Theorem 3.5, and Theorem 3.6

- the study of degree-two Boolean #CSPs with variable weights in Chapter 5 refers to Proposition 4.4 (in the discussion) and Lemma 4.15
- Chapter 7 relies on the definition of functional clones given in Chapter 6

Except in Chapter 6, the domain D is always $\{0,1\}$. We defined Holant problems as read-twice #CSPs. But in Chapter 3 we use an alternate definition in terms of annotated graphs we call circuits. This allows us to use the language of graph theory throughout that chapter.

In Chapters 6 and 7, the input to a #CSP is specified as a *pps-formula*, using *atomic formulas* $F(v_1,\dots,v_k)$ instead of constraints $\langle(v_1,\dots,v_k),F\rangle$. This is just a notational difference, matching the conventions in [22] and [77]. (Pps-formulas are essentially the same as the $\mathbb{N}$-formulas defined in 5.2.1.)

There are some common definitions mentioned below, and each chapter uses some additional specialised definitions.

1.7.1 Indexing

For Chapters 3 to 5, it is very useful to consider relations and weight-functions defined using $\{0,1\}^J$ instead of $\{0,1\}^k$, for finite sets J. We consider $\{0,1\}^k$ and $\{0,1\}^{\{1,\dots,k\}}$ to be the same (here $k \geq 0$).

This does not change the definition of $\#\mathsf{CSP}(\mathcal{F})$: by definition, constraint languages exclusively use standard weight-functions, that is, weight-functions defined on $\{0,1\}^k$ for some non-negative integer k. Chapter 3 also uses a special notation "$\{0,1\}^{k+J}$" — see Section 3.2. However, Chapters 6 and 7 exclusively use standard weight-functions.

If $\mathbf{x} \in D^n$ and $\mathbf{y} \in D^m$ $(n,m \geq 0)$ then $(\mathbf{x},\mathbf{y})$ denotes the concatenation $(x_1,\dots,x_n,y_1,\dots,y_m)$, which is an element of D^{n+m}. However, if $\mathbf{x} \in D^I$ and $\mathbf{y} \in D^J$ for some disjoint finite sets I,J, we let $(\mathbf{x},\mathbf{y})$ denote the unique common extension of $\mathbf{x}$ and $\mathbf{y}$ to a configuration of $I \cup J$.

The Cartesian product $R \times S$ of relations $R \subseteq D^n$ and $S \subseteq D^m$ $(n,m \geq 0)$ is $\{(\mathbf{x},\mathbf{y}) \mid \mathbf{x} \in R, \mathbf{y} \in S\} \subseteq D^{n+m}$. The Cartesian product of relations $R \subseteq D^I$, $S \subseteq D^J$, when I and J are finite sets, is also defined as $\{(\mathbf{x},\mathbf{y}) \mid \mathbf{x} \in R, \mathbf{y} \in S\}$, which is then a subset of $D^{I \cup J}$.

1.7.2 Common definitions

We use the convention that $\mathbb{N}$ is the set of non-negative integers, including zero. As usual, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Q}_{\geq 0}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{C}$ denote the sets of integers, rationals, non-negative rationals, reals, non-negative reals, and complex numbers respectively.

We will always write $A \subseteq B$ and $A \subsetneq B$ to denote subsets and proper subsets respectively, avoiding the contested notation $\subset$. However, we will also write $A \subset B$ when it is obvious that A cannot be B, specifically when A is declared to be finite and B is declared to be infinite. The restriction of a function σ to a set X is denoted $\sigma|_X$ throughout, except in Chapter 2 where it is denoted $\sigma(X)$ (so that vertical lines can be reserved for conditional probabilities). We will implicitly identify subsets $R \subseteq S$ with their characteristic function $R\colon S \to \{0,1\}$, but to help make the definitions clearer in Chapter 3, we write the characteristic function in boldface.

Here are some other definitions that are useful in more than one chapter.

A **copy** of $F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ is a function $G\colon \{0,1\}^I \to \mathbb{Q}_{\geq 0}$ of the form $G(\mathbf{x}) = F(\mathbf{x} \circ \pi)$ for some bijection $\pi\colon J \to I$ (here $\mathbf{x} \in \{0,1\}^I$, so $\mathbf{x} \circ \pi \in \{0,1\}^J$). In other words, we permute the variables. For all $F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$, define $\lambda F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ by $(\lambda F)(\mathbf{x}) = \lambda F(\mathbf{x})$. A **constant multiple** of F is any function of the form λF.

Define $\mathbf{x} \oplus \mathbf{y} \in \{0,1\}^J$ by $(\mathbf{x} \oplus \mathbf{y})_i \equiv x_i + y_i \pmod 2$. For all $\mathbf{x} \in \{0,1\}^J$ define $\overline{\mathbf{x}} \in \{0,1\}^J$ by $\overline{x}_i = 1 - x_i$.

We denote the all-zeros configuration of a finite set V by $\underline{0}$, and the all-ones configuration by $\underline{1}$. We give common relations a name: $\mathrm{EQ}_k = \{\underline{0}, \underline{1}\} \subseteq \{0,1\}^k$, $\mathrm{NEQ} = \{(0,1),(1,0)\}$, $\mathrm{PIN}_0 = \{(0)\}$, $\mathrm{PIN}_1 = \{(1)\}$, $\mathrm{NAND} = \{(0,0),(0,1),(1,0)\}$, $\mathrm{OR} = \{(0,1),(1,0),(1,1)\}$, $\mathrm{IMP} = \{(0,0),(0,1),(1,1)\}$, and $\mathrm{PM}_3 = \{(0,0,1),(0,1,0),(1,0,0)\}$. The **pinning** of a weight-function $F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ by $\mathbf{p} \in \{0,1\}^I$ ($I \subseteq J$) is the weight-function $F(\mathbf{p},\cdot)\colon \{0,1\}^{J \setminus I} \to \mathbb{Q}_{\geq 0}$ defined by $(F(\mathbf{p},\cdot))(\mathbf{x}) = F(\mathbf{x},\mathbf{p})$.

For any list C of constraints, the **degree** $\deg_C(v)$ of a variable v is the number of times it is used in total in C. Note that variables can be used multiple times in a constraint. For example, a constraint $\langle (x,y,y), R \rangle$ contributes two to the degree of y. Let $\#\mathsf{CSP}_{\leq d}(\Gamma)$ denote the restriction of $\#\mathsf{CSP}(\Gamma)$ to instances in which every variable has degree at most d. (A more general class of problems $\#\mathsf{CSP}_K^W(\mathcal{F})$ is defined in Chapter 5.)

IMconj is the set of relations that can be written as a conjunction of implications (which in this context become less-than-or-equal relations) and constants. For example the relation $\{\mathbf{x} \in \{0,1\}^4 \mid x_1 \leq x_2 \leq x_3 \text{ and } x_2 = 1\}$ is in IMconj. A relation $R \subseteq \{0,1\}^k$ is **affine** if for all x, y, z in R the configuration $x \oplus y \oplus z \in \{0,1\}^k$, defined by $(x \oplus y \oplus z)_i = x_i + y_i + z_i \pmod 2$ for all $1 \leq i \leq k$, is also in R. We denote the set of all log-supermodular weight-functions by LSM.

For all $F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$, we define $\overline{F}\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ by $\overline{F}(\mathbf{x}) = F(\overline{\mathbf{x}})$ (for all $\mathbf{x} \in \{0,1\}^J$). For all $FG\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$, we define FG by $(FG)(\mathbf{x}) = F(\mathbf{x})G(\mathbf{x})$ (for all $\mathbf{x} \in \{0,1\}^J$). In particular, $F\overline{F}\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ satisfies $(F\overline{F})(\mathbf{x}) = F(\mathbf{x})F(\overline{\mathbf{x}})$. In Chapter 7 we also denote $F\overline{F}$ by $F^\star$.

For any matrix $T \in R^{2\times 2}$ where R is $\mathbb{Q}_{\geq 0}$, $\mathbb{Q}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}$ or $\mathbb{C}$ (or more generally any semiring), and any $F\colon \{0,1\}^J \to \mathbb{R}$ where J is a finite set, define $T^{\otimes J}F\colon \{0,1\}^J \to \mathbb{R}$ by

$$(T^{\otimes J}F)(\mathbf{z}) = \sum_{\mathbf{y} \in \{0,1\}^J} \left(\prod_{i \in J} T_{z_i, y_i} \right) F(\mathbf{y}) \qquad (\mathbf{z} \in \{0,1\}^J).$$

Here the rows and columns of T are considered to be indexed by $\{0,1\}$.

1.7.3 AP-reduction compatibility

Some of the papers we cite use a different definition for AP-reductions, where an AP-reduction may assume the oracle provides integers. For example, [48, Theorem 3] gives (among other things) a reduction $\mathcal{A}$ from #BIS to #CSP(IMP). But that reduction is not guaranteed to be an AP-reduction in our sense, because $\mathcal{A}$ is only proved correct under the assumption that the oracle always provides an integer.

Throughout the thesis, we will cite reductions such as #$\mathsf{BIS} \leq_{\mathrm{AP}}$ #CSP(IMP) without mentioning the difference in definitions. To justify this we need to show that oracle calls can be adapted from providing rationals to providing integers. A similar argument is given in the proof of [46, Theorem 3].

We are given a POTM $\mathcal{A}$ which satisfies all the requirements of an AP-reduction from some function f to an integer-valued function g, under the extra assumption that the oracle always returns an integer. We modify $\mathcal{A}$ in the following way to give an AP-reduction $f \leq_{\mathrm{AP}} g$.

To simulate any oracle call (v, δ), with $0 < \delta < 1/2$ say, we call the real oracle with $(v, \delta/16)$ and return a nearest integer to the result q. Denote this integer by $[q]$, and denote $g(v)$ by Z. With probability $3/4$ we have $e^{-\delta/32} Z \leq q \leq e^{\delta/32} Z$. There are then two cases. If $Z < 8/\delta$ then $Z(1 - e^{-\delta/32})$ and $Z(e^{\delta/32} - 1)$ are each less than $1/2$, so $Z - 1/2 < q < Z + 1/2$. If $Z \geq 8/\delta$ then $q \geq 4/\delta$, so $e^{-\delta/2} q \leq q - 1/2 \leq [q] \leq q + 1/2 \leq e^{\delta/2} q$. In either case $e^{-\delta} Z \leq [q] \leq e^{\delta} Z$.

Chapter 2

Approximating the partition function of planar two-state spin systems

(This chapter consists of [63] with a modified introduction, and with references changed to the published version of [104].)

We consider the problem of approximating the partition function of the hard-core model on planar graphs of degree at most 4. We show that when the activity λ is sufficiently large, there is no fully polynomial randomised approximation scheme for evaluating the partition function unless NP = RP. The result extends to a nearby region of the parameter space in a more general two-state spin system with three parameters. We also give a polynomial-time randomised approximation scheme for the logarithm of the partition function.

2.1 Introduction

A spin system is a model of particle interaction on a graph, generalising the Ising model. Every vertex of the graph is assigned a state, called a spin. A *configuration* assigns a spin to every vertex, and the weight of the configuration is determined by interactions of neighbouring spins.

In this chapter, we consider the following two-spin model, which applies to spin systems on a graph $G = (V, E)$. The model has three parameters, β, γ and λ. It is easiest to view these as non-negative rationals for now — we will be slightly more general later. A configuration $\sigma\colon V(G) \to \{0, 1\}$ is an assignment of the two spins "0" and "1" to the vertices in V. The configuration σ has a *weight* $w_G(\sigma)$, which depends upon β, γ and λ. Let $b(\sigma)$ denote the number of edges (u, v) of G with $\sigma(u) = \sigma(v) = 0$, let $c(\sigma)$ be the number of edges (u, v) of G with $\sigma(u) = \sigma(v) = 1$ and let $\ell(\sigma)$ be the number of vertices u of G with $\sigma(u) = 1$. Then $w_G(\sigma) = \beta^{b(\sigma)}\gamma^{c(\sigma)}\lambda^{\ell(\sigma)}$, where here and throughout the chapter we use the convention that $0^0 = 1$. The *partition function* of the model is given by

$$Z_{\beta,\gamma,\lambda}(G) = \sum_{\sigma:V(G)\to\{0,1\}} w_G(\sigma).$$

Two important special cases are

- the case $\beta = 1$, $\gamma = 0$, which is the *hard-core model*, and
- the case $\beta = \gamma$, which is the *Ising model*.

The *hard-core model* [5] is a model of a gas in which vertices are either occupied by a particle (in which case they have spin 1) or unoccupied (in which case they have spin 0). The particles cannot overlap and adjacent vertices are close together, hence $\gamma = 0$. The Ising model is a model of ferromagnetism. In this chapter we study the hard-core model and a region of nearby two-state spin systems.

2.1.1 Previous work

The partition function factorises in the following way when $\beta\gamma = 1$. We have $w_G(\sigma) = \beta^{b(\sigma)-c(\sigma)}\lambda^{\ell(\sigma)}$ for any configuration σ, and

$$b(\sigma) - c(\sigma) = \sum_{v:\sigma(v)=0} \deg(v)/2 - \sum_{v:\sigma(v)=1} \deg(v)/2$$

where $\deg(v)$ denotes the degree of a vertex v. So $Z_{\beta,\gamma,\lambda}$ is

$$\sum_{\sigma:V(G)\to\{0,1\}} \beta^{\sum_{v:\sigma(v)=0}\deg(v)/2-\sum_{v:\sigma(v)=1}\deg(v)/2}\lambda^{\ell(\sigma)} = \prod_{v\in V(G)} (\beta^{\deg(v)/2} + \lambda\beta^{-\deg(v)/2}),$$

which is easy to evaluate.

In other cases, the complexity of evaluation has been studied in detail. When $\lambda = 1$, the problem of computing the partition function on planar Δ-regular graphs is called Pl-Hol$_\Delta(a, b)$ in [30], where a corresponds to β and b corresponds to γ. There is a dichotomy [30, Theorem 1]: for non-negative a, b, the problem Pl-Hol$_\Delta(a, b)$ can be computed in polynomial time in the trivial cases $ab = 1$ and $a = b = 0$, and in the case of the Ising model with no external field, $a = b$. In all other cases the problem is #P-hard.

A standard transformation extends this dichotomy to arbitrary $\lambda > 0$. Consider a configuration $\sigma\colon V(G) \to \{0, 1\}$ of a planar Δ-regular graph G. Counting the number of edges adjacent to a "1" spin in two ways, we have $\Delta\ell(\sigma) = 2c(\sigma) + (|E(G)| - b(\sigma) - c(\sigma))$. Therefore,

$$Z_{\beta,\gamma,\lambda}(G) = \lambda^{|E(G)|/\Delta} Z_{\beta\lambda^{-1/\Delta},\gamma\lambda^{1/\Delta},1}(G),$$

which is as hard to compute as Pl-Hol$_\Delta(\beta\lambda^{-1/\Delta}, \gamma\lambda^{1/\Delta})$. Suppose β and γ are not both 0. Unless $\lambda = 1$, we have either $\beta\lambda^{-1/3} \neq \gamma\lambda^{1/3}$ or $\beta\lambda^{-1/4} \neq \gamma\lambda^{1/4}$. If $\beta\gamma \neq 1$ then in either case, we can conclude from above that evaluating $Z_{\beta,\gamma,\lambda}(G)$ is #P-hard when the input G is restricted to be a planar graph of degree at most 4.

Since the complexity of exactly evaluating the partition function is intractable, much effort has focussed on the difficulty of *approximately* evaluating the partition function for a given set of parameters β, γ and λ.

The complexity of approximating the partition function of the hard-core model and the Ising model in general (not necessarily planar) graphs is well-understood. The *Gibbs measure* is the distribution on configurations $\sigma\colon V(G) \to \{0,1\}$ in which the probability of configuration σ is proportional to $w_G(\sigma)$. This notion of Gibbs measure extends to certain infinite graphs, for example infinite regular trees, where it may or may not be unique. For the hard-core model, there is a critical point $\lambda_c(\Delta) = (\Delta-1)^{\Delta-1}/(\Delta-2)^{\Delta}$ such that the infinite Δ-regular tree has a unique Gibbs measure if and only if $\lambda \leq \lambda_c$. An important result of Weitz [100] showed that, in *every* graph with maximum degree at most Δ, the correlations between spins in the hard-core model decay rapidly with distance as long as $\lambda \leq \lambda_c$. As a result, he gives [100, Corollary 2.8] a fully-polynomial (deterministic) approximation scheme (FPTAS) for evaluating the hard-core partition function on graphs of degree at most Δ for any $\lambda < \lambda_c$. By contrast, Sly and Sun [93, Theorem 1] (see also the earlier hardness results of Sly [92] and Galanis et al. [55]) show that, unless NP = RP, there is no fully-polynomial randomised approximation scheme (FPRAS) on Δ-regular graphs (for $\Delta \geq 3$) for any $\lambda > \lambda_c(\Delta)$. Thus, the difficulty of approximation is resolved, apart from at the boundary $\lambda = \lambda_c(\Delta)$.

We say that the two-spin model is *ferromagnetic* if $\beta\gamma > 1$ and *antiferromagnetic* if $\beta\gamma < 1$. For the antiferromagnetic Ising model, Sinclair et al. [91, Corollary 1.2] show that there is an FPTAS for evaluating the Ising partition function on graphs of degree at most Δ for any choice of parameters β and λ which is in the interior of the uniqueness region of the Δ-ary tree. By contrast, Sly and Sun [93, Theorem 2] show that, unless NP = RP, there is no FPRAS on Δ-regular graphs (for $\Delta \geq 3$) if β and λ are outside the uniqueness region. (So, once again, the situation is fully resolved, apart from the boundary.) The result of Sinclair et al. [91, Corollary 1.3] extends to general anti-ferromagnetic two-state spin systems in regular graphs, and also in a somewhat wider class of graphs.

For general anti-ferromagnetic two-state spin systems, the best positive result that is known is due to Li, Lu, and Yin [79]. They use a stronger notion of correlation decay than Weitz, which enables them to obtain an FPTAS, even for graphs with unbounded degree. They show [79, Theorem 1.2] that for any finite $\Delta \geq 3$, or for $\Delta = \infty$, there is an FPTAS for the partition function of the two-state spin system on graphs of maximum degree at most Δ if the parameters of the system are antiferromagnetic, and for every $d \leq \Delta$, they lie in the interior of the uniqueness region of the infinite d-regular tree. By contrast [79, Theorem 1.3], the results of Sly and Sun imply that, for any finite $\Delta \geq 3$, or for $\Delta = \infty$, unless NP = RP, there is no FPRAS for the partition function of the two-state spin system on graphs of maximum degree at most Δ if the parameters of the system are antiferromagnetic, and for some $d \leq \Delta$, they lie outside the interior of the uniqueness region of the infinite d-regular tree. Thus, the approximation complexity is resolved in the antiferromagnetic case, apart from at the boundaries of the uniqueness regions. Note that the result of Sun and Sly was independently discovered by Galanis, Štefankovič and Vigoda [56] for the case $\lambda = 1$.

The situation is not completely resolved in the ferromagnetic case. Building on Jerrum and Sinclair's FPRAS for the ferromagnetic Ising model [71], Goldberg, Jerrum and Paterson [64] gave an FPRAS for the ferromagnetic two-spin model which applies if $\beta \geq \gamma$ and $\lambda \leq \sqrt{\beta/\gamma}$ (or, equivalently, if $\beta \leq \gamma$ and $\lambda \geq \sqrt{\beta/\gamma}$). The approximation applies without these constraints on the parameters if the input is a regular graph.

For the hard-core model, an important issue which arises in statistical physics is approximating the partition function for planar graphs, including regular lattices. While there do not seem to be any other hardness results known for this problem, the complexity of particular algorithms have been studied. For example, Randall [86] showed that a particular MCMC algorithm provides a bad approximation on subsets of $\mathbb{Z}^2$, because Glauber dynamics mixes slowly when $\lambda \geq 8.066$. (By contrast, results of Restrepo et al. [87] showed that the mixing time is $O(n \log n)$ when $\lambda < 2.3883$, and that Weitz's algorithm [100] gives a (deterministic) fully-polynomial-approximation scheme in this case.) Recently tree decompositions of planar graphs have been used to give FPTASes for certain partition functions on planar graphs — see [104].

2.1.2 Contribution

Our objective is to determine whether approximating the partition function of the hard-core model is computationally intractable on planar graphs for sufficiently large λ. It turns out that this is so. Our main result (see Theorem 2.1) is that, for a wide range of two-spin parameters, there is no FPRAS, even for planar graphs with degree at most 4. The applicable range of parameters includes the hard-core model with $\lambda \geq 312$. Thus, we show that approximation is difficult for this problem (see Corollary 2.2).

An interesting difference between the general case and the planar case is that, in general, it is difficult to approximate the logarithm of the partition function, a quantity which has physical significance and is called the *mean free energy*. Sly and Sun (see the proofs of Theorems 1 and 2 in [93]) showed that there is a fixed $c > 1$ such that no algorithm can approximate $Z_{\beta,\gamma,\lambda}(G)$ within a factor $c^{|V(G)|}$ unless NP=RP. By contrast, we show (see Theorem 2.3) that, in the planar case, there is a polynomial-time approximation scheme for $\log Z_{\beta,\gamma,\lambda}(G)$ (which implies an approximation within a factor $c^{|V(G)|}$ since $Z_{\beta,\gamma,\lambda}(G)$ is at most $C^{|V(G)|}$ for a quantity C which depends only on β, γ and λ).

At a high level, our hardness result is a reduction from the optimisation problem of computing a maximum independent set in a cubic planar graph G to the problem of estimating the partition function of a much larger graph, which is constructed from G. Each vertex of G is represented by a gadget which is a "wrapped" rectangular lattice C_ν (see Figure 2.1). Similar to previous results of Goldberg and Jerrum [60] and Sly [92], and Sly and Sun [93], we exploit the phase transition of the gadget to enable a reduction from a hard optimisation problem.

The optimisation problem from which we start (computing a maximum independent set in a cubic planar graph) plays a similar role to that of the maximum cut problem in the reduction of Sly and Sun [93]. However, there is a key difference. Since, as we discuss below, it turns out that the logarithm of our partition function is efficiently approximable, it is therefore necessary that the optimisation problem from which we start is also easy to approximate (otherwise, we would get a contradiction). This means that our reduction has to be more carefully tuned — the approximation of the partition function has to allow us to exactly solve the optimisation problem.

A key technical challenge in the proof is to characterise the Gibbs distribution of the two-spin model on the lattice gadget. We show that the spins of the vertices do exhibit long-range correlation. In fact, the gadget is almost always in one of two phases. Each of these phases are equally likely. Also, conditioned on the phase, the spins of certain vertices along the boundary of the gadget are nearly independent, and their distribution can be determined. Thus, although there is long-range correlation between spins, all of the correlation is captured by the phase. Conditioned on the phase, the spins are not very correlated. The analysis of the Gibbs distribution of the gadget uses contour arguments adapted from Dobrushin [45] and Borgs et al. [10]. Randall's slow-mixing result is also based on contour arguments.

We have described the first step, showing that the Gibbs distribution of this gadget can be described as a combination of two phases. We now describe how this phase transition is used to simulate a new two-spin system. The argument is a modification of how phase transitions are used in previous results such as [60, 92]. Given a cubic planar graph G, we form a new graph J' with one copy of C_ν for each vertex of G, adding some carefully chosen new vertices and edges. The partition function of this new graph can be written as a sum over choices of a phase for each gadget, up to a small error. For each choice of phases, we get a term corresponding to a weight of another two-spin system on G. In this way the partition function $Z_{\beta,\gamma,\lambda}(J')$ can be written as a partition function of this new two-spin system on G. We then show that an approximation of that partition function determines the maximum size of an independent set in G, which is NP-hard to compute.

In statistical physics it is sometimes useful to approximate the logarithm of the partition function, even when the partition function itself cannot be approximated (for example, in the situation of Theorem 2.1). Bandyopadhyay and Gamarnik [3] have shown how to estimate the logarithm of the partition function of the hard-core model when λ is small and the graph is regular, with large girth. They show that, in this case, the approximate value does not depend on the graph, given its degree and size! We give (Theorem 2.3) an approximation scheme for the logarithm of the partition function which applies to all planar graphs, for sufficiently large λ. The algorithm is based on the decomposition technique that Baker [2] used to give approximation schemes for optimisation problems on planar graphs. There is a parameter k which is governed by desired approximation quality. The graph G is decomposed into pieces which are

k-outerplanar, and therefore have bounded tree-width. The partition functions of these pieces can be calculated directly using an algorithm of Yin and Zhang [104]. These are combined to give the estimate.

Here is a summary of the main arguments in this chapter.

- We characterise the Gibbs distribution of a gadget $C_{|}size$ using contour arguments (Proposition 2.5).
- We combine these gadgets to simulate another two-spin system partition function (Lemma 2.25).
- A reduction from Max Cubic Planar IS shows that these simulated partition functions are hard to approximate, giving the main result (Theorem 2.1).
- We give a PRAS for the logarithm of the partition function using a decomposition into graphs of bounded treewidth. (Theorem 2.3).

2.2 Preliminaries and statement of results

In our main result, we will assume that the parameters β, γ and λ satisfy the following conditions.

$$\lambda \geq 1, \quad \beta \geq 1 > \gamma \geq 0, \quad \beta\gamma < 1, \quad \beta\lambda^{-1/4} \leq 0.238, \text{ and } \gamma\lambda^{3/8} \leq 0.238. \tag{2.1}$$

Note that these conditions are satisfied by the hard-core model when $\lambda \geq 312$ (by setting $\beta = 1$ and $\gamma = 0$).

The notion of a fully polynomial randomised approximation scheme (FPRAS) is defined in Section 1.2. Following [60, 61], we say that a real number z is *efficiently approximable* if there is an FPRAS for the problem of computing z. For fixed efficiently approximable reals β, γ and λ satisfying (2.1), we consider the problem of (approximately) computing $Z_{\beta,\gamma,\lambda}(G)$, given an input graph G. In order to make our (negative) result as strong as possible, we restrict the input G to have degree at most 4 as well as being planar. Thus, we study the following computational problem.

Name. $\mathsf{DegreeFourPlanarTwoSpin}(\beta, \gamma, \lambda)$.
Instance. A planar graph G with maximum degree at most 4.
Output. The value $Z_{\beta,\gamma,\lambda}(G)$.

Our main result is the following.

Theorem 2.1. *Suppose that β, γ and λ are efficiently approximable reals satisfying* (2.1)*. There is no FPRAS for* $\mathsf{DegreeFourPlanarTwoSpin}(\beta, \gamma, \lambda)$ *unless* NP $\subseteq$ BPP.

The inclusion NP $\subseteq$ BPP would imply NP = RP [76, Theorem 2]. So, for any fixed β, γ and λ satisfying (2.1), there is no FPRAS for $\mathsf{DegreeFourPlanarTwoSpin}(\beta, \gamma, \lambda)$ unless NP = RP.

Of course, our result has an immediate consequence for the problem of approximating the partition function in the hard-core model. Thus, Theorem 2.1 implies Corollary 2.2 for the following computational problem.

Name. DegreeFourPlanarHardCore(λ).
Instance. A planar graph G with maximum degree at most 4.
Output. The value $Z_{1,0,\lambda}(G)$.

Corollary 2.2. *Suppose that $\lambda \geq 312$ is an efficiently approximable real. There is no FPRAS for* DegreeFourPlanarHardCore(λ) *unless* NP $\subseteq$ BPP.

Despite Theorem 2.1, we show that the logarithm of the partition function can be approximated. In particular, we study the following computational problem, where, for concreteness, we use the natural logarithm (to the base e).

Name. PlanarLogTwoSpin(β, γ, λ).
Instance. A planar graph G.
Output. The value $\log(Z_{\beta,\gamma,\lambda}(G))$.

A randomised approximation scheme is said to be a *polynomial randomised approximation scheme* or *PRAS* if, for each ε, its running time is bounded by a polynomial in the length of the input. Our result is that there is a polynomial-time randomised approximation scheme (PRAS) for PlanarLogTwoSpin(β, γ, λ).

Theorem 2.3. *Suppose that β, γ and λ are efficiently approximable reals satisfying $\beta \geq 1 > \gamma \geq 0$ and $\lambda \geq 1$. There is a PRAS for* PlanarLogTwoSpin(β, γ, λ).

The randomness used by the algorithm promised by Theorem 2.3 is only needed to approximate the parameters β, γ and λ. If these are deterministically approximable, then the approximation is deterministic.

We will need some notation to refer to the Gibbs distribution of the two-spin model on a graph G, which is the distribution in which the probability of each configuration is proportional to its weight. We will use $\boldsymbol{\sigma}_G$ to denote a random configuration drawn from this distribution. Thus, for any configuration $\sigma \colon V(G) \to \{0, 1\}$,

$$\Pr(\boldsymbol{\sigma}_G = \sigma) = w_G(\sigma)/Z_{\beta,\gamma,\lambda}(G).$$

(In general, as here, we use boldface for the random variable and normal type for the values that it takes on.) Finally, given a subset S of $V(G)$ and a configuration $\sigma : V(G) \to \{0, 1\}$, let $\sigma(S) : S \to \{0, 1\}$ denote the configuration induced by σ on S.

As mentioned in the introduction, the value of ε in the definition is usually not important. For any graph G, if we denote by $k \cdot G$ the graph composed of k disjoint copies of G, then $Z_{\beta,\gamma,\lambda}(k \cdot G) = Z_{\beta,\gamma,\lambda}(G)^k$, Setting $k = O(\varepsilon^{-1})$, a constant factor approximation to $Z_{\beta,\gamma,\lambda}(k \cdot G)$ will yield (by taking the kth root) an FPRAS for DegreeFourPlanarTwoSpin(β, γ, λ). Clearly, an approximation within a polynomial factor would also suffice. Note that the same argument does not necessarily apply to log-partition functions.

2.3 The Gadget

We will assume throughout this section that β, γ and λ satisfy (2.1), so we do not keep repeating this condition in the statement of our lemmas.

The gadget C_ν has vertex set $V(C_\nu) = \mathbb{Z}/2\nu\mathbb{Z} \times \{0, \dots, \nu\}$. Vertices (x, y) and (x', y') are adjacent in C_ν if

- $y = y'$ and $x = x' \pm 1$ (where of course, the arithmetic is modulo 2ν since x and x' are in $\mathbb{Z}/2\nu\mathbb{Z}$), or
- $x = x'$ and $y = y' \pm 1$.

Let $E(C_\nu)$ denote the set of edges of C_ν. See the leftmost picture in Figure 2.1.

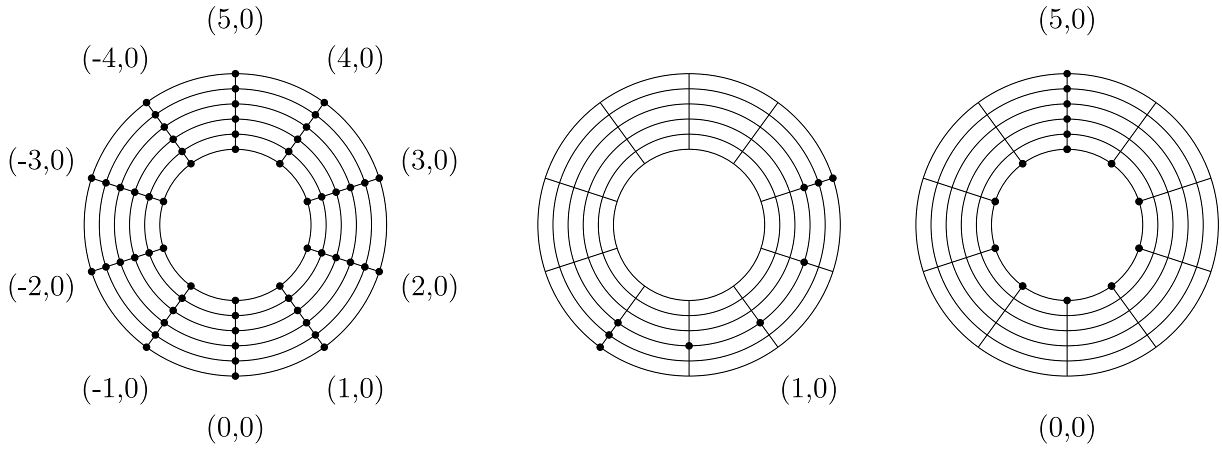

FIGURE 2.1: C_5, and the vertex subsets $B_{1,2}$, and $B_{0,5}$.

2.3.1 Goalposts and keyholes

Given a vertex $(x, 0) \in V(C_\nu)$ and a value $m \in \{0, \dots, \nu\}$ let $B_{x,m}$ be the set containing the vertices on the rectangular (goalpost-shaped) path at distance m around the terminal. In particular, let

$$B_{x,m} = \bigcup_{0 \le j \le m} \{(x - m, j), (x - j, m), (x + j, m), (x + m, j)\}.$$

Again, the arithmetic is done modulo 2ν since $x \in \mathbb{Z}/2\nu\mathbb{Z}$. See the middle picture in Figure 2.1.

When $m = \nu$, the vertices in $\{(x - m, j) \mid 0 \le j \le m\}$ coincide with the vertices in $\{(x + m, j) \mid 0 \le j \le m\}$ so $B_{x,m}$ becomes the "keyhole" which is depicted in the rightmost picture of Figure 2.1 (for $x = 0$).

We shall often be working with configurations on gadgets. For convenience the notation $\boldsymbol{\sigma}_{C_\nu}$ will be contracted to $\boldsymbol{\sigma}_\nu$, and no confusion should result.

2.3.2 Parity-0 ones and parity-1 ones

We say that a vertex $(x, y) \in V(C_\nu)$ has parity 1 if $x + y$ is odd, and that it has parity 0 otherwise. Suppose that S is a subset of $V(C_\nu)$ and that $s \in \{0, 1\}$. We say that $\sigma(S)$ has *parity-s ones* if $\{(x, y) \in S \mid \sigma(x, y) = 1\}$ is exactly the set of parity-s vertices in S.

2.3.3 Idealised probabilities

Define

$$p^{=} = \limsup_{\nu\to\infty} \Pr(\boldsymbol{\sigma}_\nu(0, 0) = 1 \mid \boldsymbol{\sigma}_\nu(B_{0,\nu}) \text{ has parity-0 ones}), \text{and}$$
$$p^{\neq} = \limsup_{\nu\to\infty} \Pr(\boldsymbol{\sigma}_\nu(1, 0) = 1 \mid \boldsymbol{\sigma}_\nu(B_{1,\nu}) \text{ has parity-0 ones}).$$

The notation $p^{=}$ is meant to connote that we are looking at the probability of a 1 at a vertex of parity s, conditioned on certain parity-t ones, where $s = t$; for $p^{\neq}$ we are interested in $s \neq t$. As we shall see later, it will turn out that $p^{=} > p^{\neq}$. This is a non-trivial fact about the spin system: if there were no long-range correlations, we would have $p^{=} = p^{\neq}$. The following straightforward lemma is also useful.

Lemma 2.4. *$p^{\neq} > 0$ and $p^{=} < 1$.*

Proof. Suppose $\nu \geq 2$. Consider vertex $(1, 0)$ of C_ν. Let $S = \{(1, 0), (2, 0), (0, 0), (1, 1)\}$ be the set containing $(1, 0)$ and its immediate neighbours. Let $S' = \{(-1, 0), (0, 1), (1, 2), (2, 1), (3, 0)\}$ be the set containing the neighbours of S. Given any $\sigma : S' \to \{0, 1\}$,

$$\Pr(\boldsymbol{\sigma}_\nu(S) \text{ has parity-1 ones} \mid \boldsymbol{\sigma}_\nu(S') = \sigma) \geq \lambda/16\lambda^4\beta^{10} > 0.$$

Now let $S'' = \{(-1, 0), (0, 1), (1, 0)\}$ be the neighbours of $(0, 0)$. Given any $\sigma : S'' \to \{0, 1\}$,

$$\Pr(\boldsymbol{\sigma}_\nu(0, 0) = 1 \mid \boldsymbol{\sigma}_\nu(S'') = \sigma) \leq \frac{\lambda}{1 + \lambda} < 1.$$

□

The events that $\boldsymbol{\sigma}_\nu(B_{0,\nu})$ has parity-0 ones and that $\boldsymbol{\sigma}_\nu(B_{1,\nu})$ has parity-0 ones have low probability, so it may seem strange to condition on these events, but the purpose of this conditioning is to identify two phases of the idealised gadget. We will refer to certain vertices $(x, 0)$ of C_ν as "terminals", and it will turn out to be the case that the spins of these terminals are nearly independent of each other in the distribution of $\boldsymbol{\sigma}_\nu$.

We will study the distribution that $\boldsymbol{\sigma}_\nu$ induces on the terminals by considering an idealised distribution with two phases. In each of these two gadget phases, the spins of the terminals will be chosen independently. Some terminals will be assigned spin 1 with probability $p^{=}$ and others will be assigned spin 1 with probability $p^{\neq}$. This will be explained further in the next section.

2.3.4 Terminals

Fix positive integers d and k. Let $\nu = 2dk$ and let $C_{k,d}$ denote the gadget C_ν. We will work with $C_{k,d}$ for the rest of the chapter. We will use both notations, C_ν and $C_{k,d}$, depending on whether we want to emphasize the role of ν or the role of k and d. Similarly, the alternative notations, $\boldsymbol{\sigma}_{k,d}$ and $\boldsymbol{\sigma}_\nu$ will be used as convenient.

Some of the vertices around the boundary of $C_{k,d}$ ($2k$ of them) are designated as "terminals". The set of "parity-1 terminals" is

$$T^1_{k,d} = \{(4jd + 1, 0) \mid 0 \le j \le k - 1\}.$$

The set of "parity-0 terminals" is

$$T^0_{k,d} = \{(4jd + 2d, 0) \mid 0 \le j \le k - 1\}.$$

Let $T_{k,d} = T^1_{k,d} \cup T^0_{k,d}$ denote the set of terminals.

For parity $s \in \{0, 1\}$, let $\mu^s_{k,d}$ be the distribution on configurations $\sigma : T_{k,d} \to \{0, 1\}$ in which the spin of each terminal is chosen independently as follows: For each parity-s terminal $(x, 0)$, set $\sigma(x, 0) = 1$ with probability $p^=$ (and set $\sigma(x, 0) = 0$ otherwise). For each terminal $(x, 0)$ with parity $1 \oplus s$, set $\sigma(x, 0) = 1$ with probability $p^{\neq}$ (and set $\sigma(x, 0) = 0$ otherwise).

Informally, the distribution $\mu^s_{k,d}$ will be relevant when an idealised gadget is in a phase which prefers 1-spins at parity-s terminals. In this distribution, the probability that a terminal is given spin 1 is higher if the terminal has parity s than if it has parity $1 \oplus s$.

Let $\mu_{k,d}$ be the distribution on configurations $\sigma : T_{k,d} \to \{0, 1\}$ given by $\mu_{k,d}(\sigma) = (\mu^0_{k,d}(\sigma) + \mu^1_{k,d}(\sigma))/2$. We will show that, provided that d is sufficiently large, the distribution of $\boldsymbol{\sigma}_{k,d}(T_{k,d})$ is close to $\mu_{k,d}$.

Thus, the gadget can be thought of informally as having two phases, phases 0 and 1. We will show that the gadget almost always occupies one of these two phases, and they occur with equal probability. In phase 0, the distribution of $\boldsymbol{\sigma}_{k,d}(T_{k,d})$ is close to $\mu^0_{k,d}$. In phase 1, the distribution of $\boldsymbol{\sigma}_{k,d}(T_{k,d})$ is close to $\mu^1_{k,d}$.

Proposition 2.5. *There is a $c > 1$ such that, if d is a sufficiently large multiple of 16, k is an integer greater than or equal to 1 and τ is a configuration $\tau : T_{k,d} \to \{0, 1\}$, then*

$$|\Pr(\boldsymbol{\sigma}_{k,d}(T_{k,d}) = \tau) - \mu_{k,d}(\tau)| \le c^{-d} k^2.$$

Proposition 2.5 is established at the end of this section. We will use contour arguments adapted from Dobrushin [45] and Borgs et al. [10]. The outline of the argument is as follows. We first define "contours" in Section 2.3.5. We show, in Section 2.3.6, that long contours are unlikely. In Section 2.3.7, we show that, in the absence of long contours, the spins of terminals are nearly independent. With high probability, the gadget has a phase s and there is a boundary around each terminal, whose spins are consistent with s. Conditioned on s, the distribution of the spins of the terminals is close to $\mu^s_{k,d}$.

2.3.5 The Dual Gadget, trails, and contours

The dual gadget C_ν^* has vertex set $V(C_\nu^*) = \{x + \frac{1}{2} \mid x \in \mathbb{Z}/2\nu\mathbb{Z}\} \times \{y + \frac{1}{2} \mid y \in \{-1, \ldots, \nu\}\}$. Vertices (x, y) and (x', y') are adjacent in C_ν^* if

- $y = y'$ and $y \notin \{-\frac{1}{2}, \nu + \frac{1}{2}\}$, and $x = x' \pm 1$ (where of course, the arithmetic is modulo 2ν), or
- $x = x'$ and $y = y' \pm 1$.

$E(C_\nu^*)$ denotes the edge set of C_ν^*. This is illustrated in Figure 2.2.

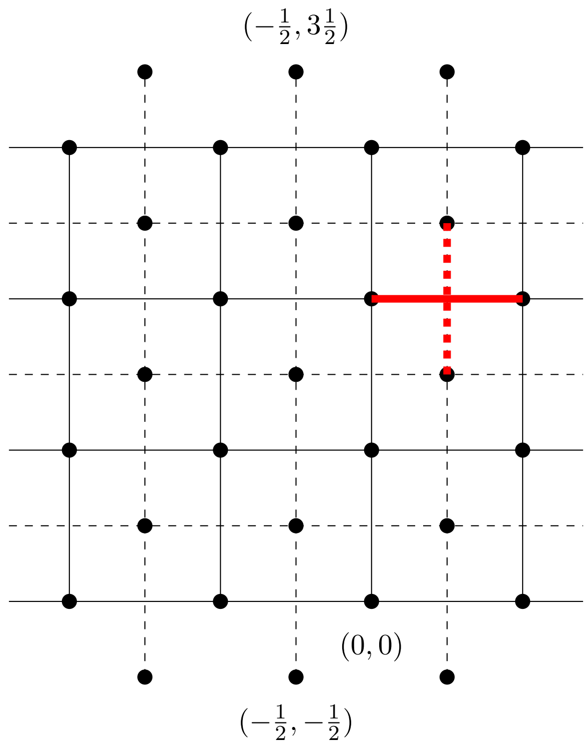

FIGURE 2.2: Part of C_3 and C_3^*; solid lines are edges of C_3, dashed lines are edges of C_3^*. The red thickened lines are a dual pair of edges.

There is a bijection called "duality" between edges of C_ν and edges of C_ν^*. In particular, the dual of edge $e = ((x, y), (x+1, y))$ of C_ν is $e^* = ((x + \frac{1}{2}, y - \frac{1}{2}), (x + \frac{1}{2}, y + \frac{1}{2}))$ and the dual of edge e^* is e. Similarly, the dual of edge $f = ((x, y), (x, y+1))$ of C_ν is $f^* = ((x - \frac{1}{2}, y + \frac{1}{2}), (x + \frac{1}{2}, y + \frac{1}{2}))$ and the dual of f^* is f. We use the * operation to move between an edge and its dual, so every edge e satisfies $(e^*)^* = e$.

A *trail* in C_ν^* is a sequence $g = v_1, \ldots, v_j$ of vertices in $V(C_\nu^*)$ such that each pair (v_i, v_{i+1}) is an edge of C_ν^*, and no edge is used twice. A *contour* is a trail $g = v_1, \ldots, v_j$ in C_ν^* satisfying one of the following:

- $v_1 = v_j$, or
- The y-coordinate of v_1 and the y-coordinate of v_j are both in $\{-\frac{1}{2}, \nu + \frac{1}{2}\}$.

The *length* of g is $j-1$. We say that g is a *cross contour* if the y-coordinate of v_1 is $-\frac{1}{2}$ and the y-coordinate of v_j is $\nu+\frac{1}{2}$ (or vice-versa). A cross contour goes from one boundary of the gadget to the other. We say that every other contour is a *simple* contour.

Given $\sigma : V(C_\nu) \to \{0,1\}$, let σ^* be the set of edges of C_ν^* which are dual to monochromatic edges. In particular, $\sigma^* = \{(u,v)^* \in E(C_\nu^*) \mid \sigma(u) = \sigma(v)\}$.

Definition 2.6. A *contour* of σ is a contour $g = v_1, \ldots, v_j$ satisfying the following two properties.

- The edges of g are monochromatic: That is, for all $1 \le i < j$, $(v_i, v_{i+1}) \in \sigma^*$.
- The contour g always turns at degree-4 vertices: That is, for all $1 < i < j$, if four edges of σ^* meet at vertex v_i, then v_{i-1} and v_{i+1} differ in both the x component and the y component. Similarly, if four edges of σ^* meet at $v_1 = v_j$ then v_2 and v_{j-1} differ in both the x component and the y component.

Note that contours of σ cannot cross each other, though two contours can share a vertex without crossing. Also, two contours can have a common portion (before turning off in two different directions). Finally, every edge of σ^* is contained in at least one contour of σ.

Let $\sigma : V(C_\nu) \to \{0,1\}$ be a configuration, and let g be a contour of σ. We say that a vertex $u \in V(C_\nu)$ is adjacent to g if there is an edge $(u,v) \in E(C_\nu)$ such that $e^* \in g$. The set of vertices adjacent to g can be written as the union of two sets, $L(g)$ and $R(g)$, where $L(g)$ is the set of vertices which are on the left (relative to the direction of travel) when we follow the trail g from v_1 to v_j, and $R(g)$ is the set of vertices which are on the right (relative to the direction of travel). See Figure 2.3.

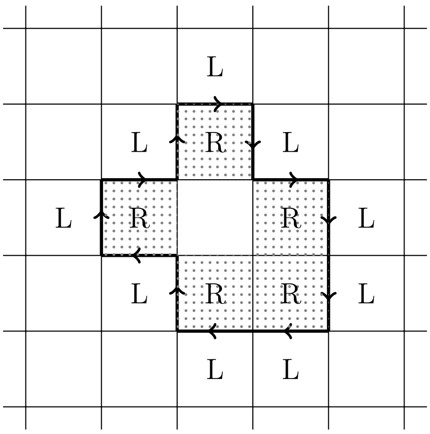

FIGURE 2.3: Left and right vertices of a contour of σ. The shaded squares represent vertices of C_ν with parity-1 ones and the unshaded squares represent vertices of C_ν with parity-0 ones.

A key property of contours is the following.

Lemma 2.7. *Let $\sigma : V(C_\nu) \to \{0,1\}$ be a configuration, and let g be a contour of σ. Then for some $s \in \{0,1\}$, $\sigma(L(g))$ has parity-s ones and $\sigma(R(g))$ has parity-$(1 \oplus s)$ ones.*

Proof. Pick $s \in \{0,1\}$ such that the vertex on the left as we go from v_1 to v_2 has parity-s ones. By induction on i, we will show that for each i the vertex on the left as we go from v_i to v_{i+1} has parity-s ones. Suppose without loss of generality that the edge (v_{i-1}, v_i) increases the x-component (the other three cases are similar). So $v_{i-1} = (x - \frac{1}{2}, y + \frac{1}{2})$ and $v_i = (x + \frac{1}{2}, y + \frac{1}{2})$. Since g is a contour of σ, $\sigma(x, y) = \sigma(x, y + 1) = s \oplus x \oplus y$. There are three cases.

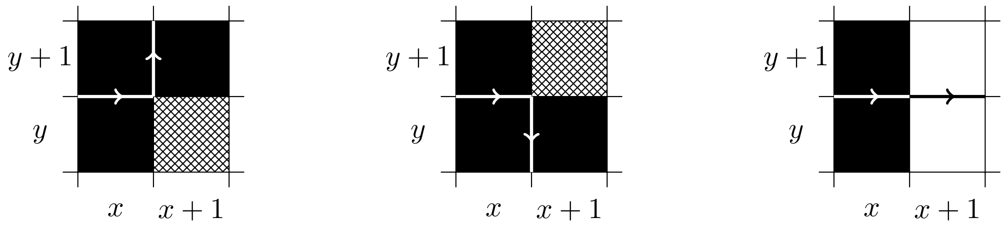

FIGURE 2.4: Cases 1, 2, and 3. Black squares have the same spin as $\sigma(x, y)$; white squares have the opposite spin, and hatched squares can be either.

1. $v_{i+1} = (x + \frac{1}{2}, y + \frac{3}{2})$. In this case the vertex $(x, y + 1)$ is still on the left as we go from v_i to v_{i+1}.

2. $v_{i+1} = (x + \frac{1}{2}, y - \frac{1}{2})$. In this case the vertex $(x + 1, y)$ is on the left as we go from v_i to v_{i+1}, but since g is a contour of σ we have $\sigma(x+1, y) = \sigma(x, y) = \sigma(x, y+1)$. So $(x + 1, y)$ has parity-s ones.

3. $v_{i+1} = (x + \frac{3}{2}, y + \frac{1}{2})$. In this case $(x + 1, y + 1)$ is on the left, and $(x + 1, y)$ is on the right. Since g is a contour of σ, $\sigma(x + 1, y) = \sigma(x + 1, y + 1)$, and we know $\sigma(x, y) = \sigma(x, y + 1)$. Since the contour did not turn, the vertex $(x + \frac{1}{2}, y + \frac{1}{2})$ cannot have degree 4 in σ^*, so $\sigma(x+1, y+1) = s \oplus x \oplus y \oplus 1$, so $(x+1, y+1)$ has parity-s ones. □

The following lemma allows $w_G(\sigma)$ to be expressed more easily in terms of the contours of σ. Suppose $\nu > 2$. A *side vertex* of C_ν is a vertex $(x, y) \in V(C_\nu)$ with $y = 0$ or $y = \nu$. A *side edge* is an edge in $E(C_\nu)$ between two side vertices.

Lemma 2.8. *Fix $\nu > 2$ and a configuration $\sigma : V(C_\nu) \to \{0,1\}$. Let $b'(\sigma)$ be the number of side edges (u, v) of C_ν with $\sigma(u) = \sigma(v) = 0$ and let $c'(\sigma)$ be the number of side edges (u, v) of C_ν with $\sigma(u) = \sigma(v) = 1$. Then*

$$\ell(\sigma) = \tfrac{1}{4}(c(\sigma) - b(\sigma)) + \tfrac{1}{8}(c'(\sigma) - b'(\sigma)) + \nu(\nu + 1).$$

Proof. Let $\ell'(\sigma)$ be the number of side vertices u with $\sigma(u) = 1$. Let E' be the set of all side edges in C_ν. By double-counting pairs $(u, (u, v))$ with $\sigma(u) = 1$ and $(u, v) \in E(C_\nu)$,

$$(|E(C_\nu)| - b(\sigma) - c(\sigma)) + 2c(\sigma) = 4\ell(\sigma) - \ell'(\sigma).$$

By double-counting pairs $(u,(u,v))$ with $\sigma(u)=1$ and $(u,v)\in E'$, we have

$$(|E'|-b'(\sigma)-c'(\sigma))+2c'(\sigma)=2\ell'(\sigma).$$

Rearranging gives $\ell(\sigma)=\frac{1}{4}(c(\sigma)-b(\sigma))+\frac{1}{8}(c'(\sigma)-b'(\sigma))+\frac{1}{4}|E(C_\nu)|+\frac{1}{8}|E'|$. Consider the configuration with alternating 0s and 1s given by $\sigma(x,y)=x\oplus y$. For this configuration σ, we have $b(\sigma)=c(\sigma)=b'(\sigma)=c'(\sigma)=0$ and $\ell(\sigma)=\nu(\nu+1)$, so the constant term $\frac{1}{4}|E(C_\nu)|+\frac{1}{8}|E'|$ is $\nu(\nu+1)$. ☐

2.3.6 Long contours are unlikely

Lemma 2.9. *There is a $c>1$ such that, for all sufficiently large h, all $\nu>2$, and all $U\subseteq V(C_\nu^*)$,*

$$\Pr(\boldsymbol{\sigma}_\nu \textit{ has a simple contour of length at least } h \textit{ starting in } U)\le |U|\,c^{-h}.$$

Proof. Suppose that g is a simple length-r contour of a configuration $\sigma\colon V(C_\nu)\to\{0,1\}$. Consider the connected components of the graph $\big(V(C_\nu),E(C_\nu)\setminus\{e^*\mid e\in g\}\big)$. We say that a component is "left" if it contains at least one vertex in $L(g)$ (but no vertices in $R(g)$). We say that it is "right" if it contains at least one vertex in $R(g)$ (but no vertices in $L(g)$). Every component is either left or right. Let S be the set of vertices in left components. Let $\overline{S}=V(C_\nu)\setminus S$. Let $S'=\{(x,y)\in S\mid (x-1,y)\in\overline{S}\}$, where, as usual, the arithmetic on x is done modulo 2ν.

Suppose that $\sigma(R(g))$ has parity-s ones. By Lemma 2.7, this is true for some $s\in\{0,1\}$. Define a configuration $\sigma^g: V(C_\nu)\to\{0,1\}$ as follows: $\sigma^g(\overline{S})=\sigma(\overline{S})$, $\sigma^g(S')$ has parity-s ones, and, for every $(x,y)\in S\setminus S'$, $\sigma^g(x,y)=\sigma(x-1,y)$.

Note that $\sigma\mapsto\sigma^g$ is a map from the set of configurations σ with g as a contour to the set of all configurations; further, it does not lose information, and hence is injective. Note also that $(\sigma^g)^*$ is the same as σ^*, but with g removed and with the contours in S shifted by one. By Lemma 2.8,

$$w_{C_\nu}(\sigma)=(\beta\lambda^{-1/4})^{b(g)}(\gamma\lambda^{1/4})^{c(g)}\lambda^{(c'(g)-b'(g))/8}w_{C_\nu}(\sigma^g)$$

where $b(g),c(g),b'(g),c'(g)$ are the contributions to $b(\sigma),c(\sigma),b'(\sigma),c'(\sigma)$ coming from edges whose duals are in g. As the map $\sigma\mapsto\sigma^g$ is injective,

$$\begin{aligned}\Pr(g\subseteq\boldsymbol{\sigma}_\nu^*)&\le(\beta\lambda^{-1/4})^{b(g)}(\gamma\lambda^{1/4})^{c(g)}\lambda^{(c'(g)-b'(g))/8}\\&\le(\beta\lambda^{-1/4})^{b(g)}(\gamma\lambda^{3/8})^{c(g)},\end{aligned}$$

where we have used the facts that $\lambda\ge1$ and $c'(g)\le c(g)$. There are at most $|U|\,3^r$ relevant contours of length r in total ($|U|$ choices of starting point, and at most three different directions at each step), so

$$\begin{aligned}&\Pr(\boldsymbol{\sigma}_\nu \text{ has a simple contour of length at least } h \text{ starting in } U)\\&\quad\le |U|\sum_{r\ge h} 3^r \max(\beta\lambda^{-1/4},\gamma\lambda^{3/8})^r\\&\quad\le |U|\,\frac{(3\times 0.238)^h}{1-3\times 0.238}.\end{aligned}$$

There is a $c>1$ such that $\frac{(3\times 0.238)^h}{1-3\times 0.238} < c^{-h}$ for all sufficiently large h. $\square$

Lemma 2.10. *Let $i\in\{0,1\}$. For every $\nu>2$, and every simple contour g of length r,*

$$\Pr(g \textit{ is a contour of } \boldsymbol{\sigma}_\nu \;\mid \boldsymbol{\sigma}_\nu(B_{i,\nu}) \textit{ has parity-}0 \textit{ ones}) \le \max(\beta\lambda^{-1/4},\gamma\lambda^{3/8})^r. \quad (2.2)$$

Furthermore, there is a $c>1$ such that, for all sufficiently large h, all $\nu>2$, and all $U\subseteq V(C_\nu)$, the conditional probability that $\boldsymbol{\sigma}_\nu$ has a simple contour of length at least h which contains an edge whose dual connects two vertices in U, conditioned on the fact that $\boldsymbol{\sigma}_\nu(B_{i,\nu})$ has parity-0 ones, is at most $|U|\,c^{-h}$.

Proof. The proof of (2.2) is similar to the first half of the proof of Lemma 2.9, except that we have to take care to choose S to be on the correct side of the contour g. Previously, it did not matter whether we formed S from the left or right components, and we arbitrarily chose the former. Now we choose S (either taking all the left or all the right components) in such a way that $S\cap B_{i,\nu}=\emptyset$. This is possible because all the vertices in $B_{i,\nu}$ are in a single connected component (the contour g does not cross any edges whose endpoints lie in $B_{i,\nu}$). Now define σ^g as in the proof of Lemma 2.9 and continue as before. This establishes (2.2).

For all $1\le s\le r$, and all $u\in U$, there are at most $3^r\times 4$ contours $v_1\dots v_r$ for which u is on the left as we go from v_{s-1} to v_s: a choice of initial direction and direction at each step determines the contour. Summing over s and u, this implies that there are at most $4|U|r3^r$ length-r contours with an edge whose dual connects vertices of U. By (2.2),

$$\begin{aligned}&\Pr\left(\begin{array}{c}\boldsymbol{\sigma}_\nu \text{ has a simple contour of length at least } h \text{ which}\\ \text{contains an edge whose dual connects two vertices in } U\end{array}\;\middle|\; \boldsymbol{\sigma}_\nu(B_{i,\nu}) \text{ has parity-0 ones}\right)\\&\quad\le 4|U|\sum_{r\ge h} r3^r\max(\beta\lambda^{-1/4},\gamma\lambda^{3/8})^r\\&\quad= 4|U|(3\times 0.238)^h\sum_{t\ge 0}(t+h)(3\times 0.238)^t\\&\quad= 4|U|(3\times 0.238)^h\,\frac{3\times 0.238 + h(1-3\times 0.238)}{(1-3\times 0.238)^2}\end{aligned}$$

There is a $c>1$ such that $4(3\times 0.238)^h\,\frac{3\times 0.238+h(1-3\times 0.238)}{(1-3\times 0.238)^2} < c^{-h}$ for all sufficiently large h. $\square$

By the *upper boundary of* C_ν we mean the set of all vertices of the form (x, ν) for some x.

Lemma 2.11. *Let* $i \in \{0, 1\}$. *There is a* $c > 1$ *such that, for all sufficiently large* h *and all* $\nu > h$*, the probability that* $\boldsymbol{\sigma}_\nu$ *has a simple contour that separates the set* $\{-h+i, \ldots, h+i\} \times \{0, \ldots, h\}$ *from the upper boundary of* C_ν*, conditioned on the fact that* $\boldsymbol{\sigma}_\nu(B_{i,\nu})$ *has parity-*0 *ones, is at most* c^{-h}.

Proof. Note that the separating contour cannot wrap around, owing to the boundary condition that $\boldsymbol{\sigma}_\nu(B_{i,\nu})$ has parity-0 ones. If the separating contour has length $r+2$ then its right-endpoint is in the set $\{(h+i+x-\frac{1}{2}, -\frac{1}{2}) \mid 1 \le x \le r\}$. There is a unique choice for the edge incident to each endpoint. Thus, there are at most $r3^r$ possible contours. By Lemma 2.10, the probability that $\boldsymbol{\sigma}_\nu$ has such a simple contour, conditioned on the fact that $\boldsymbol{\sigma}_\nu(B_{i,\nu})$ has parity-0 ones, is at most

$$\sum_{r=h-2}^{\infty} r3^r \max(\beta\lambda^{-1/4}, \gamma\lambda^{3/8})^{r+2}.$$

Thus, the probability is at most

$$\max(\beta\lambda^{-1/4}, \gamma\lambda^{3/8})^2 \sum_{r=h-2}^{\infty} r3^r \max(\beta\lambda^{-1/4}, \gamma\lambda^{3/8})^r,$$

which, as in the proof of Lemma 2.10, is exponentially small in h. $\square$

Lemma 2.12. *There is a* $c > 1$ *such that, for all sufficiently large* ν,

$$\Pr(\boldsymbol{\sigma}_\nu \textit{ has a cross contour }) \le c^{-\nu}.$$

Proof. Let g be a cross contour. There must be at least one other cross contour g'. For otherwise there would be a path p in C_ν from $L(g)$ to $R(g)$ such that $\boldsymbol{\sigma}_\nu(V(p))$ has parity-0 ones or parity-1 ones, which would violate parity. Orient g and g' in opposite senses (one away from $y = -\frac{1}{2}$ and one towards). Consider the connected components of the graph $\big(V(C_\nu), E(C_\nu) \setminus \{e^* \mid e \in g \cup g'\}\big)$, and let S be the union of all connected components that are left of either g or g'. Now proceed as in the proof of Lemma 2.9, using the fact that a cross contour has length at least ν and the set of possible starting points has size 2ν. $\square$

Lemma 2.13. *Let* $i \in \{0, 1\}$. *Fix* $\nu \ge 1$. *Conditioned on* $\boldsymbol{\sigma}_\nu(B_{i,\nu})$ *having parity-*s *ones (for any* $s \in \{0, 1\}$*),* $\boldsymbol{\sigma}_\nu$ *has no cross contour.*

Proof. A cross contour would have to cross a side edge in $B_{i,\nu}$, which is impossible. $\square$

Lemma 2.14. $p^{=} > p^{\neq}$.

Proof. Fix $\nu > 2$. Suppose that $\boldsymbol{\sigma}_\nu(B_{0,\nu})$ has parity-0 ones. If $\boldsymbol{\sigma}_\nu(0,0) = 0$ then there is a simple contour of $\boldsymbol{\sigma}_\nu$ that separates $(0,\nu)$ from $(0,0)$. (Note that, by Lemma 2.13, cross contours cannot separate these two vertices.) If the separating contour has length $r+2$ then its right-endpoint is in the range $(\frac{1}{2}, -\frac{1}{2}), \ldots, (r-\frac{1}{2}, -\frac{1}{2})$. There is a unique choice for the edge incident to each endpoint. Thus, there are at most $r3^r$ possible contours. By Lemma 2.10, the probability that $\boldsymbol{\sigma}_\nu$ has such a simple contour, conditioned on the fact that $\boldsymbol{\sigma}_\nu(B_{0,\nu})$ has parity-0 ones, is at most

$$\sum_{r=1}^{\infty} r3^r \max(\beta\lambda^{-1/4}, \gamma\lambda^{3/8})^{r+2}.$$

Thus,

$$\begin{aligned}
\Pr(\boldsymbol{\sigma}_\nu(0,0) = 0 \mid \boldsymbol{\sigma}_\nu(B_{0,\nu}) \text{ has parity-0 ones}) &\le \max(\beta\lambda^{-1/4}, \gamma\lambda^{3/8})^2 \sum_{r=1}^{\infty} r3^r \max(\beta\lambda^{-1/4}, \gamma\lambda^{3/8})^r \\
&\le (0.238)^2 \frac{3 \times 0.238}{(1 - 3 \times 0.238)^2} \\
&< 1/2.
\end{aligned}$$

Thus, $p^{=} > \frac{1}{2}$.

Similarly, suppose that $\boldsymbol{\sigma}_\nu(B_{1,\nu})$ has parity-0 ones. If there is no simple contour of $\boldsymbol{\sigma}_\nu$ that separates $(1,\nu)$ from $(1,0)$, then $\boldsymbol{\sigma}_\nu(1,0) = 0$. We already saw that the probability that no such contour exists is greater than $\frac{1}{2}$. Thus,

$$\Pr(\boldsymbol{\sigma}_\nu(1,0) = 0 \mid \boldsymbol{\sigma}_\nu(B_{1,\nu}) \text{ has parity-0 ones}) > 1/2.$$

So $p^{\neq} < \frac{1}{2}$. Putting the two inequalities together, we have $p^{\neq} < \frac{1}{2} < p^{=}$. □

2.3.7 In the absence of long contours, the spins of the terminals are nearly independent

The $*$-*distance* between two points (x,y) and (x',y') in $V(C_\nu)$ is $\max(|x-x'|, |y-y'|)$ where $|x-x'|$ is the minimum non-negative integer such that $x = x' + |x-x'|$ modulo 2ν or $x' = x + |x-x'|$ modulo 2ν. The $*$-*distance* between two points (x,y) and (x',y') in $V(C_\nu^*)$ is defined similarly. Let $U_{x,h}$ be the set of vertices of C_ν whose $*$-distance from $(x,0)$ is at most h. Vertices in $V(C_\nu)$ (or $V(C_\nu^*)$) are $*$-*adjacent* if the $*$-distance between them is 1. A $*$-path on $V(C_\nu)$ is a sequence $v_1, \ldots, v_h$ of vertices in $V(C_\nu)$ such that, for each $j \in \{1, \ldots, h-1\}$, the vertices v_j and v_{j+1} are $*$-adjacent. A $*$-path on $V(C_\nu^*)$ is defined similarly.

Definition 2.15. Suppose ν and h are positive integers. An h-*boundary* of a vertex $(x,0)$ is a set of vertices $B \subseteq V(C_\nu)$ such that the following are true.

(i) $(x,0)$ is not connected to (x,ν) in the graph $C_\nu \setminus B$, and

(ii) B does not intersect $U_{x,h/4}$, and

(iii) B is a subset of $U_{x,h/2}$.

(iv) The subgraph of C_ν induced by B is connected.

See Figure 2.5.

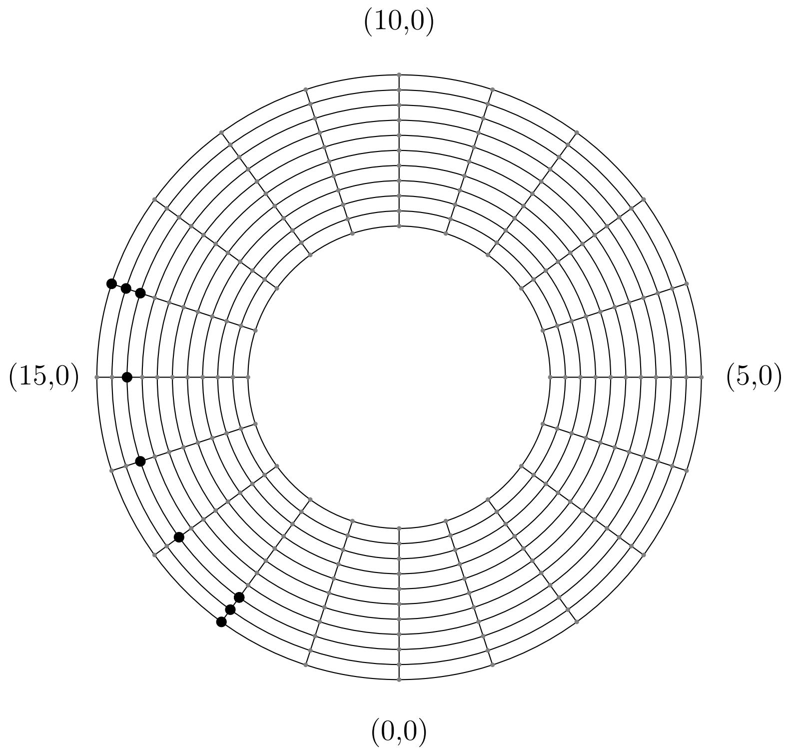

FIGURE 2.5: Example h-boundary of vertex $(16,0)$ with $\nu = 20$ and $h = 5$.

Here is the relevant fact about h-boundaries. If B is an h-boundary of a vertex $(x,0)$ and $\sigma(B)$ has parity-s ones then there is no contour of σ which contains an edge whose dual connects two vertices in B.

If B is an h-boundary of vertex $(x,0)$ and B' is an h'-boundary of $(x,0)$, then we say that B is inside of B' if every path in C_ν from B' to $(x,0)$ passes through B. Suppose that B and B' are h-boundaries of $(x,0)$ and, for some $s \in \{0,1\}$, $\sigma(B)$ has parity-s ones and $\sigma(B')$ has parity-$(1 \oplus s)$ ones. Then $B \cap B' = \emptyset$, so exactly one of the following occurs.

- B is inside of B', or
- B' is inside of B.

Definition 2.16. Suppose that $s \in \{0,1\}$ and that k and d are positive integers. Let $\nu = 2kd$. Suppose that σ is a configuration $\sigma : V(C_\nu) \to \{0,1\}$. We say that σ has phase s if the following holds for every terminal $(x,0)$.

- $(x, 0)$ has a d-boundary B for which $\sigma(B)$ has parity-s ones.
- For every d-boundary B' of $(x, 0)$ for which $\sigma(B')$ has parity-$(1 \oplus s)$ ones, B' is inside of B.

Note that a configuration σ can have exactly one phase (phase 0 or phase 1) or it can have no phase. Suppose that $\nu \geq 1$ and that the configuration $\sigma : V(C_\nu) \to \{0, 1\}$ has phase s. Say that a d-boundary B of a vertex $(x, 0)$ is *consistent* if $\sigma(B)$ has parity-s ones. From the consistent d-boundaries, we want to select a canonical one, that is in some precise sense "outermost" and also "minimal". For each terminal $(x, 0)$, let $\widehat{\mathcal{B}}_x(\sigma)$ be the union of all d-boundaries of $(x, 0)$ which are consistent. Observe that $B = \widehat{\mathcal{B}}_x(\sigma)$ satisfies the first three bullet points in Definition 2.15, but not the final one, as the subgraph $C_\nu[B]$ of C_ν induced by B may not be connected. Suppose that $C_\nu[B]$ has j connected components. Partition $\widehat{\mathcal{B}}_x(\sigma) = \widehat{\mathcal{B}}^1_x(\sigma) \cup \cdots \cup \widehat{\mathcal{B}}^j(\sigma)$ so that $C_\nu[\widehat{\mathcal{B}}^1_x(\sigma)], \ldots, C_\nu[\widehat{\mathcal{B}}^j_x(\sigma)]$ is an enumeration of these j connected components. Each set $\widehat{\mathcal{B}}^i_x(\sigma)$ is itself a d-boundary. To see this, consider any vertex $v \in \widehat{\mathcal{B}}^i_x(\sigma)$. From the construction of $\widehat{\mathcal{B}}_x(\sigma)$, this vertex is contained in some d-boundary, which in turn is contained in $\widehat{\mathcal{B}}^i_x(\sigma)$. So $\widehat{\mathcal{B}}^i_x(\sigma)$ satisfies the first three bullet points in Definition 2.15, in addition to inducing a connected graph.

From the first bullet point of Definition 2.15 it follows that $\widehat{\mathcal{B}}^1_x(\sigma), \ldots, \widehat{\mathcal{B}}^j_x(\sigma)$ are nested; suppose the numbering indicates the level of nesting, with $\widehat{\mathcal{B}}^1_x(\sigma)$ being the outermost. We now want to identify a minimal d-boundary within $\widehat{\mathcal{B}}^1_x(\sigma)$. Let

$$\operatorname{Ext} \widehat{\mathcal{B}}^1_x(\sigma) = \{v \in V(C_\nu) \mid \text{there is a } *\text{-path from } v \text{ to } (x, \nu) \text{ in } V(C_\nu) \setminus \widehat{\mathcal{B}}^1_x(\sigma)\}.$$

denote the set of vertices lying in the "exterior" of $\widehat{\mathcal{B}}^1_x(\sigma)$. Finally define

$$\mathcal{B}_x(\sigma) = \{v \in \widehat{\mathcal{B}}^1_x(\sigma) \mid v \text{ is } *\text{-adjacent to some vertex in } \operatorname{Ext} \widehat{\mathcal{B}}^1_x(\sigma)\}.$$

Note that $\mathcal{B}_x(\sigma)$ is a d-boundary of $(x, 0)$ which is consistent. (Informally, $\mathcal{B}_x(\sigma)$ is the outermost such d-boundary.) To see this, observe that any path in the graph C_ν from $(x, 0)$ to (x, ν) has a last vertex in the set $\widehat{\mathcal{B}}^1_x(\sigma)$, and this vertex must be in $\mathcal{B}_x(\sigma)$; this deals with the first bullet point in Definition 2.15. Consider the trail in the dual graph C^*_ν separating $\mathcal{B}_x(\sigma)$ and $\operatorname{Ext} \widehat{\mathcal{B}}^1_x(\sigma)$; the trail in the primal graph that shadows it at $*$-distance $\frac{1}{2}$ inside takes in all the vertices of $\mathcal{B}_x(\sigma)$ and establishes connectivity of $C_\nu[\mathcal{B}_x(\sigma)]$. This deals with the final bullet mark, and the remaining two are immediate.

The d-boundary $\mathcal{B}_x(\sigma)$ is our desired canonical d-boundary, and it has the following important property. If σ' is a configuration that agrees with σ on $\mathcal{B}_x(\sigma) \cup \operatorname{Ext} \widehat{\mathcal{B}}^1_x(\sigma)$, then $\mathcal{B}_x(\sigma') = \mathcal{B}_x(\sigma)$. The reason is as follows. The set $B = \mathcal{B}_x(\sigma)$ is a d-boundary of $(x, 0)$ which is consistent with respect to σ', i.e., $\sigma'(B)$ has parity-s ones. It therefore gets incorporated into $\widehat{\mathcal{B}}_x(\sigma') \supseteq \mathcal{B}_x(\sigma)$ and hence into $\widehat{\mathcal{B}}^1_x(\sigma') \supseteq \mathcal{B}_x(\sigma)$. So $\operatorname{Ext} \mathcal{B}^1_x(\sigma') = \operatorname{Ext} \mathcal{B}^1_x(\sigma)$ and $\mathcal{B}_x(\sigma') = \mathcal{B}_x(\sigma)$. This fact becomes significant when we come to consider events supported on spins in the interior $U = V(C_\nu) \setminus (B \cup \operatorname{Ext} B)$ of some d-boundary B.

Specifically, conditioning on the event $\mathcal{B}_x(\boldsymbol{\sigma}_\nu) = B$ (and on the phase s of $\boldsymbol{\sigma}_\nu$) is equivalent to selecting $\boldsymbol{\sigma}_\nu(U)$ according to the Gibbs distribution, with the boundary condition "$\boldsymbol{\sigma}_\nu(B)$ has parity-s ones". We refer to this property as *canonicity* of $\mathcal{B}_x(\sigma)$.

Before proceeding we need some definitions and observations concerning connected subgraphs K of the graph $(V(C_\nu^*), \sigma^*)$. The *$*$-diameter of* K is the maximum, over pairs of vertices in K, of the $*$-distance between those vertices. We say that K *reaches the lower boundary of* C_ν if it contains a vertex of the form $(x, -\frac{1}{2})$ for some x. We say that it reaches the upper boundary of C_ν if it contains a vertex of the form $(x, \nu + \frac{1}{2})$. We say that it is a *cross subgraph* if it reaches both boundaries. We say that K *wraps around* if it contains the image of some path from $(0, y)$ to $(2\nu, y)$ in $\mathbb{Z} \times \{0, \dots, \nu\}$, under the quotient map to $V(C_\nu)$. We say that K is *local* if it is not a cross subgraph and does not wrap around. We say that K *intersects* $U_{x,h}$ if some edge e^* in K is dual to an edge $e \in E(C_\nu)$ with both endpoints in $U_{x,h}$. A contour is a connected subgraph of $(V(C_\nu^*), \sigma^*)$ so all of the above definitions apply to contours. We refer to the connected components of $(V(C_\nu^*), \sigma^*)$ as σ^*-components.

Lemma 2.17. *If σ has only local contours then σ^* has only local components.*

Proof. Suppose to the contrary that σ^* has non-local component K. Consider first the case of a cross component.

So suppose K reaches both the upper and lower boundaries of C_ν, and that K has $2j > 0$ degree-1 vertices. (All vertices other than the degree-1 vertices have even degree, and the number of odd-degree vertices in a graph is even.) By adapting the standard algorithm for finding an Eulerian trail in a (connected) Eulerian graph, we may decompose K into j contours beginning and ending at degree-1 vertices. The method is as follows. Starting at a degree-1 vertex, trace out a trail in K subject only to the rule that we must turn through a right-angle at any degree-4 vertex. This trail can only end at another degree-1 vertex. The trail so formed is a contour; remove the trail from K and repeat $j - 1$ more times to obtain j contours in total. If any edges of K remain, start at any remaining vertex and trace out a closed trail that returns to the start vertex. Again, the rule is always to turn through a right-angle at any degree 4 vertex. Repeat until there are no edges remaining in K. We are left with j non-closed contours and an unspecified number of closed ones. Whenever a non-closed contour meets a closed one, we may splice the latter into the former, reducing the number of closed contours by one. Repeating as necessary, we obtain the sought-for decomposition of K into j contours beginning and ending at degree-1 vertices.

If one of these j contours joins the upper and lower boundaries of C_ν we are done, as we have already found a cross contour and obtained a contradiction. Otherwise, there must be at least one vertex at which a lower-to-lower contour touches a upper-to-upper contour. Simply reroute the trails at this vertex to obtain two cross contours.

Now consider the case where K wraps around. We may assume that K does not reach one of the boundaries of C_ν, say the upper one. Trace a closed trail along the upper boundary of K: this trail is a contour that wraps around, providing a contradiction. $\square$

Since non-local contours are unlikely, Lemma 2.17 allows us to concentrate on local σ^*-components. A σ^*-component K that is local has a well defined inside and outside, and a boundary that is a valid contour. (More precisely, there is a canonical contour that has exactly the same edges as the boundary of K.) If K reaches neither the upper nor lower boundary of C_ν, then we may trace clockwise around K, always taking the leftmost option, until we return to our starting point. This procedure yields a simple contour; we refer to vertices of $V(C_\nu)$ that lie within this contour as forming the *interior of* K, denoted $\operatorname{Int} K$.

If K reaches the lower boundary but not the top (or vice versa), then a slightly modified construction can be used. First lift K to a grid: for sufficiently large N, there is a connected subset $\widehat{K}$ of $\{1,\dots,N\}\times\{0,\dots,\nu\}$ which maps bijectively to K under the quotient map to C_ν. Note that lifting can only increase the diameter of $\widehat{K}$ relative to K. We now have a natural ordering of the degree-1 vertices of K, namely by increasing x-coordinate. Start at the least degree-1 vertex in this ordering and and trace the boundary of $\widehat{K}$ in a clockwise-leftmost fashion until the greatest degree-1 vertex is reached. This procedure yields a simple contour which partitions the vertices of $V(C_\nu)$ into an inside and an outside (containing the point $(0,\nu)$); again we refer to the former as the *interior of* K.

Lemma 2.18. *Consider $\sigma\colon V(C_\nu)\to\{0,1\}$. Let $h\le\nu$ be an integer multiple of 8. Suppose*

- *σ contains only local contours.*
- *σ contains no contour of length at least $h/8$ that intersects $U_{x,h/2}$.*
- *σ contains no simple contour separating $U_{x,h/4}$ from the upper boundary of C_ν.*

Then every σ^*-component that intersects $U_{x,h/2}$ has $*$-diameter at most $h/8$.

Proof. Suppose K is a σ^*-component intersecting $U_{x,h/2}$. By Lemma 2.17, K is local. The interior of K contains some vertex in $U_{x,h/2}$. By assumption, the contour defined by the boundary of K does not separate $U_{x,h/4}$ from the upper boundary of C_ν, so it does not separate $U_{x,h/2}$ from the upper boundary of C_ν. The only remaining possibility is that this contour intersects $U_{x,h/2}$, and hence has length at most $h/8$. It follows that K has $*$-diameter at most $h/8$. □

Lemma 2.19. *Suppose that h is a sufficiently large multiple of 8, and $\nu\ge h$. Consider a configuration $\sigma\colon V(C_\nu)\to\{0,1\}$ and a terminal $(x,0)$. Suppose that the following are true.*

- *σ has no cross contours.*
- *σ has no simple contour of length at least $h/8$ that intersects $U_{x,h/2}$.*
- *σ has no simple contour separating $U_{x,h/4}$ from the upper boundary of C_ν.*

Then, for some $s \in \{0,1\}$,

1. *$(x,0)$ has an h-boundary B for which $\sigma(B)$ has parity-s ones.*
2. *There is no h-boundary B' of $(x,0)$ for which $\sigma(B')$ has parity-$(1 \oplus s)$ ones.*
3. *For any terminal $(x',0)$ that has an h-boundary B' for which $\sigma(B')$ has parity-s' ones, if σ has no contour separating $U_{x,h/4}$ from $U_{x',h/4}$ then $s' = s$.*
4. *If $\sigma(B_{0,\nu})$ has parity-s' ones then $s' = s$.*

Proof. There are no contours that wrap around, since any such contour would either intersect $U_{x,h/2}$, or would separate $U_{x,h/4}$ from the upper boundary of C_ν. Thus, by Lemma 2.17, all σ^*-components are local. Let S be the set of all vertices in $V(C_\nu)$ that are not in the interior of some σ^*-component. That is

$$S = V(C_\nu) \setminus \bigcup \big\{ \operatorname{Int} K \mid K \text{ is a } \sigma^*\text{-component} \big\}.$$

Note that $\sigma(S)$ has parity-s ones, for some $s \in \{0,1\}$. Define $\overline{S} = V(C_\nu) \setminus S$. Note that $\sigma(\overline{S}\,)$ in general has mixed parity; the salient feature is that $\sigma(S)$ has consistent parity.

We work first towards conclusion (1) of the lemma. By Lemma 2.18, every σ^*-component that intersects $U_{x,h/2}$ has $*$-diameter at most $h/8$. Now (informally) we will construct the required h-boundary by tracing round the inside of $B_{x,h/2}$, making a detour towards $(x,0)$ around any σ^*-components that stand in the way. (Recall that $B_{x,h/2} = U_{x,h/2} \setminus U_{x,h/2-1}$ is the “goalpost” at distance $h/2$ from $(x,0)$.) This strategy ensures we remain in the set S and, since all the σ^*-components are small, our detours will not be too great.

More formally, let W be the union of the set $V(C_\nu) \setminus U_{x,h/2}$ together with any sets of the form $\operatorname{Int} K$ that intersect $B_{x,h/2}$. That is,

$$W = (V(C_\nu) \setminus U_{x,h/2}) \cup \bigcup \{\operatorname{Int} K \mid K \text{ is a } \sigma^*\text{-component and } \operatorname{Int}(K) \cap B_{x,h/2} \neq \emptyset\}.$$

The set

$$\partial W = \{v \in V(C_\nu) \mid v \notin W \text{ and } v \text{ is } *\text{-adjacent to some vertex in } W\}$$

is almost the h-boundary B that we seek. Observe that any set of the form $\operatorname{Int} K$ is contained in a maximal set of the form $\operatorname{Int} K'$, and the $*$-neighbours of $\operatorname{Int} K'$ are all in S. Thus ∂W is a subset of S, and necessarily has parity-s ones.

And as we shall see presently, ∂W satisfies the first three conditions of an h-boundary B — (i) every path from $(x,0)$ to (x,ν) intersects B, (ii) $B \cap U_{x,h/4} = \emptyset$, (iii) $B \subseteq U_{x,h/2}$ — but not necessarily the final one, namely: (iv) the induced graph $C_\nu[B]$ is connected. (There may be islands of vertices of ∂W lying outside the h-boundary we are trying to home in on.) However, we can ensure (iv) by defining B to be the subset of vertices in ∂W that can be reached from $(x,0)$ by a path in C_ν whose vertices all lie in $V(C_\nu) \setminus W$.

For (i), observe that any path from $(x,0)$ to (x,ν) has a first vertex w in W. The vertex immediately preceding w is not in W but is adjacent to a vertex in W, and hence in B. (ii) follows from the fact that every vertex in $W \cap U_{x,h/2}$ is within $*$-distance $h/8$ of a vertex in $B_{x,h/2}$. (iii) is immediate from the construction. To see (iv), denote by W° the set of all vertices in $V(C_\nu)$ that can be reached from $(x,0)$ by a path whose vertices all lie in $V(C_\nu) \setminus W$. Note that $C_\nu[W^\circ]$ is connected and that $B \subseteq W^\circ$. Let ϱ^* be the set of edges in C_ν^* separating W° and W; thus, $e^* \in \varrho^*$ iff e has one endpoint in W and the other in W°. Since $C_\nu[W^\circ]$ is connected, the edges in ϱ^* form a trail in C_ν^* starting and ending at vertices with y-coordinate $-\frac{1}{2}$. Following this trail anticlockwise, vertices in W° lie to the left and those in W to the right. In fact, the vertices immediately to the left of the ϱ^*-trail (i.e., those at $*$-distance $\frac{1}{2}$ from it) are precisely the vertices forming B: they are all $*$-adjacent to some vertex in W, and no other vertices in W° have this property. Thus, any two vertices in B are connected by a path, which is obtained by shadowing ϱ^* at $*$-distance $\frac{1}{2}$.

For conclusion (2) of the lemma, observe that there cannot be an h-boundary B' of $(x,0)$ such that $\sigma(B')$ has parity-$(s\oplus 1)$ ones, as such a B' would have to exist entirely within $\overline{S}$, and all $*$-connected components of $\overline{S}$ are small ($*$-diameter at most $h/8$).

As for conclusion (3), it is impossible for $s' \neq s$. Consider the connected component of $C_\nu[S]$ containing B. If $s' \neq s$ then the boundary of this component contains a contour separating B and B', and hence $U_{x,h/4}$ and $U_{x',h/4}$. If $\sigma(B_{0,\nu})$ has parity-s' ones for $s' \neq s$ then the boundary of the connected component of $C_\nu[S]$ containing B is a simple contour separating B from the upper boundary of C_ν hence, separating $U_{x,h/4}$ from the upper boundary of C_ν, establishing (4). □

Corollary 2.20. *Suppose that $k \geq 1$ that d is a sufficiently large multiple of 8 and that $\nu = 2dk$. Suppose $\sigma\colon V(C_\nu) \to \{0,1\}$ has no contours of length at least $d/8$. Either σ has phase 0 or σ has phase 1.*

Proof. Since there are no contours of length at least $d/8$ of any kind, the premises of Lemma 2.19 are all satisfied for every terminal $(x,0)$ and every other terminal $(x',0)$. □

The following monotonicity property is useful for comparing different contours and boundary conditions.

Lemma 2.21. *Let $x \in \mathbb{Z}/2\nu\mathbb{Z}$, let B be an h-boundary of $(x,0)$ for some h, and let B' be an h'-boundary of $(x,0)$ for some h' such that B is inside of B'. Then*

$$\begin{aligned}&\Pr(\boldsymbol{\sigma}_\nu \textit{ has parity-}0 \textit{ ones at } (x,0) \mid \boldsymbol{\sigma}_\nu(B) \textit{ has parity-}0 \textit{ ones})\\ &\qquad \geq \Pr(\boldsymbol{\sigma}_\nu \textit{ has parity-}0 \textit{ ones at } (x,0) \mid \boldsymbol{\sigma}_\nu(B') \textit{ has parity-}0 \textit{ ones}).\end{aligned}$$

Proof. For each $S \subseteq V(C_\nu)$ let $\sigma^S\colon C_\nu \to \{0,1\}$ denote the configuration which has parity-0 ones exactly on S. So $\sigma^S(x,y) = 1$ if and only if one of these two conditions hold: $(x,y) \in S$ and $x+y$ is even, or $(x,y) \notin S$ and $x+y$ is odd.

For all $X, Y \subseteq V(C_\nu)$ we have

$$w_{C_\nu}(\sigma^X)w_{C_\nu}(\sigma^Y) = w_{C_\nu}(\sigma^{X\cap Y})w_{C_\nu}(\sigma^{X\cup Y})(\beta\gamma)^k$$

where k is the number of edges $uv \in E(G)$ such that $\{(\sigma^X(u), \sigma^X(v)), (\sigma^Y(u), \sigma^Y(v))\} = \{(0,0),(1,1)\}$ (so either $u \in X$ and $v \notin X$ and $u \notin Y$ and $v \in Y$, or $v \in X$ and $u \notin X$ and $v \notin Y$ and $u \in Y$). In particular,

$$w_{C_\nu}(\sigma^X)w_{C_\nu}(\sigma^Y) \leq w_{C_\nu}(\sigma^{X\cap Y})w_{C_\nu}(\sigma^{X\cup Y}).$$

Let

$$\begin{aligned}\mathcal{X} &= \{S \mid \{(x,0)\} \cup B' \subseteq S \subseteq V(C_\nu)\}\\ \mathcal{Y} &= \{S \mid B \cup B' \subseteq S \subseteq V(C_\nu)\}\end{aligned}$$

By the FKG inequality [54] we have

$$\left(\sum_{S\in\mathcal{X}} w_{C_\nu}(\sigma^S)\right)\left(\sum_{S\in\mathcal{Y}} w_{C_\nu}(\sigma^S)\right) \leq \left(\sum_{S\in\mathcal{X}\wedge\mathcal{Y}} w_{C_\nu}(\sigma^S)\right)\left(\sum_{S\in\mathcal{X}\vee\mathcal{Y}} w_{C_\nu}(\sigma^S)\right)$$

where $\mathcal{X}\wedge\mathcal{Y}$ is the family of sets $X \subseteq V(C_\nu)$ such that $(\{(x,0)\}\cup B')\cap(B\cup B') = B' \subseteq X$, and $\mathcal{X}\vee\mathcal{Y}$ is the family of sets $X \subseteq V(C_\nu)$ such that $\{(x,0)\} \cup B \cup B' \subseteq X$. Finally,

$$\begin{aligned}\Pr(\boldsymbol{\sigma}_\nu \text{ has parity-0 ones at } (x,0) \mid \boldsymbol{\sigma}_\nu(B') \text{ has parity-0 ones}) &= \frac{\sum_{S\in\mathcal{X}} w_{C_\nu}(\sigma^S)}{\sum_{S\in\mathcal{X}\wedge\mathcal{Y}} w_{C_\nu}(\sigma^S)}\\ \Pr(\boldsymbol{\sigma}_\nu \text{ has parity-0 ones at } (x,0) \mid \boldsymbol{\sigma}_\nu(B) \text{ has parity-0 ones}) &= \frac{\sum_{S\in\mathcal{X}\vee\mathcal{Y}} w_{C_\nu}(\sigma^S)}{\sum_{S\in\mathcal{Y}} w_{C_\nu}(\sigma^S)}.\square\end{aligned}$$

Lemma 2.22. *There is a $c > 1$ such that the following is true for any $k \geq 1$, any $s \in \{0,1\}$, any sufficiently large d which is a multiple of 16, and any assignment $\{B_x\}$ of d-boundaries for each terminal $(x,0)$:*

- *For every parity-s terminal $(x,0)$,*

$$|\Pr(\boldsymbol{\sigma}_{k,d}(x,0) = 1 \mid \boldsymbol{\sigma}_{k,d} \textit{ has phase } s \textit{ and } \mathcal{B}_x(\boldsymbol{\sigma}_{k,d}) = B_x) - p^{=}| \leq c^{-d}.$$

- *For every parity-$(1\oplus s)$ terminal $(x,0)$,*

$$|\Pr(\boldsymbol{\sigma}_{k,d}(x,0) = 1 \mid \boldsymbol{\sigma}_{k,d} \textit{ has phase } s \textit{ and } \mathcal{B}_x(\boldsymbol{\sigma}_{k,d}) = B_x) - p^{\neq}| \leq c^{-d}.$$

Proof. By symmetry (rotating the gadget so that parity-0 vertices become parity-1 vertices and vice-versa), it suffices to prove the inequalities for $s = 0$. For any $m \geq 1$, define

$$p^{=}(m) = \Pr(\boldsymbol{\sigma}_m(0,0) = 1 \mid \boldsymbol{\sigma}_m(B_{0,m}) \text{ has parity-0 ones}), \text{and}$$
$$p^{\neq}(m) = \Pr(\boldsymbol{\sigma}_m(1,0) = 1 \mid \boldsymbol{\sigma}_m(B_{1,m}) \text{ has parity-0 ones}).$$

Now note that for any $\nu \geq m$, $p^{=}(m)$ and $p^{\neq}(m)$ (as defined above) are the same as the equivalent expressions in the gadget C_ν. In particular,

$$p^{=}(m) = \Pr(\boldsymbol{\sigma}_\nu(0,0) = 1 \mid \boldsymbol{\sigma}_\nu(B_{0,m}) \text{ has parity-0 ones}), \text{and}$$
$$p^{\neq}(m) = \Pr(\boldsymbol{\sigma}_\nu(1,0) = 1 \mid \boldsymbol{\sigma}_\nu(B_{1,m}) \text{ has parity-0 ones}).$$

Thus, by fixing large ν and increasing m, Lemma 2.21 implies that $p^{=}(m)$ is weakly decreasing in m and that $p^{\neq}(m)$ is weakly increasing. Thus, $p^{=} = \lim_{m\to\infty} p^{=}(m)$ and $p^{\neq} = \lim_{m\to\infty} p^{\neq}(m)$. Also, for a parity-0 terminal $(x, 0)$, the target probability

$$\Pr(\boldsymbol{\sigma}_{k,d}(x,0) = 1 \mid \boldsymbol{\sigma}_{k,d} \text{ has phase 0 and } \mathcal{B}_x(\boldsymbol{\sigma}_{k,d}) = B_x)$$

is between $p^{=}(d/2)$ and $p^{=}(d/4)$. Similarly, for a parity-1 terminal $(x, 0)$, the target probability

$$\Pr(\boldsymbol{\sigma}_{k,d}(x,0) = 1 \mid \boldsymbol{\sigma}_{k,d} \text{ has phase 0 and } \mathcal{B}_x(\boldsymbol{\sigma}_{k,d}) = B_x)$$

is between $p^{\neq}(d/4)$ and $p^{\neq}(d/2)$. (Here we use crucially the canonicity property of $\mathcal{B}_x(\cdot)$; refer to the discussion following Definition 2.16.) Thus it suffices to show

$$p^{=}(d/4) \leq p^{=} + c^{-d} \quad \text{and} \quad p^{\neq}(d/4) \geq p^{\neq} - c^{-d},$$

First we take a qualitative step. Pick $w \geq 8d$ sufficiently large that $p^{=}(w) \leq p^{=} + |U_{0,d/4}|(c')^{-d/16}$ and $p^{\neq}(w) \geq p^{\neq} - |U_{1,d/4}|(c')^{-d/16}$, where c' is the maximum of the constants given in Lemma 2.10 and Lemma 2.11. This can be done since d is sufficiently large and $p^{=} = \lim_{m\to\infty} p^{=}(m)$, and $p^{\neq} = \lim_{m\to\infty} p^{\neq}(m)$ though w may be quite a lot larger than d.

For $i \in \{0, 1\}$, let F_i be the event that there is a $d/2$-boundary B of vertex $(i, 0)$ in gadget C_w such that $\boldsymbol{\sigma}_w(B)$ has parity-0 ones. Recall from the definition that a $d/2$-boundary of $(i, 0)$ is a subset of $U_{i,d/4}$. Let E_i be the event that $\boldsymbol{\sigma}_w(B_{i,w})$ has parity-0 ones.

For each $i \in \{0, 1\}$, applying Lemma 2.13, and Lemma 2.10 with $h = d/16$ and $U = U_{i,d/4}$, and Lemma 2.11 with $h = d/8$, we find that, the conditional probability that the following hold, conditioned on E_i, is at least $1 - 2|U_{i,d/4}|(c')^{-d/16}$.

- $\boldsymbol{\sigma}_w$ has no cross contour.
- $\boldsymbol{\sigma}_w$ has no simple contour of length at least $d/16$ which contains an edge between two vertices in $U_{i,d/4}$.

- $\boldsymbol{\sigma}_w$ has no simple contour that separates $U_{i,d/8}$ from the upper boundary of C_ν.

Now, applying Lemma 2.19 with $h = d/2$ and $\nu = w$ and $x = i$, if all of these hold and event E_i occurs then event F_i occurs. Thus,

$$\Pr(F_i \mid E_i) \geq 1 - 2|U_{i,d/4}|(c')^{-d/16}.$$

But by Lemma 2.21, we have

$$\begin{aligned}
&\Pr(\boldsymbol{\sigma}_w(0,0) = 1 \mid F_0 \wedge E_0) \geq p^{=}(d/4), \text{ and} \\
&\Pr(\boldsymbol{\sigma}_w(1,0) = 1 \mid F_1 \wedge E_1) \leq p^{\neq}(d/4).
\end{aligned}$$

So

$$\begin{aligned}
p^{=}(d/4) &\leq \frac{\Pr(\boldsymbol{\sigma}_w(0,0) = 1 \wedge F_0 \mid E_0)}{\Pr(F_0 \mid E_0)} \leq \frac{\Pr(\boldsymbol{\sigma}_w(0,0) = 1 \mid E_0)}{\Pr(F_0 \mid E_0)} = \frac{p^{=}(w)}{\Pr(F_0 \mid E_0)} \\
&\leq \frac{p^{=}(w)}{1 - 2|U_{0,d/4}|(c')^{-d/16}} \leq p^{=}(w) + 4|U_{0,d/4}|(c')^{-d/16},
\end{aligned}$$

since $2|U_{0,d/4}|(c')^{-d/16} \leq 1/2$.

A similar inequality holds for $p^{\neq}(d/4)$:

$$\begin{aligned}
p^{\neq}(d/4) &\geq \frac{\Pr(\boldsymbol{\sigma}_w(1,0) = 1 \wedge F_1 \mid E_1)}{\Pr(F_1 \mid E_1)} \\
&\geq \Pr(\boldsymbol{\sigma}_w(1,0) = 1 \mid E_1) - \Pr(\neg F_1 \mid E_1) \\
&\geq p^{\neq}(w) - 2|U_{1,d/4}|(c')^{-d/16}.
\end{aligned}$$

Thus,

$$\begin{aligned}
p^{=}(d/4) &\leq p^{=}(w) + 4|U_{0,d/4}|(c')^{-d/16} \leq p^{=} + 5|U_{0,d/4}|(c')^{-d/16}, \text{ and} \\
p^{\neq}(d/4) &\geq p^{\neq}(w) - 2|U_{1,d/4}|(c')^{-d/16} \geq p^{\neq} - 3|U_{1,d/4}|(c')^{-d/16}.
\end{aligned}$$

The result follows by noting that $|U_{i,d/4}|$ is $O(d^2)$ and picking $c = (c')^{1/17}$, say. □

We now prove the main proposition.

Proposition 2.5. There is a $c > 1$ such that, if d is a sufficiently large multiple of 16, k is an integer greater than or equal to 1 and τ is a configuration $\tau : T_{k,d} \to \{0,1\}$, then

$$|\Pr(\boldsymbol{\sigma}_{k,d}(T_{k,d}) = \tau) - \mu_{k,d}(\tau)| \leq c^{-d}k^2.$$

Proof. Fix $k \geq 1$, d a sufficiently large multiple of 16, and $\tau : T_{k,d} \to \{0,1\}$. Let c' be the minimum value of the constant c from the Lemmas 2.9, 2.12, and 2.22.

The probability $\Pr(\boldsymbol{\sigma}_{k,d}(T_{k,d}) = \tau)$ is the sum of the following probabilities (conditioned on disjoint events)

- $\Pr(\boldsymbol{\sigma}_{k,d}(T_{k,d}) = \tau \mid \boldsymbol{\sigma}_{k,d}$ does not have a phase$)\Pr(\boldsymbol{\sigma}_{k,d}$ does not have a phase$)$.

- (summed over all assignments B_x of d-boundaries for each terminal $(x, 0)$)

$$\Pr(\boldsymbol{\sigma}_{k,d}(T_{k,d}) = \tau \mid \ \boldsymbol{\sigma}_{k,d} \text{ has phase 0 and for all terminals } (x,0),\ \mathcal{B}_x(\boldsymbol{\sigma}_{k,d}) = B_x) \times$$
$$\Pr(\ \boldsymbol{\sigma}_{k,d} \text{ has phase 0 and for all terminals } (x,0),\ \mathcal{B}_x(\boldsymbol{\sigma}_{k,d}) = B_x)$$

- (summed over all assignments B_x of d-boundaries for each terminal $(x, 0)$)

$$\Pr(\boldsymbol{\sigma}_{k,d}(T_{k,d}) = \tau \mid \ \boldsymbol{\sigma}_{k,d} \text{ has phase 1 and for all terminals } (x,0),\ \mathcal{B}_x(\boldsymbol{\sigma}_{k,d}) = B_x) \times$$
$$\Pr(\ \boldsymbol{\sigma}_{k,d} \text{ has phase 1 and for all terminals } (x,0),\ \mathcal{B}_x(\boldsymbol{\sigma}_{k,d}) = B_x)$$

By Lemmas 2.9 and 2.12 and Corollary 2.20, the probability of the first of these is at most $2|V(C^*_{k,d})|(c')^{-d/8}$. (We will use this below.)

Now consider an assignment B_x of d-boundaries for each terminal $(x, 0)$. For any two terminals $(x', 0)$ and $(x'', 0)$, the random variables $\boldsymbol{\sigma}_{k,d}(x', 0)$ are $\boldsymbol{\sigma}_{k,d}(x'', 0)$ are independent, conditioned on the fact that $\boldsymbol{\sigma}_{k,d}$ has a given phase, and for all terminals $(x, 0)$, $\mathcal{B}_x(\boldsymbol{\sigma}_{k,d}) = B_x$. Also, by Lemma 2.22, for all $s \in \{0, 1\}$,

- For every parity-s terminal $(x', 0)$,

$$\left|\Pr(\boldsymbol{\sigma}_{k,d}(x',0) = 1 \mid \ \boldsymbol{\sigma}_{k,d} \text{ has phase } s \text{ and for all terminals } (x,0),\ \mathcal{B}_x(\boldsymbol{\sigma}_{k,d}) = B_x) - p^{=}\right| \le (c')^{-d}.$$

- For every parity-$(1 \oplus s)$ terminal $(x', 0)$,

$$\left|\ \Pr(\boldsymbol{\sigma}_{k,d}(x',0) = 1 \mid \ \boldsymbol{\sigma}_{k,d} \text{ has phase } s \text{ and for all terminals } (x,0),\ \mathcal{B}_x(\boldsymbol{\sigma}_{k,d}) = B_x) - p^{\neq}\right| \le (c')^{-d}.$$

Now, for any probabilities $a_1, b_1, \dots, a_k, b_k$, we have

$$\left|\prod_{i=1}^{k} a_i - \prod_{i=1}^{k} b_i\right| = \left|\sum_{j=1}^{k} a_1 \dots a_{j-1}(a_j - b_j)b_{j+1} \dots b_k\right| \le \sum_{i=1}^{k} |a_i - b_i|\,,$$

so if we fix a given phase, and τ assigns spin 1 to k' terminals whose parity agrees with that phase, and spin 1 to k'' terminals whose parity disagrees with that phase then, letting

$$\hat{p} = (p^{=})^{k'}(1 - p^{=})^{k-k'}(p^{\neq})^{k''}(1 - p^{\neq})^{k-k''},$$

we have

$$\begin{aligned}
&\hat{p} - 2k(c')^{-d} \\
&\quad \le \Pr(\boldsymbol{\sigma}_{k,d}(T_{k,d}) = \tau \mid \ \boldsymbol{\sigma}_{k,d} \text{ has the given phase and for all terminals } (x,0),\ \mathcal{B}_x(\boldsymbol{\sigma}_{k,d}) = B_x) \\
&\quad \le \hat{p} + 2k(c')^{-d}.
\end{aligned}$$

So summing up, $\Pr(\boldsymbol{\sigma}_{k,d}(T_{k,d}) = \tau \mid \boldsymbol{\sigma}_{k,d}$ has the given phase) is between $\hat{p}-2k(c')^{-d}$ and $\hat{p}+2k(c')^{-d}$ so, since the phases are equally likely,

$$\mu_{k,d}(\tau) - 2k(c')^{-d} \le \Pr(\boldsymbol{\sigma}_{k,d}(T_{k,d}) = \tau \mid \boldsymbol{\sigma}_{k,d} \text{ has a phase}) \le \mu_{k,d}(\tau) + 2k(c')^{-d}.$$

Finally, since the probability that $\boldsymbol{\sigma}_{k,d}$ has no phase is at most $2|V(C^*_{k,d})|(c')^{-d/8}$, as we observed above,

$$\mu_{k,d}(\tau) - 2(k+|V(C^*_{k,d})|)(c')^{-d/8} \le \Pr(\boldsymbol{\sigma}_{k,d}(T_{k,d}) = \tau) \le \mu_{k,d}(\tau) + 2(k+|V(C^*_{k,d})|)(c')^{-d/8}.$$

The proposition follows by choosing c to be sufficiently small with respect to c'. □

2.4 Proof of Theorem 2.1

2.4.1 Efficiently approximable reals

Lemma 2.23. *Suppose that β, γ and λ are efficiently approximable reals satisfying* (2.1). *Then $p^{=}$ and $p^{\neq}$ are efficiently approximable reals.*

Proof. Recall that $p^{\neq} > 0$ (Lemma 2.4). Let q be a multiple of 16 greater than $(2 + \log_2(1/p^{\neq}))/\log(c)$ where c is the constant given by Lemma 2.22. Consider the following algorithm.

- Input an error parameter $0 < \varepsilon < 1/2$.
- Set $m = q\lceil \log(\varepsilon^{-1}) \rceil$.
- Compute rational approximations $\hat{\beta}, \hat{\gamma}, \hat{\lambda}$ satisfying

$$\begin{aligned} \beta e^{-\varepsilon/8|E(C_m)|} &\le \hat{\beta} \le \beta e^{\varepsilon/8|E(C_m)|} \\ \gamma e^{-\varepsilon/8|E(C_m)|} &\le \hat{\gamma} \le \gamma e^{\varepsilon/8|E(C_m)|} \\ \lambda e^{-\varepsilon/8|V(C_m)|} &\le \hat{\lambda} \le \lambda e^{\varepsilon/8|V(C_m)|}. \end{aligned}$$

- Using the algorithm of [104, Theorem 2.2], compute

$$\begin{aligned} Z &= \sum_{\sigma} \hat{\beta}^{b(\sigma)} \hat{\gamma}^{c(\sigma)} \hat{\lambda}^{\ell(\sigma)} \\ Z' &= \sum_{\sigma:\sigma(0,0)=1} \hat{\beta}^{b(\sigma)} \hat{\gamma}^{c(\sigma)} \hat{\lambda}^{\ell(\sigma)} \\ Z'' &= \sum_{\sigma:\sigma(1,0)=1} \hat{\beta}^{b(\sigma)} \hat{\gamma}^{c(\sigma)} \hat{\lambda}^{\ell(\sigma)}, \end{aligned}$$

  where the sums range over configurations σ of C_m such that $\sigma(B_{0,m})$ has parity-0 ones.

- Output Z'/Z as the approximation to $p^{=}$, and Z''/Z as the approximation to $p^{\neq}$.

For the computation of Z, Z', and Z'' we use the fact that the grid graph $C_m \setminus B_{0,m}$ has treewidth m [7, Corollary 89]. We also use the fact that its tree decomposition is easy to compute. So this algorithm runs in time bounded by a polynomial in $1/\varepsilon$. We will show that the algorithm is an FPRAS for $p^{=}$ and $p^{\neq}$. Define

$$\begin{aligned} W &= \sum_{\sigma} \beta^{b(\sigma)}\gamma^{c(\sigma)}\lambda^{\ell(\sigma)} \\ W' &= \sum_{\sigma:\sigma(0,0)=1} \beta^{b(\sigma)}\gamma^{c(\sigma)}\lambda^{\ell(\sigma)} \\ W'' &= \sum_{\sigma:\sigma(1,0)=1} \beta^{b(\sigma)}\gamma^{c(\sigma)}\lambda^{\ell(\sigma)}, \end{aligned}$$

where the sums range over configurations σ of C_m such that $\sigma(B_{0,m})$ has parity-0 ones.

For any σ we have

$$\beta^{b(\sigma)}\gamma^{c(\sigma)}\lambda^{\ell(\sigma)}e^{-\varepsilon/4} \le \hat\beta^{b(\sigma)}\hat\gamma^{c(\sigma)}\hat\lambda^{\ell(\sigma)} \le \beta^{b(\sigma)}\gamma^{c(\sigma)}\lambda^{\ell(\sigma)}e^{\varepsilon/4}$$

This implies $e^{-\varepsilon/4}W \le Z \le e^{\varepsilon/4}W$ and similarly for Z' and Z'', and therefore $e^{-\varepsilon/2}W'/W \le Z'/Z \le e^{\varepsilon/2}W'/W$ and $e^{-\varepsilon/2}W''/W \le Z''/Z \le e^{\varepsilon/2}W''/W$. We will show

$$p^{=} \le W'/W \le p^{=}e^{\varepsilon/2} \tag{2.3}$$

$$e^{-\varepsilon/2}p^{\neq} \le W''/W \le p^{\neq} \tag{2.4}$$

W'/W and W''/W are just the probabilities that an even or odd terminal gets assigned 1 in a random configuration of C_m, conditioned on a certain $2m$-boundary. By Lemma 2.21 we have $p^{=} \le W'/W$ and $W''/W \le p^{\neq}$ for any m, establishing the first inequality in (2.3) and the second inequality in (2.4).

By Lemma 2.22, there exists $c > 1$ such that

$$W'/W \le p^{=} + c^{-q\log(\varepsilon^{-1})} = p^{=}(1 + \varepsilon^{q\log(c)}/p^{=}).$$

Since

$$\varepsilon^{(q\log(c)-1)} \le (1/2)^{(q\log(c)-1)} \le p^{\neq}/2,$$

which is less than $p^{=}$ by Lemma 2.14, we have

$$W'/W \le p^{=}(1+\varepsilon) \le e^{\varepsilon}.$$

This establishes (2.3). Similarly, by Lemma 2.22,

$$\begin{aligned} W''/W &\ge p^{\neq} - c^{-q\log(\varepsilon^{-1})} \\ &\ge p^{\neq}(1 - \varepsilon^{q\log(c)}/p^{\neq}) \\ &\ge p^{\neq}(1-\varepsilon/2) \\ &\ge p^{\neq}e^{-\varepsilon} \end{aligned}$$

This establishes (2.4). □

Lemma 2.23 gives us a way to obtain multiplicative approximations $\widehat{p^{=}}$ and $\widehat{p^{\neq}}$ of the real numbers $p^{=}$ and $p^{\neq}$. When we use these approximations, we will need to know that $1-\widehat{p^{=}}$ and $1-\widehat{p^{\neq}}$ are also good multiplicative approximations to $1-p^{=}$ and $1-p^{\neq}$, respectively. As we show below, this follows from the fact that $p^{=}$ and $p^{\neq}$ are in $(0,1)$ (which follows from Lemma 2.4 and Lemma 2.14). The following lemma gives us what we need. The reason for introducing the rational p' in the statement of the lemma is that, since it is rational, it can be hard-wired into any algorithms (whereas a real number can't be).

Lemma 2.24. *Suppose that $p \in (0,1)$ is an efficiently approximable real number. Let p' be a positive rational with $p < p' < 1$. For any $\delta \in (0,1)$, and any real number $\widehat{p}$ satisfying $e^{-\delta(1-p')/2}\widehat{p} \le p \le e^{\delta(1-p')/2}\widehat{p}$, we have $e^{-\delta}(1-p) \le 1-\widehat{p} \le e^{\delta}(1-p)$.*

Proof. Let $\delta' = \delta(1-p')/2$. Since $\widehat{p} \ge e^{-\delta'}p \ge p(1-\delta') \ge p-\delta'$ and similarly $p \ge \widehat{p}-\delta'$, we have

$$(1-p)\left(1-\frac{\delta(1-p')}{2(1-p)}\right) = (1-p)-\delta' \le 1-\widehat{p} \le (1-p)+\delta' = (1-p)\left(1+\frac{\delta(1-p')}{2(1-p)}\right).$$

Thus,

$$(1-p)(1-\delta/2) \le 1-\widehat{p} \le (1-p)(1+\delta/2),$$

which suffices. □

The following problem is NP-complete [58].

Name. PlanarCubicIS.
Instance. A planar cubic graph G and a positive integer h.
Output. "Yes", if G contains an independent set of size h, and "No", otherwise.

Suppose that β, γ and λ are efficiently approximable reals satisfying (2.1). We will give a randomised polynomial-time algorithm for PlanarCubicIS, using as an oracle, an FPRAS for DegreeFourPlanarTwoSpin(β,γ,λ). The oracle will be used to approximate $Z_{1,\tilde{\gamma},\tilde{\lambda}}(G)$, for some suitably-defined parameters $\tilde{\gamma}$ and $\tilde{\lambda}$, where $\tilde{\gamma}$ is exponentially small in $|V(G)|$ and $\tilde{\lambda}$ is exponentially large. From this, it will be easy to determine whether G has an independent set of size h.

Lemma 2.25. *Suppose that β, γ and λ are efficiently approximable reals satisfying* (2.1). *There is a polynomial-time randomised algorithm that, given a planar cubic graph G with $|V(G)|$ sufficiently large, outputs planar graphs J and J' with maximum degree at most* 4 *and randomised approximation schemes for positive reals K, $\tilde{\gamma}$ and $\tilde{\lambda}$. The running time of each of these approximation schemes is bounded from above by a polynomial in $|V(G)|$ and the desired accuracy parameter ε. With probability at least* $14/15$, *the parameters satisfy $\tilde{\lambda} \ge 4^{|V(G)|}$ and $\tilde{\gamma} \le \tilde{\lambda}^{-|V(G)|}$ and*

$$e^{-1/4}Z_{1,\tilde{\gamma},\tilde{\lambda}}(G) \le K\frac{Z_{\beta,\gamma,\lambda}(J')}{Z_{\beta,\gamma,\lambda}(J)} \le e^{1/4}Z_{1,\tilde{\gamma},\tilde{\lambda}}(G). \tag{2.5}$$

Proof. Let $G = (V, E)$ be a planar cubic graph and let n denote $|V|$.

The algorithm for constructing J and J' uses a quantity $\delta \in (0, 1)$. It will be important for the proof that δ is sufficiently small. Rather than giving a technical definition here, we introduce upper bounds on δ in natural places throughout the proof. The reader can verify that each of these upper bounds is at least the inverse of a polynomial in n (so the algorithm runs in polynomial time).

The first step is to use the given FPRASes for β, γ and λ, and the FPRASes for $p^=$ and $p^{\neq}$ from Lemma 2.23 to compute values $\widehat{\beta}$, $\widehat{\gamma}$, $\widehat{\lambda}$, $\widehat{p^=}$ and $\widehat{p^{\neq}}$ satisfying

$$
\begin{aligned}
e^{-\delta/3}\beta \le \widehat{\beta} &\le e^{\delta/3}\beta, \\
e^{-\delta/3}\gamma \le \widehat{\gamma} &\le e^{\delta/3}\gamma, \\
e^{-\delta/3}\lambda \le \widehat{\lambda} &\le e^{\delta/3}\lambda, \\
e^{-\delta/3}p^= \le \widehat{p^=} &\le e^{\delta/3}p^=, \\
e^{-\delta/3}p^{\neq} \le \widehat{p^{\neq}} &\le e^{\delta/3}p^{\neq} \\
e^{-\delta/3}(1-p^=) \le 1-\widehat{p^=} &\le e^{\delta/3}(1-p^=), \\
e^{-\delta/3}(1-p^{\neq}) \le 1-\widehat{p^{\neq}} &\le e^{\delta/3}(1-p^{\neq}). \\
\widehat{\beta} &\ge 1.
\end{aligned}
\tag{2.6}
$$

The first five lines in (2.6) follow directly from the definition of FPRAS in Section 1.2. The next two lines follow from Lemma 2.24, using the fact that $p^=$ and $p^{\neq}$ are in $(0, 1)$, as argued just before Lemma 2.24. Since $\beta \ge 1$ by (2.1), we can ensure that $\widehat{\beta} \ge 1$ by taking $\widehat{\beta}$ to be the maximum of 1 and the output of the FPRAS. For this step we adjust the failure probability of the FPRASes (as described in Section 1.2) so that the probability that Equation (2.6) fails to hold is at most 1/15. Note that the running time of the FPRASes is polynomial in $1/\delta$ (even though the application of Lemma 2.24 means that we have to call the FPRASes for $p^=$ and $p^{\neq}$ with slightly smaller values δ'.).

We will show below how to use G and these approximations to define positive integers k_1, k_2 and d, which will be used in the construction of J and J'. These quantities will be bounded from above by a polynomial in n.

We first show how to construct J and J', using k_1, k_2, d and $k = \max(k_1, 3k_2)$. The high-level construction is illustrated in Figure 2.6.

The construction of J is straightforward. Essentially, J consists of $|V|$ copies of $C_{k,d}$, with one copy for every vertex in V. Thus, the vertex set $V(J)$ is the set of ordered pairs $V(J) = V \times V(C_{k,d})$ and the edge set $E(J)$ is given by $E(J) = V \times E(C_{k,d})$. We will use $C[u]$ to denote the gadget corresponding to vertex $u \in V$. Formally, $C[u]$ is the graph with vertex set $\{u\} \times V(C_{k,d})$ and edge set $\{u\} \times E(C_{k,d})$. To simplify the notation, for $u \in V$ and $0 \le j \le k-1$, let $T^1[u, j]$ denote the j'th parity-1 terminal of $C[u]$. Formally, this is the vertex $(u, (4jd + 1, 0))$ of J. Similarly, let $T^0[u, j]$ denote the j'th parity-0 terminal of $C[u]$. Formally, this is the vertex $(u, (4jd + 2j, 0))$ of J. Let $T[u]$ be the set of terminals of $C[u]$. Let μ^0_u, μ^1_u and μ_u be the distributions on

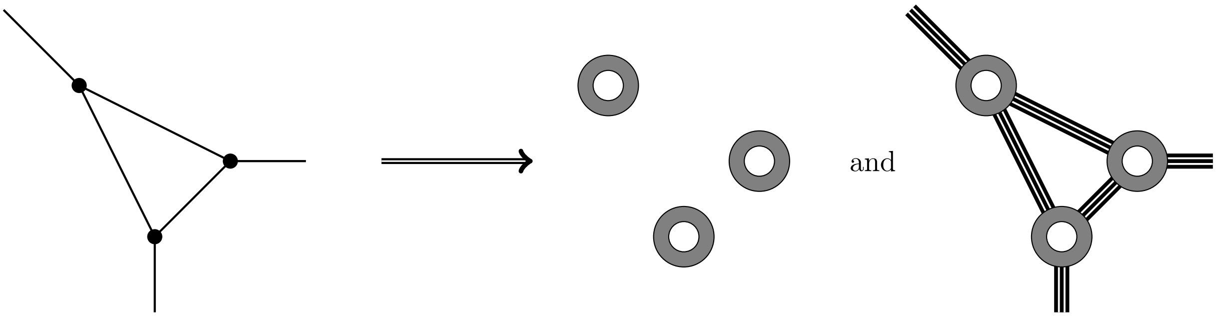

FIGURE 2.6: An illustration of how G is transformed into the graphs J and J'. A fragment of G is shown on the left. The graph J is a collection of copies of $C_{k,d}$, one for each vertex of G. A fragment of J is shown in the middle. The copies of $C_{k,d}$ are shown as grey annuli. The corresponding fragment of J' is shown on the right. The stripes represent the sets of edges between copies of $C_{k,d}$ in J'. J' also contains some "bristles" (described later) which are not shown.

configurations $\sigma : T[u] \to \{0,1\}$ corresponding to the distributions $\mu^0_{k,d}$, $\mu^1_{k,d}$ and μ defined in Section 2.3.4.

To simplify the description of J', consider a planar embedding of G in which each vertex u of G is associated with three "endpoints" u_0, u_1 and u_2, which are arranged together in clockwise order in the plane. The edge set E can then be viewed as a matching $\mathcal{M}$ on the points $\bigcup_{u \in V}\{u_0, u_1, u_2\}$ such that

- $(u,v) \in E$ if and only if there are exactly two points u_i and v_j such that $(u_i, v_j) \in \mathcal{M}$,
- No two edges of $\mathcal{M}$ cross.

The vertex set $V(J')$ consists of $V(J)$, together with a set of nk_1 new vertices, called "bristles". Formally, $V(J') = V(J) \cup \{(u,j) \mid u \in V, 0 \le j \le k_1 - 1\}$. Finally, the edge set of J' consists of $E(J)$, together with new edges connecting the bristles to the parity-1 terminals of the gadgets, and new edges matching the parity-0 terminals of the gadgets (guided by the matching $\mathcal{M}$). The edges connecting the bristles to parity-1 terminals of the gadgets are those in the set

$$E_B = \{((u,j), T^1[u,j]), u \in V, 0 \le j \le k_1 - 1\}.$$

It is more complicated to describe the edges matching the parity-0 terminals of the gadgets. The idea (see Figure 2.7) that if u_a is matched to v_b in $\mathcal{M}$ (where $a \in \{0,1,2\}$ and $b \in \{0,1,2\}$) then the parity-0 terminals $T^0[u, ak_2], \ldots, T^0[u, ak_2 + k_2 - 1]$ get matched to the parity-0 terminals $T^0[u, bk_2], \ldots, T^0[u, bk_2 + k_2 - 1]$. However, there is a further complication: To ensure that J' is planar we must ensure that one of these sequences of terminals is matched in clockwise order, and the other in anti-clockwise order. Thus, let

$$E_{\mathcal{M}} = \{(T^0[u, ak_2 + j], T^0[v, bk_2 + k_2 - 1 - j]) \mid u < v, (u_a, v_b) \in \mathcal{M}, 0 \le j \le k_2 - 1\}.$$

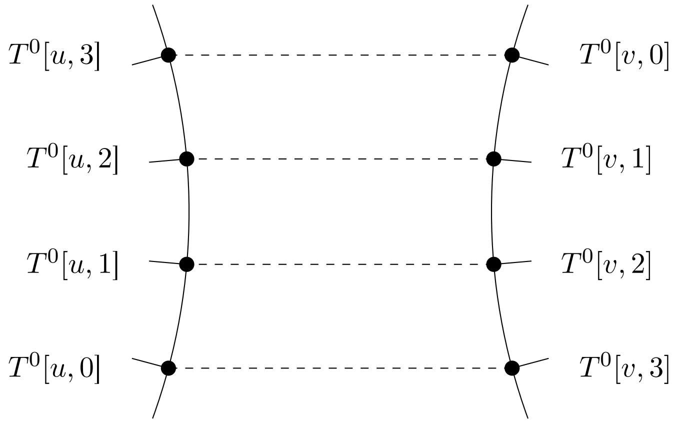

FIGURE 2.7: Terminals of u are matched to terminals in v, reversing the order.

Then $E(J') = E(J) \cup E_B \cup E_{\mathcal{M}}$. Note that both J and J' are planar as required.

We next show how to define the positive integers k_1, k_2 and d. Define

$$P = \begin{pmatrix} 1-p^{=} & p^{=} \\ 1-p^{\neq} & p^{\neq} \end{pmatrix}, \qquad M = P\begin{pmatrix} \beta & 1 \\ 1 & \gamma \end{pmatrix}P^t, \qquad W = P\begin{pmatrix} \beta & 1 \\ 1 & \gamma \end{pmatrix}\begin{pmatrix} 1 \\ \lambda \end{pmatrix}$$

where P^t denotes the transpose of the matrix P. Also, define

$$\widehat{P} = \begin{pmatrix} 1-\widehat{p^{=}} & \widehat{p^{=}} \\ 1-\widehat{p^{\neq}} & \widehat{p^{\neq}} \end{pmatrix}, \qquad \widehat{M} = \widehat{P}\begin{pmatrix} \widehat{\beta} & 1 \\ 1 & \widehat{\gamma} \end{pmatrix}\widehat{P}^t, \qquad \widehat{W} = \widehat{P}\begin{pmatrix} \widehat{\beta} & 1 \\ 1 & \widehat{\gamma} \end{pmatrix}\begin{pmatrix} 1 \\ \widehat{\lambda} \end{pmatrix}$$

Note that if (2.6) holds then, for any $s \in \{0,1\}$ and $s' \in \{0,1\}$,

$$\begin{aligned} e^{-\delta}P_{s,s'} &\le \widehat{P}_{s,s'} \le e^{\delta}P_{s,s'}, \\ e^{-\delta}M_{s,s'} &\le \widehat{M}_{s,s'} \le e^{\delta}M_{s,s'}, \\ e^{-\delta}W_{s} &\le \widehat{W}_{s} \le e^{\delta}W_{s}. \end{aligned} \tag{2.7}$$

The matrix M has the following informal interpretation. Suppose that two parity-0 terminals t and t' are adjacent in J'. and that $\sigma\colon V(J') \to \{0,1\}$ is a configuration. If these two terminals have spins $\sigma(t)$ and $\sigma(t')$, respectively, then the edge between them contributes a factor $\left(\begin{smallmatrix} \beta & 1 \\ 1 & \gamma \end{smallmatrix}\right)_{\sigma(t),\sigma(t')}$ to $w_{J'}(\sigma)$. We will show below that, if t is a terminal of $C[u]$ and the spins of $C[u]$ are chosen from the idealised distribution μ_u^s, then the probability that the spin of terminal t is j is $P_{s,j}$. Thus, informally, $M_{s,s'}$ captures the expected contribution of this connection (in the idealised distribution), where s' represents the phase of the gadget of terminal t'.

The informal interpretation of W is that, given any configuration $\sigma : V(J) \to \{0,1\}$, a parity-1 terminal t which is connected to a bristle b will contribute a factor $\left[\left(\begin{smallmatrix} \beta & 1 \\ 1 & \gamma \end{smallmatrix}\right)\left(\begin{smallmatrix} 1 \\ \lambda \end{smallmatrix}\right)\right]_{\sigma(t)}$ to the sum $\sum_{\sigma'} w_{J'}(\sigma')$, where the sum is over all configurations $\sigma' : V(J') \to \{0,1\}$ which agree with σ except possibly at the bristle b. This informal description is just to provide intuition — the technical details are given below. The main idea is that, if the spins of the terminals of the gadgets are chosen from the "idealised" distribution then, if the gadget of t has phase s, then the terminal t will contribute a factor of $W_{1\oplus s}$ to the expected contribution from this bristle.

We now introduce some calculation which will be needed to describe the algorithm's computation of k_1, k_2, and d and also to give the definitions of the real numbers $\tilde{\gamma}$ and $\tilde{\lambda}$. The first step is deriving some tedious but necessary bounds on the various quantities defined above. In particular, we will define positive rational numbers Δ^- and Δ^+ and a rational number $\xi \in (0,1)$ (independent of δ and n, but depending on β, γ and λ). These will be hard-wired into the algorithm. We will prove that, provided that δ is sufficiently small, each $\widehat{M}_{s,s'}$ and $\widehat{W}_s$ satisfies $\Delta^- \leq \widehat{M}_{s,s'} \leq \Delta^+$ and $\Delta^- \leq \widehat{W}_s \leq \Delta^+$. Also, each of $\widehat{p^=} - \widehat{p^{\neq}}$, $1-\widehat{\gamma}$, $\widehat{p^{\neq}}$ and $\widehat{\lambda}$ is at least ξ. We will also prove that $\widehat{p^{\neq}} \leq 1$ (and, from (2.6), we have $\widehat{\beta} \geq 1$.) Finally, we prove $\widehat{\beta}\widehat{\gamma} \leq 1-\xi$. Here are the details (which the reader may skip).

- By Lemmas 2.4 and 2.14, we can define positive rational numbers p^- and p^+ such that every element $P_{s,s'}$ of matrix P satisfies $p^- \leq P_{s,s'} \leq p^+$. Then

$$(p^-)^2(2+\beta+\gamma) \leq M_{s,s'} \leq (p^+)^2(2+\beta+\gamma),$$

  so, since $\delta < 1$,

$$e^{-1}(p^-)^2(2+\beta+\gamma) \leq \widehat{M}_{s,s'} \leq e(p^+)^2(2+\beta+\gamma),$$

  so to get the required bounds, we take any $\Delta^- < e^{-1}(p^-)^2(2+\beta+\gamma)$ and any $\Delta^+ > e(p^+)^2(2+\beta+\gamma)$. The bounds on $\widehat{W}_s$ are similar.

- To ensure $\widehat{p^=} - \widehat{p^{\neq}} \geq \xi$, choose rational numbers ρ_1, ρ_2, ρ_3, and ρ_4 such that $p^{\neq} < \rho_1 < \rho_2 < \rho_3 < \rho_4 < p^=$. These exist by Lemma 2.14, which guarantees that $p^{\neq} < p^=$. Then, if $\delta \leq \rho_4 - \rho_3$, Equation (2.6) guarantees $\widehat{p^=} \geq p^= e^{-\delta} \geq p^=(1-\delta) \geq p^= - \delta \geq p^= - (\rho_4 - \rho_3) \geq \rho_3$. (Note that the calculation used $p^= \leq 1$.) Similarly, if $\delta \leq (\rho_2-\rho_1)/2$, Equation (2.6) guarantees $\widehat{p^{\neq}} \leq e^{\delta} p^{\neq} \leq p^{\neq}(1+2\delta) \leq p^{\neq} + 2\delta \leq p^{\neq} + (\rho_2 - \rho_1) \leq \rho_2$. (Again, we used $p^{\neq} \leq 1$.) It suffices to take any $\xi \leq \rho_3 - \rho_2$.

- We can similarly establish $1-\widehat{\gamma} \geq \xi$ by considering a sequence of rational numbers between γ and 1 (using the fact that $\gamma < 1$ by (2.1)) and we can establish $\widehat{p^{\neq}} \geq \xi$ by considering a sequence of rational numbers between 0 and $p^{\neq}$ (using the fact that $p^{\neq} > 0$ by Lemma 2.4).

- Then, by (2.1), $\lambda > 0$, so taking any $\xi < \lambda$, we can choose a rational number ξ' with $0 < \xi < \xi' < \lambda$. Then choosing $\delta \leq \log(\xi'/\xi)$ ensures $e^{-\delta}\xi' \geq \xi$ so $\widehat{\lambda} \geq e^{-\delta}\lambda \geq e^{-\delta}\xi' \geq \xi$.

- Note that the second bullet point already establishes $\widehat{p^{\neq}} \leq 1$.

- Finally, (2.1) guarantees $\beta\gamma < 1$, so choose rationals $\beta' \geq \beta$ and $\gamma' \geq \gamma$ with $\beta'\gamma' < 1$. Choose ξ sufficiently small that $\beta'\gamma' \leq e^{-3\xi}$. Then choose $\delta \leq \xi/2$ to ensure
$$\widehat{\beta}\widehat{\gamma} \leq e^{2\delta}\beta\gamma \leq e^{\xi}\beta'\gamma' \leq e^{-2\xi} \leq 1-\xi.$$

We can make the following conclusions.

$$\begin{aligned}
\widehat{M}_{1,1} - \widehat{M}_{0,1} &= (\widehat{p^{=}} - \widehat{p^{\neq}})((\widehat{\beta}-1)(1-\widehat{p^{\neq}}) + (1-\widehat{\gamma})\widehat{p^{\neq}}) \geq \xi^3 \\
\widehat{M}_{0,1} - \widehat{M}_{0,0} &= (\widehat{p^{=}} - \widehat{p^{\neq}})((\widehat{\beta}-1)(1-\widehat{p^{=}}) + (1-\widehat{\gamma})\widehat{p^{=}}) \geq \xi^3 \\
\widehat{W}_1 - \widehat{W}_0 &= (\widehat{p^{=}} - \widehat{p^{\neq}})((\widehat{\beta}-1) + (1-\widehat{\gamma})\widehat{\lambda}) \geq \xi^3.
\end{aligned}$$

We can now define k_2. Since

$$\widehat{M}_{0,0}\widehat{M}_{1,1} - \widehat{M}_{0,1}^2 = \det(\widehat{M}) = \det(\widehat{P})^2(\widehat{\beta}\widehat{\gamma}-1) = (\widehat{p^{=}} - \widehat{p^{\neq}})^2(\widehat{\beta}\widehat{\gamma}-1) \leq -\xi^3, \text{so}$$

$$\frac{\widehat{M}_{0,0}\widehat{M}_{1,1}}{\widehat{M}_{0,1}^2} = \frac{\widehat{M}_{0,1}^2 + (\widehat{M}_{0,0}\widehat{M}_{1,1} - \widehat{M}_{0,1}^2)}{\widehat{M}_{0,1}^2} \leq \frac{\widehat{M}_{0,1}^2 - \xi^3}{\widehat{M}_{0,1}^2} = 1 - \frac{\xi^3}{\widehat{M}_{0,1}^2} \leq 1 - \frac{\xi^3}{(\Delta^+)^2} \leq e^{-\xi^3/(\Delta^+)^2}.$$

Then let
$$k_2 = \left\lceil \frac{(n^2+n)2\log(5)(\Delta^+)^2}{\xi^3} \right\rceil.$$

Then, if we ensure that $\delta < (\xi^3/(\Delta^+)^2)/8$, we have

$$\begin{aligned}
\left(e^{4\delta}\frac{\widehat{M}_{0,0}\widehat{M}_{1,1}}{\widehat{M}_{0,1}^2}\right)^{k_2} &\leq e^{-k_2\xi^3/(2(\Delta^+)^2)} \\
&\leq 5^{-n^2-n}.
\end{aligned}$$

Then define
$$\tilde{\gamma} = \left(\frac{M_{0,0}M_{1,1}}{M_{0,1}^2}\right)^{k_2}.$$

By (2.7),
$$\tilde{\gamma} \leq e^{4\delta k_2}\left(\frac{\widehat{M}_{0,0}\widehat{M}_{1,1}}{\widehat{M}_{0,1}^2}\right)^{k_2} \leq 5^{-n^2-n}.$$

Also, there is a randomised approximation scheme for $\tilde{\gamma}$ whose running time is at most a polynomial in n and in the desired accuracy parameter ε.

Next, we will define k_1. Recall that $\widehat{W}_1 > \widehat{W}_0$ and $\widehat{M}_{1,1} > \widehat{M}_{0,1}$. If n is sufficiently large with respect to $\log(\Delta^+/\Delta^-) \geq \log(\widehat{W}_1/\widehat{W}_0)$ then there is a positive integer k_1

(which the algorithm can compute) satisfying

$$\frac{3k_2\log(\widehat{M}_{1,1}/\widehat{M}_{0,1})}{\log(\widehat{W}_1/\widehat{W}_0)}+\frac{\log(4.1)n}{\log(\widehat{W}_1/\widehat{W}_0)}\le k_1\le\frac{3k_2\log(\widehat{M}_{1,1}/\widehat{M}_{0,1})}{\log(\widehat{W}_1/\widehat{W}_0)}+\frac{\log(4.9)n}{\log(\widehat{W}_1/\widehat{W}_0)}.$$

Note that $k_1 = O(n^2)$. Also,

$$(4.1)^n\le\left(\frac{\widehat{W}_1}{\widehat{W}_0}\right)^{k_1}\left(\frac{\widehat{M}_{0,1}}{\widehat{M}_{1,1}}\right)^{3k_2}$$

and

$$\left(\frac{\widehat{W}_1}{\widehat{W}_0}\right)^{k_1}\left(\frac{\widehat{M}_{0,1}}{\widehat{M}_{1,1}}\right)^{3k_2}\le(4.9)^n.$$

Now if we ensure $\delta \le n^{-2.5}$ then, for sufficiently large n, $\delta \le n\log(4.1/4)/(2k_1+6k_2)$ and $\delta \le n\log(5/4.9)/(2k_1+6k_2)$, so

$$4^n\le\left(e^{-2\delta}\frac{\widehat{W}_1}{\widehat{W}_0}\right)^{k_1}\left(e^{-2\delta}\frac{\widehat{M}_{0,1}}{\widehat{M}_{1,1}}\right)^{3k_2}$$

and

$$\left(e^{2\delta}\frac{\widehat{W}_1}{\widehat{W}_0}\right)^{k_1}\left(e^{2\delta}\frac{\widehat{M}_{0,1}}{\widehat{M}_{1,1}}\right)^{3k_2}\le 5^n.$$

Note that

$$5^n\le\frac{1}{\tilde{\gamma}^{1/n}}.$$

Then define

$$\tilde{\lambda}=\left(\frac{W_1}{W_0}\right)^{k_1}\left(\frac{M_{0,1}}{M_{1,1}}\right)^{3k_2}.$$

Note that there is a randomised approximation scheme for $\tilde{\lambda}$ whose running time is bounded from above by a polynomial in n and the desired accuracy parameter ε. Also, $\tilde{\lambda}\le\frac{1}{\tilde{\gamma}^{1/n}}$ so $\tilde{\gamma}\le\tilde{\lambda}^{-n}$ and $\tilde{\lambda}\ge 4^n$, as required.

Now let $k=\max(k_1,3k_2)$. Finally, the gadget will use a parameter d. By Proposition 2.5, there is a $c>1$ (not depending on k) such that, for all sufficiently large d which are multiples of 16, and all configurations $\tau : T_{k,d}\to\{0,1\}$,

$$|\Pr(\boldsymbol{\sigma}_{k,d}(T_{k,d})=\tau)-\mu_{k,d}(\tau)|\le c^{-d}k^2.$$

The algorithm will choose d to be a multiple of 16 such that $d=O(n^3)$ and

$$c^{-d}k^2<\frac{(e^{-\delta}\widehat{W}_0)^{k_1 n}(e^{-\delta}\widehat{M}_{1,1})^{k_2|E|}}{(e^{\delta}(\widehat{\beta}+\widehat{\lambda}))^{k_1 n}(e^{\delta}\widehat{\beta})^{k_2|E|}2^{2kn}2^n n}.$$

This can be done, since $|E| = O(n)$. We will use below the fact that

$$\max_{\tau:T_{k,d}\to\{0,1\}} |\Pr(\boldsymbol{\sigma}_{k,d}(T_{k,d}) = \tau) - \mu_{k,d}(\tau)| < \frac{W_0^{k_1 n} M_{1,1}^{k_2|E|}}{(\beta+\lambda)^{k_1 n}\beta^{k_2|E|}2^{2kn}2^n n}, \tag{2.8}$$

which follows from Equations (2.6) and (2.7).

Let K be the positive real given by

$$K = \frac{2^n}{W_0^{k_1 n} M_{1,1}^{k_2|E|}}.$$

Note that there is a randomised approximation scheme for K whose running time is at most a polynomial in n and in the desired accuracy parameter, ε.

All that remains is to establish (2.5), which we do in the remainder of the proof. Let $T = \cup_{u\in V} T[u]$ be the set of terminals in $V(J)$. For every configuration $\tau : T \to \{0,1\}$, let $\mathrm{wt}(\tau) = \sum_{\sigma\in V(J)\to\{0,1\}:\sigma(T)=\tau} w_J(\sigma)$. The quantity $\mathrm{wt}(\tau)$ is the contribution to $Z_{\beta,\gamma,\lambda}(J)$ from configurations σ with $\sigma(T) = \tau$. Similarly, let $\mathrm{wt}'(\tau)$ be the contribution to $Z_{\beta,\gamma,\lambda}(J')$ from these configurations. Since $V(J') = V(J)$, we have $\mathrm{wt}'(\tau) = \sum_{\sigma\in V(J)\to\{0,1\}:\sigma(T)=\tau} w_{J'}(\sigma)$. Let $F(\tau)$ denote $\mathrm{wt}'(\tau)/\mathrm{wt}(\tau)$. Then

$$\frac{Z_{\beta,\gamma,\lambda}(J')}{Z_{\beta,\gamma,\lambda}(J)} = \frac{\sum_{\tau:T\to\{0,1\}} \mathrm{wt}'(\tau)}{Z_{\beta,\gamma,\lambda}(J)} = \frac{\sum_{\tau:T\to\{0,1\}} \mathrm{wt}(\tau)F(\tau)}{Z_{\beta,\gamma,\lambda}(J)} = \mathbb{E}[F(\boldsymbol{\sigma}_J(T))]. \tag{2.9}$$

We can write $F(\tau)$ in terms of β, γ, and λ:

$$F(\tau) = \left[\prod_{u\in V}\prod_{j=0}^{k_1-1}\left(\begin{pmatrix}\beta & 1\\ 1 & \gamma\end{pmatrix}\begin{pmatrix}1\\ \lambda\end{pmatrix}\right)_{\tau(T^1[u,j])}\right]\left[\prod_{(u_a,v_b)\in\mathcal{M},u<v}\prod_{j=0}^{k_2-1}\begin{pmatrix}\beta & 1\\ 1 & \gamma\end{pmatrix}_{\tau(T^0[u,ak_2+j]),\tau(T^0[v,bk_2+k_2-1-j])}\right] \tag{2.10}$$

We now define an "idealised" distribution on configurations assigning spins to the terminals. First, let $\tilde{\boldsymbol{\sigma}}$ be a random variable which is drawn uniformly from $V \to \{0,1\}$. Each realisation $\tilde{\sigma}$ of $\tilde{\boldsymbol{\sigma}}$ can be thought of as specifying, for every vertex $u \in V$, a phase $\tilde{\sigma}(u) \in \{0,1\}$ for the gadget $C[u]$. Conditioned on the realisation $\tilde{\boldsymbol{\sigma}} = \tilde{\sigma}$, the random variable $\hat{\boldsymbol{\sigma}} : T \to \{0,1\}$ is distributed as follows: for each $u \in V$, $\hat{\boldsymbol{\sigma}}(T[u])$ is chosen independently from the distribution $\mu_u^{\tilde{\sigma}(u)}$. Note that the (unconditioned) random variable $\hat{\boldsymbol{\sigma}}$ has the property that $\hat{\boldsymbol{\sigma}}(T[u])$ is distributed as μ_u, independently of all $\hat{\boldsymbol{\sigma}}(T[u'])$ for $u' \neq u$.

We wish to estimate $\mathbb{E}[F(\boldsymbol{\sigma}_J(T))]$, but the distribution of $\boldsymbol{\sigma}_J(T)$ is somewhat complicated. Instead, we will first estimate $\mathbb{E}[F(\hat{\boldsymbol{\sigma}})]$, and we will later use Proposition 2.5 to show that these two quantities are close. From the definition of $\hat{\boldsymbol{\sigma}}$, we have

$$\mathbb{E}[F(\hat{\boldsymbol{\sigma}})] = \frac{1}{2^n}\sum_{\tilde{\sigma}:V\to\{0,1\}} \mathbb{E}[F(\hat{\boldsymbol{\sigma}}) \mid \tilde{\boldsymbol{\sigma}} = \tilde{\sigma}].$$

Now we will argue that if $\tilde{\sigma}(u) = s$ then, for every parity-0 terminal t of $C[u]$, it is the case that $\Pr(\hat{\boldsymbol{\sigma}}(t) = j \mid \tilde{\boldsymbol{\sigma}} = \tilde{\sigma}) = P_{s,j}$. (To see this, consider the possible cases. If $s = 0$ then, from the definition of $\mu^0_{k,d}$, the probability that $\hat{\boldsymbol{\sigma}}(t) = 1$ is $p^{=}$, which is $P_{0,1}$, but the probability that $\hat{\boldsymbol{\sigma}}(t) = 0$ is $1 - p^{=} = P_{0,0}$. The situation is similar if $s = 1$.) On the other hand, similar reasoning shows that, for every parity-1 terminal t of $C[u]$, it is the case that $\Pr(\hat{\boldsymbol{\sigma}}(t) = j \mid \tilde{\boldsymbol{\sigma}} = \tilde{\sigma}) = P_{1\oplus s,j}$. Thus, we have

$$\mathbb{E}[F(\hat{\boldsymbol{\sigma}})] = \frac{1}{2^n} \sum_{\tilde{\sigma}:V\to\{0,1\}} \sum_{\tau:T\to\{0,1\}} F(\tau) \prod_{u\in V} \prod_{j=1}^{k} P_{\tilde{\sigma}(u),\tau(T^0[u,j])} P_{1\oplus\tilde{\sigma}(u),\tau(T^1[u,j])}.$$

Plugging in (2.10), the contribution of each $\tilde{\sigma}$ to the right-hand-side of the above equality is 2^{-n} multiplied by the product of the following terms:

$$\prod_{u\in V} \prod_{j=0}^{k_1-1} \sum_{s\in\{0,1\}} P_{1\oplus\tilde{\sigma}(u),s} \left(\begin{pmatrix} \beta & 1 \\ 1 & \gamma \end{pmatrix} \begin{pmatrix} 1 \\ \lambda \end{pmatrix} \right)_s,$$

$$\prod_{u\in V} \prod_{j=k_1}^{k-1} \sum_{s\in\{0,1\}} P_{1\oplus\tilde{\sigma}(u),s},$$

$$\prod_{(u,v)\in E} \prod_{j=0}^{k_2-1} \sum_{s\in\{0,1\}} \sum_{s'\in\{0,1\}} P_{\tilde{\sigma}(u),s} P_{\tilde{\sigma}(u),s'} \begin{pmatrix} \beta & 1 \\ 1 & \gamma \end{pmatrix}_{s,s'},$$

$$\prod_{u\in V} \prod_{j=3k_2}^{k-1} \sum_{s\in\{0,1\}} P_{\tilde{\sigma}(u),s}.$$

The second and fourth of these terms are equal to 1, and the first and third simplify using the matrices that we defined earlier, so we get

$$\mathbb{E}[F(\hat{\boldsymbol{\sigma}})] = \frac{1}{2^n} \sum_{\tilde{\sigma}:V\to\{0,1\}} \prod_{u\in V} W^{k_1}_{1\oplus\tilde{\sigma}(u)} \prod_{(u,v)\in E} {M_{\tilde{\sigma}(u),\tilde{\sigma}(v)}}^{k_2}.$$

Then, plugging in our notation from earlier, we have

$$\begin{aligned}
\mathbb{E}[F(\hat{\boldsymbol{\sigma}})] &= \frac{1}{2^n} \sum_{\tilde{\sigma}:V\to\{0,1\}} W_1^{k_1(n-\ell(\tilde{\sigma}))} W_0^{k_1\ell(\tilde{\sigma})} M_{0,0}^{k_2 b(\tilde{\sigma})} M_{0,1}^{k_2(|E|-b(\tilde{\sigma})-c(\tilde{\sigma}))} M_{1,1}^{k_2 c(\tilde{\sigma})} \\
&= \frac{W_0^{k_1 n} M_{0,1}^{k_2|E|}}{2^n} \sum_{\tilde{\sigma}:V\to\{0,1\}} \left(\frac{M_{0,0}}{M_{0,1}}\right)^{k_2 b(\tilde{\sigma})} \left(\frac{M_{1,1}}{M_{0,1}}\right)^{k_2 c(\tilde{\sigma})} \left(\frac{W_1}{W_0}\right)^{k_1(n-\ell(\tilde{\sigma}))}.
\end{aligned}$$

Replacing $\tilde{\sigma}(u)$ with $1 \oplus \tilde{\sigma}(u)$, we get

$$\mathbb{E}[F(\hat{\boldsymbol{\sigma}})] = \frac{W_0^{k_1 n} M_{0,1}^{k_2|E|}}{2^n} \sum_{\tilde{\sigma}:V\to\{0,1\}} \left(\frac{M_{1,1}}{M_{0,1}}\right)^{k_2 b(\tilde{\sigma})} \left(\frac{M_{0,0}}{M_{0,1}}\right)^{k_2 c(\tilde{\sigma})} \left(\frac{W_1}{W_0}\right)^{k_1\ell(\tilde{\sigma})}.$$

Since G is cubic, we can count the pairs $(v,(u,v)) \in V \times E$ with $\tilde{\sigma}(v) = 1$ in two ways to get $3\ell(\tilde{\sigma}) = 2c(\tilde{\sigma}) + (|E| - b(\tilde{\sigma}) - c(\tilde{\sigma}))$. So $b(\tilde{\sigma}) = c(\tilde{\sigma}) + |E| - 3\ell(\tilde{\sigma})$, which implies that

$$\begin{aligned}\mathbb{E}[F(\hat{\boldsymbol{\sigma}})] &= \frac{W_0^{k_1 n} M_{0,1}^{k_2|E|}}{2^n} \left(\frac{M_{1,1}}{M_{0,1}}\right)^{k_2|E|} \sum_{\sigma:V\to\{0,1\}} \left[\left(\frac{M_{0,0}M_{1,1}}{M_{0,1}^2}\right)^{k_2}\right]^{c(\sigma)} \left[\left(\frac{W_1}{W_0}\right)^{k_1} \left(\frac{M_{0,1}}{M_{1,1}}\right)^{3k_2}\right]^{\ell(\sigma)} \\ &= K^{-1} Z_{(1,\tilde{\gamma},\tilde{\lambda})}(G) \end{aligned} \tag{2.11}$$

Plugging (2.9) and (2.11) into (2.5), it remains to prove

$$e^{-1/4}\,\mathbb{E}[F(\hat{\boldsymbol{\sigma}})] \le \mathbb{E}[F(\boldsymbol{\sigma}_J(T))] \le e^{1/4}\,\mathbb{E}[F(\hat{\boldsymbol{\sigma}})]. \tag{2.12}$$

Let $\psi = |\,\mathbb{E}[F(\boldsymbol{\sigma}_J(T))] - \mathbb{E}[F(\hat{\boldsymbol{\sigma}})]|$. Now

$$\psi \le \left(\max_{\tau:T\to\{0,1\}} F(\tau)\right) \sum_{\tau:T\to\{0,1\}} \left|\Pr(\boldsymbol{\sigma}_J(T) = \tau) - \Pr(\hat{\boldsymbol{\sigma}} = \tau)\right|.$$

To emphasise that summation over $\tau : T \to \{0,1\}$ can be broken into summation over each restriction $\tau(T[u])$, we will write the summation index as $\forall u, \tau(T[u]) : T[u] \to \{0,1\}$. By (2.10), $F(\tau) \le (\beta+\lambda)^{k_1 n}\beta^{k_2|E|}$, so we can write

$$\begin{aligned}\psi &\le (\beta+\lambda)^{k_1 n}\beta^{k_2|E|} \sum_{\forall u,\tau(T[u]):T[u]\to\{0,1\}} \left|\prod_{u\in V}\Pr(\boldsymbol{\sigma}_J(T[u]) = \tau(T[u])) - \prod_{u\in V}\Pr(\hat{\boldsymbol{\sigma}}(T[u]) = \tau(T[u]))\right| \\ &= (\beta+\lambda)^{k_1 n}\beta^{k_2|E|} \sum_{\forall u,\tau(T[u]):T[u]\to\{0,1\}} \left|\prod_{u\in V}\Pr(\boldsymbol{\sigma}_J(T[u]) = \tau(T[u])) - \prod_{u\in V}\mu(\tau(T[u]))\right|.\end{aligned}$$

Using the inequality $|\prod a_u - \prod b_u| \le \sum |a_u - b_u|$ valid for values $0 \le a_u, b_u \le 1$, as in the proof of Proposition 2.5, we have

$$\psi \le (\beta+\lambda)^{k_1 n}\beta^{k_2|E|} 2^{2kn} n \max_{\tau:T_{k,d}\to\{0,1\}} \left|\Pr(\boldsymbol{\sigma}_{k,d} = \tau) - \mu_{k,d}(\tau)\right|.$$

Applying (2.8),

$$\psi \le (\beta+\lambda)^{k_1 n}\beta^{k_2|E|} 2^{2kn} n \left(\frac{W_0^{k_1 n} M_{1,1}^{k_2|E|}}{(\beta+\lambda)^{k_1 n}\beta^{k_2|E|}2^{2kn}2^n n}\right) = \frac{W_0^{k_1 n} M_{1,1}^{k_2|E|}}{2^n} = K^{-1}.$$

So to establish (2.12), we note that

$$1 - \frac{1}{Z_{1,\tilde{\gamma},\tilde{\lambda}}(G)} = 1 - \frac{1}{K\,\mathbb{E}[F(\hat{\boldsymbol{\sigma}})]} \le \frac{\mathbb{E}[F(\boldsymbol{\sigma}_J(T))]}{\mathbb{E}[F(\hat{\boldsymbol{\sigma}})]} \le 1 + \frac{1}{K\,\mathbb{E}[F(\hat{\boldsymbol{\sigma}})]} = 1 + \frac{1}{Z_{1,\tilde{\gamma},\tilde{\lambda}}(G)}.$$

The result follows from the extremely crude bound $Z_{1,\tilde{\gamma},\tilde{\lambda}}(G) \ge 8$. □

We can now prove our main theorem.

Theorem 2.1. Let β, γ and λ be efficiently-approximable reals satisfying (2.1). There is no FPRAS for $\mathsf{DegreeFourPlanarTwoSpin}(\beta, \gamma, \lambda)$ unless NP $\subseteq$ BPP.

Proof. We will give a randomised algorithm for $\mathsf{PlanarCubicIS}$, using an FPRAS for $\mathsf{DegreeFourPlanarTwoSpin}(\beta, \gamma, \lambda)$ as an oracle (and also using the given FPRASes for β, γ and λ).

After receiving an instance G and h, our algorithm uses Lemma 2.25 which provides planar graphs J and J' with maximum degree at most 4 and also some approximation schemes for the reals K, $\tilde{\gamma}$ and $\tilde{\lambda}$. With probability at least $1 - 1/15$, these satisfy $\tilde{\lambda} \geq 4^{|V(G)|}$ and $\tilde{\gamma} \leq \tilde{\lambda}^{-|V(G)|} < 1$ and Equation (2.5). The algorithm then makes four calls to approximation schemes, suitably powered so that each call fails with probability at most $1/15$. Thus, with probability at least $2/3$, Equation (2.5) is satisfied and all calls to the approximation schemes succeed. In that case, we will show how to determine (from the outputs of the approximation schemes) whether or not G has an independent set of size h.

Let $n = |V(G)|$. By Equation (2.5), we have

$$e^{-1/4} K \frac{Z_{\beta,\gamma,\lambda}(J')}{Z_{\beta,\gamma,\lambda}(J)} \leq Z_{1,\tilde{\gamma},\tilde{\lambda}}(G) \leq e^{1/4} K \frac{Z_{\beta,\gamma,\lambda}(J')}{Z_{\beta,\gamma,\lambda}(J)}.$$

Using the given approximation schemes for $Z_{\beta,\gamma,\lambda}(J')$, $Z_{\beta,\gamma,\lambda}(J)$ and K, each with accuracy parameter $1/12$ and failure probability $1/15$, we can compute a value $\hat{Z}$ which, with probability at least $1 - 3/15$, satisfies

$$e^{-1/4} \hat{Z} \leq K \frac{Z_{\beta,\gamma,\lambda}(J')}{Z_{\beta,\gamma,\lambda}(J)} \leq e^{1/4} \hat{Z},$$

so

$$e^{-1/2} \hat{Z} \leq Z_{1,\tilde{\gamma},\tilde{\lambda}}(G) \leq e^{1/2} \hat{Z}. \tag{2.13}$$

Using the given approximation scheme for $\tilde{\lambda}$ with accuracy parameter $1/2h$ and failure probability $1/15$, we can compute a value $\hat{\lambda}$ which, with probability at least $1 - 1/15$, satisfies $e^{-1/2h} \hat{\lambda} \leq \tilde{\lambda} \leq e^{1/2h} \hat{\lambda}$ so

$$e^{-1/2} \hat{\lambda}^h \leq \tilde{\lambda}^h \leq e^{1/2} \hat{\lambda}^h. \tag{2.14}$$

Suppose that all four calls to the approximation schemes succeed so that (2.13) and (2.14) hold. Recall that

$$Z_{1,\tilde{\gamma},\tilde{\lambda}}(G) = \sum_{\sigma : V(G) \to \{0,1\}} \tilde{\gamma}^{c(\sigma)} \tilde{\lambda}^{\ell(\sigma)}.$$

If G has an independent set of size h then $Z_{1,\tilde{\gamma},\tilde{\lambda}}(G) \geq \tilde{\lambda}^h$ so, plugging in (2.13) and (2.14),

$$\hat{Z} \geq e^{-1/2} Z_{1,\tilde{\gamma},\tilde{\lambda}}(G) \geq e^{-1/2} \tilde{\lambda}^h \geq e^{-1} \hat{\lambda}^h.$$

Also, if G has no independent set of size h, then $Z_{1,\tilde{\gamma},\tilde{\lambda}}(G) \leq 2^n \max(\tilde{\lambda}^{h-1}, \tilde{\lambda}^n \tilde{\gamma}) \leq 2^n \tilde{\lambda}^{h-1}$. So, plugging in (2.13) and (2.14) and our lower bound for $\tilde{\lambda}$,

$$\hat{Z} \leq e^{1/2} Z_{1,\tilde{\gamma},\tilde{\lambda}}(G) \leq e^{1/2} 2^n \tilde{\lambda}^{h-1} \leq e^{1/2} \frac{2^n \tilde{\lambda}^h}{4^n} = e^{1/2} 2^{-n} \tilde{\lambda}^h \leq e 2^{-n} \hat{\lambda}^h.$$

As long as $n \geq 3$, $e^{-1} > e2^{-n}$, so it is possible to determine from $\hat{Z}$ and $\hat{\lambda}$ whether or not G has an independent set of size h. $\square$

2.5 Approximating the log-partition function

We start with a preliminary lemma, which will help us to show that our approximation is sufficiently accurate.

Lemma 2.26. *Suppose that β, γ and λ are real numbers satisfying $\beta \geq 1 > \gamma \geq 0$ and $\lambda \geq 1$. Then, for every planar graph G, $Z_{\beta,\gamma,\lambda}(G) \geq (1+\lambda)^{|V(G)|/4}$.*

Proof. Let I be the largest colour class in a proper 4-colouring of G. Then I is an independent set of G of size at least $|V(G)|/4$. For every configuration $\sigma\colon V(G) \to \{0,1\}$ which assigns spin 0 to every vertex in $V(G) \setminus I$, $w_G(\sigma) \geq \lambda^{\ell(\sigma)}$. Thus $Z_{\beta,\gamma,\lambda}(G) \geq (1+\lambda)^{|I|}$. $\square$

Our approximation algorithm is inspired by Baker's approximation schemes for optimisation problems on planar graphs [2]. For a good explanation of her technique (which we use in our exposition here), see Borradailes's notes [12]. We will use the following notation (from [12]) to decompose a planar graph $G = (V, E)$ which is embedded in the plane. We first define the *level* of each vertex. Vertices on the boundary of the embedding have level 0. Then, for $i \in \{0, \ldots, n\}$, the vertices with level i are those that are on the boundary on the graph formed from G by deleting all vertices whose level is less than i. For a fixed parameter k, and for every $i \in \{0, \ldots, k-1\}$, let $V_i = \{v \in V \mid \text{The level of vertex } v \text{ is equal to } i \text{ modulo } k\}$. Let G_i be the graph $G - V_i$. By construction, G_i is $(k-1)$-*outerplanar.* Also, Bodlaender [7] had shown that every k-outerplanar graph has treewidth at most $3k-1$. Also, this tree decomposition is easy to compute. Using a data structure of Lipton and Tarjan [80], Baker shows that the levels of vertices can be computed in $O(|V|)$ time.

We can now prove Theorem 2.3.

Theorem 2.3. Suppose that β, γ and λ are efficiently approximable reals satisfying $\beta \geq 1 > \gamma \geq 0$ and $\lambda \geq 1$. There is a PRAS for $\mathsf{PlanarLogTwoSpin}(\beta, \gamma, \lambda)$.

Proof. Consider input $G = (V, E)$ with at least 3 vertices and an accuracy parameter $\varepsilon \in (0, 1)$. Let $n = |V|$ and $m = |E| \leq 3n$. Let β^+, β^-, λ^+, λ^-, γ^+ and γ^- be rational numbers (built into the algorithm) such that $\beta^+ \geq \beta \geq \beta^- \geq 1$, $1 > \gamma^+ \geq \gamma \geq \gamma^- \geq 0$, and $\lambda^+ \geq \lambda \geq \lambda^- \geq 1$. Let k be any integer satisfying

$$k \geq \frac{32 \log(2\lambda^+) + 96 \log(\beta^+)}{\varepsilon \log(1 + \lambda^-)}.$$

Then let

$$\delta = \frac{2n\log(2\lambda^+)}{k(n+m)}.$$

Using the given FPRASes for β, γ and λ, compute $\widehat{\beta}$, $\widehat{\gamma}$ and $\widehat{\lambda}$ satisfying $e^{-\delta}\beta \le \widehat{\beta} \le e^{\delta}\beta$, $e^{-\delta}\lambda \le \widehat{\lambda} \le e^{\delta}\lambda$ and $e^{-\delta}\gamma \le \widehat{\gamma} \le e^{\delta}\gamma$. As in the proof of Lemma 2.25, adjust the output of the FPRASes to ensure $\beta^+ \ge \widehat{\beta} \ge \beta^-$, $\gamma^+ \ge \widehat{\gamma} \ge \gamma^-$, and $\lambda^+ \ge \widehat{\lambda} \ge \lambda^-$.

The first step is to compute a value $\widehat{Z}$ satisfying

$$\widehat{Z} \le Z_{\widehat{\beta},\widehat{\gamma},\widehat{\lambda}}(G) \le (2\lambda^+)^{2n/k}(\beta^+)^{12n/k}\widehat{Z}. \tag{2.15}$$

This step is accomplished as follows.

1. Using Baker's algorithm, construct the graphs G_i for $i \in \{0,\dots,k-1\}$. Each of these has treewidth at most $3(k-1)-1$.

2. Choose $i \in \{0,\dots,k-1\}$ as follows. Let $\mathcal{I} = \{i \mid |V_i| \le 2n/k\}$. Note that $|\mathcal{I}| \ge k/2$. Now consider the $2m$ endpoints of edges in E. Choose $i \in \mathcal{I}$ so that V_i contains at most $(2m)/|\mathcal{I}|$ of these. Note that $|V_i| \le 2n/k$ and the number of edges with endpoints in V_i is at most $4m/k \le 12n/k$.

3. Use the algorithm of Yin and Zhang [104, Theorem 2.2] to compute $\widehat{Z} = Z_{\widehat{\beta},\widehat{\gamma},\widehat{\lambda}}(G_i)$. The running time of Yin and Zhang's algorithm is at most the product of a polynomial in n and an exponential function in the treewidth of G_i. In order to apply the algorithm, we first express the partition function $Z_{\widehat{\beta},\widehat{\gamma},\widehat{\lambda}}(G_i)$ as the solution to a Holant problem $\mathrm{Holant}(\mathcal{G},\mathcal{F})$ with regular symmetric $\mathcal{F}$. See [104] for definitions and details.

4. Equation (2.15) now follows by noting that

$$Z_{\widehat{\beta},\widehat{\gamma},\widehat{\lambda}}(G) = \sum_{\tau\colon V(G_i)\to\{0,1\}} w_{G_i}(\tau) \sum_{\tau'\colon V_i\to\{0,1\}} \widehat{\lambda}^{\ell(\tau')}\widehat{\beta}^{b(\tau,\tau')}\widehat{\gamma}^{c(\tau,\tau')},$$

   where $\ell(\tau')$ is the number of vertices $u \in V_i$ with $\tau'(u) = 1$ and $b(\tau,\tau')$ is the sum of the number of edges (u,v) with $u \in V(G_i)$ and $v \in V_i$ and $\tau(u) = \tau'(v) = 0$ and the number of edges (u,v) with $u \in V_i$ and $v \in V_i$ and $\tau'(u) = \tau'(v) = 0$ and $c(\tau,\tau')$ is defined similarly (with spin 1). Then $\sum_{\tau'\colon V_i\to\{0,1\}} \widehat{\lambda}^{\ell(\tau')}\widehat{\beta}^{b(\tau,\tau')}\widehat{\gamma}^{c(\tau,\tau')}$ is at least 1 (since τ' can assign spin 0 to every vertex in V_i) and it is at most $2^{2n/k}(\widehat{\lambda})^{2n/k}\widehat{\beta}^{12n/k}$.

To finish, note that

$$e^{-\delta(n+m)}Z_{\beta,\gamma,\lambda}(G) \le Z_{\widehat{\beta},\widehat{\gamma},\widehat{\lambda}}(G) \le e^{\delta(n+m)}Z_{\beta,\gamma,\lambda}(G),$$

so since

$$\delta(n+m) \le \frac{2n}{k}\log(2\lambda^+)$$

and (from Lemma 2.26)

$$\log Z_{\beta,\gamma,\lambda}(G) \geq (n/4)(1+\lambda^-),$$

and

$$(\varepsilon/2)(n/4)\log(1+\lambda^-) \geq \frac{4n}{k}\log(2\lambda^+) + \frac{12n}{k}\log(\beta^+),$$

$$\begin{aligned}
e^{-\varepsilon}\log(Z_{\beta,\gamma,\lambda}(G)) &\leq \log(Z_{\beta,\gamma,\lambda}(G))(1-\varepsilon/2) \\
&\leq \log Z_{\beta,\gamma,\lambda}(G) - \delta(n+m) - \frac{2n}{k}\log(2\lambda^+) - \frac{12n}{k}\log(\beta^+) \\
&\leq \log Z_{\widehat{\beta},\widehat{\gamma},\widehat{\lambda}}(G) - \frac{2n}{k}\log(2\lambda^+) - \frac{12n}{k}\log(\beta^+) \\
&\leq \log \widehat{Z}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
\log \widehat{Z} &\leq \log Z_{\widehat{\beta},\widehat{\gamma},\widehat{\lambda}}(G) \\
&\leq \delta(n+m) + \log Z_{\beta,\gamma,\lambda}(G) \\
&\leq \frac{2n}{k}\log(2\lambda^+) + \log Z_{\beta,\gamma,\lambda}(G) \\
&\leq (1+\varepsilon/2)\log Z_{\beta,\gamma,\lambda}(G) \\
&\leq e^{\varepsilon}\log Z_{\beta,\gamma,\lambda}(G).
\end{aligned}$$

□

Chapter 3

Approximating Holant problems by winding

(This chapter is a revised version of [84], without the introduction to FPRASes and #CSPs.)

We give an FPRAS for Holant problems with parity constraints and not-all-equal constraints, a generalisation of the problem of counting sink-free-orientations. The approach combines a sampler for near-assignments of "windable" functions – using the cycle-unwinding canonical paths technique of Jerrum and Sinclair – with a bound on the weight of near-assignments. The proof generalises to a larger class of Holant problems; we characterise this class and show that it cannot be extended by expressibility reductions.

We then ask whether windability is equivalent to expressibility by matchings circuits (an analogue of matchgates), and give a positive answer for functions of arity three.

3.1 Introduction

In this chapter we will show that the following problem has an FPRAS.

Name. #$\mathsf{ParityNAE}$
Instance. A multigraph G in which each vertex is labelled Even, Odd, or NAE
Output. The number of subsets $F \subseteq E(G)$ such that:

- each Even vertex has an even number of incident edges in F
- each Odd vertex has an odd number of incident edges in F
- each NAE vertex has at least one incident edge in F and at least one incident edge in $E(G) \setminus F$

Theorem 3.1. *There is an FPRAS for* #$\mathsf{ParityNAE}$.

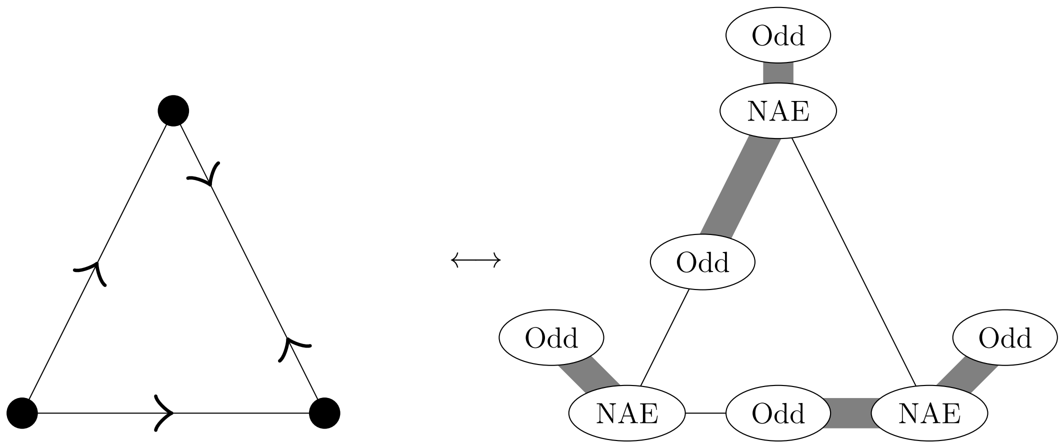

FIGURE 3.1: Reduction from #SFO to #ParityNAE. The edge with two arrows is a skew edge. A sink-free orientation is illustrated with the corresponding set F draw in thick grey.

3.1.1 Relationships with other counting problems

Consider a graph G. An *orientation* of G is a choice of orientation of each edge: exactly one of the endpoints of each edge is chosen to be the head of the edge, while the other endpoint is the tail. An orientation is sink-free if each vertex is a tail of some edge. We can also allow some edges of G to be designated "skew edges". An orientation of G then consists of an orientation of each ordinary edge, and a choice of which skew edges will point outwards. If a skew edge points outwards, both its endpoints are its heads, and otherwise both its endpoints are its tails. The problem #SFO takes as input a multigraph G where each edge is designated ordinary or skew, and outputs the number of sink-free orientations of G.

Bubley and Dyer studied #SFO and gave an FPRAS [15]. They showed as a corollary that there is an FPRAS for counting solutions to a formula in conjunctive normal form in which every variable appears at most twice, which they showed is a #P-hard problem. The first part of their argument was a standard reduction to sampling - finding a fully polynomial almost uniform sampler (FPAUS) for sink-free orientations. Then, they constructed a Markov chain that converges to the uniform distribution on sink-free orientations, and bounded its mixing time using a two-stage path coupling argument. Cohn, Pemantle and Propp later gave an exact sampler (for a simpler problem where skew edges are not allowed) with $O(|V|\cdot|E|)$ mean running time, using a kind of rejection sampling [38].

A simple reduction from #SFO to #ParityNAE is illustrated in Figure 3.1, showing that #ParityNAE generalises the problem of counting sink-free orientations in a graph (while also allowing parity constraints). Given an instance G of #SFO, label all the vertices NAE, subdivide each non-skew edge uv, label the new vertices (which we will refer to as "m_{uv}") Odd, then attach a degree-one Odd vertex to each NAE vertex. This gives an instance G' of #ParityNAE. For all orientations O of G define a set $F_O \subseteq E(G')$ by taking all edges attached to degree-one Odd vertices and all edges corresponding to heads: for non-skew edges uv take $um_{uv} \in F$ if and only if u is the head of uv, and for skew edges uv take $uv \in F$ if and only if uv is oriented outwards. Each degree-two Odd vertex in G' has exactly one incident edge in F_O, and each NAE vertex in G' has at least one incident edge in F_O, and if O is sink-free then each NAE vertex in G' has at least one incident edge not in F_O. Furthermore, any $F \subseteq E(G')$ satisfying these conditions is F_O for some sink-free orientation O. The function $O \mapsto F_O$ therefore gives a bijection from sink-free orientations of G to the set of subsets of $E(G')$ that get counted by #$\mathsf{ParityNAE}$.

#$\mathsf{ParityNAE}$, at least when restricted to bounded-degree graphs, is a type of Boolean Holant problem. In this discussion we will take the codomain to be the set of complex numbers; functions $\{0,1\}^k \to \mathbb{C}$ are called signatures in this context. But afterwards we will restrict to non-negative rational-valued functions.

For all positive integers k define $\mathbf{Even}_k, \mathbf{Odd}_k, \mathbf{NAE}_k \colon \{0,1\}^k \to \{0,1\}$ by setting $\mathbf{Even}_k(x_1, \ldots, x_k)$ to be 1 if and only if $x_1 + \cdots + x_k$ is even, setting $\mathbf{Odd}_k(x_1, \ldots, x_k)$ to be 1 if and only if $x_1 + \cdots + x_k$ is odd, and setting $\mathbf{NAE}_k(x_1, \ldots, x_k)$ to be 1 if and only if $1 \le x_1 + \cdots + x_k \le k-1$. The restriction of #$\mathsf{ParityNAE}$ to graphs of maximum degree at most d is then equivalent to $\mathsf{Holant}(\mathcal{F}_d)$ where

$$\mathcal{F}_d = \{\mathbf{Even}_1, \mathbf{Odd}_1, \mathbf{NAE}_1, \ldots, \mathbf{Even}_d, \mathbf{Odd}_d, \mathbf{NAE}_d\}.$$

By Theorem 3.1, this problem has an FPRAS for each d.

We now recall the relationships between #CSPs and Holant problems in the Hadamard basis as discussed in [66]. Note that while these equivalences are usually stated in the context of exact evaluation, the reductions just involve preprocessing the input, and so also apply in the context of approximate counting. Firstly, equality constraints can be used to break the read-twice restriction: if EQ_3 is in $\mathcal{F}$ then $\mathsf{Holant}(\mathcal{F})$ is equivalent to #$\mathsf{CSP}(\mathcal{F})$ [29, Proposition 5]. Secondly, let $\widehat{F} \colon \{0,1\}^k \to \mathbb{C}$ denote the Hadamard transform, defined by

$$\widehat{F}(x_1, \ldots, x_k) = 2^{-k/2} \sum_{\mathbf{y} \in \{0,1\}^k} F(y_1, \ldots, y_k)(-1)^{x_1 y_1 + \cdots + x_k y_k}.$$

$\mathsf{Holant}(\mathcal{F})$ is always equivalent to $\mathsf{Holant}(\{\widehat{F} \mid F \in \mathcal{F}\})$; see [29, Proposition 1] or [82]. Also, $\widehat{\widehat{F}} = F$ for any F. So if $\mathcal{F}$ contains $\widehat{\mathrm{EQ}_3}$, then $\mathsf{Holant}(\mathcal{F})$ is equivalent to #$\mathsf{CSP}(\{\widehat{F} \mid F \in \mathcal{F}\})$. But $\widehat{\mathrm{EQ}_3}$ is just $\mathbf{Even}_3$ multiplied by a factor of $\sqrt{2}$ (which can be easily accounted for).

Taking $\mathcal{F}$ to be the set $\mathcal{F}_d$ defined above, with $d \ge 3$, we find that the restriction of #$\mathsf{ParityNAE}$ to instances of degree at most d is equivalent to #$\mathsf{CSP}(\{\widehat{F} \mid F \in \mathcal{F}_d\})$. By Theorem 3.1 this problem has an FPRAS for each d. In this sense, #$\mathsf{ParityNAE}$ generalises #SFO to a #CSP. Note that $\widehat{\mathbf{Odd}_1}(0) = 1/\sqrt{2}$ and $\widehat{\mathbf{Odd}_1}(0) = -1/\sqrt{2}$. So we get a class of FPRASes for #CSPs using functions with mixed signs.

3.1.2 Techniques

Like Bubley and Dyer we will use Markov chains, but to bound the mixing time we will instead apply the canonical paths technique. More precisely, we will use a multicommodity flow with cycle-unwinding as used by Jerrum and Sinclair [70]. They proved the following relevant result: for any polynomial p we can sample efficiently from the uniform distribution of perfect matchings, in graphs G satisfying

$$\frac{\text{number of matchings of order } \frac{1}{2}|V(G)|-1}{\text{number of matchings of order } \frac{1}{2}|V(G)|} \leq p(|V(G)|). \tag{3.1}$$

Recall that a *matching* of a graph is a set of edges not sharing any vertices, and a matching is *perfect* if it has order $|V(G)|/2$. A perfect matching is a satisfying assignment to a certain system of constraints: each edge is either IN or OUT, and every variable enforces a perfect matchings constraint, that exactly one of its incident edges is IN. From this perspective a natural question is: what weight-functions can we use instead of perfect matchings constraints? We show that Jerrum and Sinclair's result generalises in a certain sense to *windable* functions, defined as follows.

Definition 3.2. For any finite set J and any configuration $\mathbf{x} \in \{0,1\}^J$ define $\mathcal{M}'_{\mathbf{x}}$ to be the set of partitions of $\{i \mid x_i = 1\}$ into pairs and singletons. A function $F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ is **windable** if there exist values $B(\mathbf{x},\mathbf{y},M) \geq 0$ for all $\mathbf{x},\mathbf{y} \in \{0,1\}^J$ and all $M \in \mathcal{M}'_{\mathbf{x}\oplus\mathbf{y}}$ satisfying:

1. $F(\mathbf{x})F(\mathbf{y}) = \sum_{M \in \mathcal{M}'_{\mathbf{x}\oplus\mathbf{y}}} B(\mathbf{x},\mathbf{y},M)$ for all $\mathbf{x},\mathbf{y} \in \{0,1\}^J$, and
2. $B(\mathbf{x},\mathbf{y},M) = B(\mathbf{x}\oplus\mathbf{S},\mathbf{y}\oplus\mathbf{S},M)$ for all $\mathbf{x},\mathbf{y} \in \{0,1\}^J$ and all $S \in M \in \mathcal{M}'_{\mathbf{x}\oplus\mathbf{y}}$.

Here $\mathbf{x}\oplus\mathbf{S}$ denotes the vector obtained by changing x_i to $1-x_i$ for the one or two elements i in S.

The next question is: what kinds of constraints guarantee a bound like (3.1)? We give one answer: *strictly terraced* functions.

Definition 3.3. A function $F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ is **strictly terraced** if

$$F(\mathbf{x}) = 0 \implies F(\mathbf{x}\oplus\mathbf{e}_i) = F(\mathbf{x}\oplus\mathbf{e}_j) \qquad \text{for all } \mathbf{x} \in \{0,1\}^J \text{ and all } i,j \in J.$$

Here $\mathbf{x}\oplus\mathbf{e}_i$ denotes the vector obtained by changing x_i to $1-x_i$.

We will discuss these definitions more throughout the chapter. Using properties of these classes, we will establish Theorem 3.1. A feature of the techniques is that they cannot be extended by expressibility reductions. We make this precise in the following theorem. *Circuits* are a natural type of gadget for Holant problem reductions which we define formally in Section 3.2.1. If $\mathcal{G}$ consists of weight-functions of circuits using functions in $\mathcal{G}'$, then $\mathsf{Holant}(\mathcal{G}) \leq_{\mathrm{AP}} \mathsf{Holant}(\mathcal{G}')$. So one might hope to find an FPRAS for a problem $\mathsf{Holant}(\mathcal{G})$ by simulating each constraint by a circuit using windable strictly terraced functions, then applying the techniques of this chapter to the resulting problem. The first bullet point of the following theorem shows that this approach does not give any extra tractable problems.

Theorem 3.4. *Let $\mathcal{F}$ be the class of strictly terraced windable functions. Then*

- *$\mathcal{F}$ is closed under taking weight-functions of connected [1] circuits*
- *$\mathcal{F}$ contains $\mathbf{Even}_k$, $\mathbf{Odd}_k$, and $\mathbf{NAE}_k$ for all $k \geq 1$*
- *for all finite subsets $\mathcal{F}' \subset \mathcal{F}$ there is an FPRAS for* $\mathsf{Holant}(\mathcal{F}')$

The reason to take $\mathcal{F}'$ to be finite is to make sense of the computational problem $\mathsf{Holant}(\mathcal{F}')$. As in Theorem 3.1, if one is careful about how the input is specified, it is also possible to allow infinite $\mathcal{F}'$ in some cases.

3.1.3 Matching circuits

In Section 3.7 we will consider a natural type of gadget for reducing Holant problems to #PM, the problem of counting the number of perfect matchings in a graph.

As mentioned in Section 1.2.1, #PM is #P-complete. This suggests that there is no efficient exact algorithm, leaving the question of whether there is an approximation algorithm. A major result in this direction is that there is an FPRAS for #PM restricted to bipartite graphs [72]. Our study of matching circuits is an attempt to identify which Holant problems reduce to #PM in the sense of expressibility.

Take a clique of order four K_4, and attach an outgoing edge d_i at the i'th vertex for each $1 \leq i \leq 4$. For each of the sixteen possible subsets $M \subseteq \{d_1, d_2, d_3, d_4\}$ of the outgoing edges, we can count the number $F(M)$ of ways to add internal edges of K_4 to the edges in M to obtain a perfect matching. Because K_4 by itself has 3 perfect matchings, we have $F(\emptyset) = 3$, while $F(\{d_1\}) = 0$ and $F(\{d_1, d_2, d_3, d_4\}) = 1$.

We will say that a function $F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ has a matchings circuit if there is a similar graph fragment, with outgoing edges J, and such that $F(\mathbf{x})$ is the number of perfect matchings containing the outgoing edges $\{i \in J \mid x_i = 1\}$. In this chapter, we will think of Holant instances as a graph whose vertices are the constraints and whose edges are the variables. Substituting each vertex of a $\mathsf{Holant}(\{F\})$ instance by the graph fragment gives a reduction from $\mathsf{Holant}(\{F\})$ to #PM. As a useful generalisation, our definition of matchings circuits will also allow non-negative edge-weights and a "fugacity" at each vertex. These parameters were used in [70] to study permanents and monomer-dimer systems. The extra flexibility turns out not to increase the computational complexity beyond #PM (even though #PM does not involve edge-weights or fugacities). We show:

Theorem 3.5. *If $\mathcal{F}$ is a finite set of weight-functions that have matchings circuits, then* $\mathsf{Holant}(\mathcal{F}) \leq_{\mathrm{AP}}$ #PM.

[1] The restriction to *connected* circuits is a technical convenience to keep $\mathcal{F}$ in Theorem 3.4 as simple as possible – disconnected circuits are only useful for expressing functions that are decomposable in the sense of Section 5.1.6.

The main result is the following theorem.

Theorem 3.6. *Let $F\colon \{0,1\}^3 \to \mathbb{Q}_{\geq 0}$. The following are equivalent:*

1. *F is windable*

2. *For all $x_1, x_2, x_3 \in \{0,1\}$ we have*

$$\begin{aligned} & F(x_1,x_2,x_3)F(1-x_1,1-x_2,1-x_3) \\ \leq\ & F(x_1,x_2,1-x_3)F(1-x_1,1-x_2,x_3) \\ +\ & F(x_1,1-x_2,x_3)F(1-x_1,x_2,1-x_3) \\ +\ & F(x_1,1-x_2,1-x_3)F(1-x_1,x_2,x_3) \end{aligned}$$

3. *F has a matchings circuit*

Theorem 3.6 gives a class of problems that reduce to counting perfect matchings. For example, the Holant problem allowing only the relation $\{(0,0,0),(1,0,0),(0,1,0),(1,0,1),(0,1,1)\}$ reduces to $\#\mathsf{PM}$, but is not known to have an FPRAS.

3.1.4 Related work

A matroid is *sbo (strongly basis orderable)* [42] if for all bases A and B there is a bijection $\pi\colon A \setminus B \to B \setminus A$ such that for all $X \subseteq A \setminus B$ the set $(A \cup \pi(X)) \setminus X$ is a basis. Bouchet and Cunningham generalised the sbo property as *linkability* for the class of even delta-matroids, and showed that this class is closed under an analogue of circuits [13]. These conditions are just windability over the two-element Boolean semiring ($\mathbb{B} = \{0,1\}, \max, \min$), for the set of bases when considered as a function $\{0,1\}^J \to \mathbb{B}$, by taking the characteristic vector of characteristic vectors of bases. Gambin used the sbo property to approximately count the number of bases in certain matroids [57].

Valiant [98] introduced *matchgates* and *matchcircuits*, which are similar to matchings circuits but give efficient exact algorithms. Matchcircuits can be understood as planar graphs with edge-weights, with no restriction to non-negative numbers. Cai and Choudhary characterised the expressibility of matchgates [28]. The name "matchings circuits" used in this chapter is meant to suggest a version of matchgates.

As discussed in Section 1.5.5, the focus on (the negative side of) expressibility for approximate counting problems appears in [22], where logsupermodular functions are shown not to express non-logsupermodular functions in the context of $\#\mathsf{CSP}$s.

Yamakami [103] has given partial classifications for classes of Holant problems, and we will also study these problems in Chapter 5. The bulk of these results deal with intractability: reductions from a named problem such as $\#\mathsf{SAT}$ to a given Holant problem. The focus of the current chapter is on tractability: either in the absolute sense of an FPRAS, or by reductions to $\#\mathsf{PM}$.

The equivalence between $\#\mathsf{PM}$ and certain points of the Tutte polynomial mentioned in Example 1.2 can be seen as an example of a $\#\mathsf{CSP}$ using weight functions with mixed signs studied in the context of approximate counting. By the proof of [59, Lemma 7], the following problems are equivalent in the sense of approximate counting[2] , for any fixed $y < -1$.

- #PM
- #CSP($\{B_y\}$) where $B_y\colon \{0,1\}^2 \to \mathbb{Q}$ is defined by $B_y(0,0) = B_y(1,1) = y$ and $B_y(0,1) = B_y(1,0) = 1$
- evaluating the Tutte polynomial at the point (x, y) where $(x-1)(y-1) = 2$

3.1.5 Outline

In Section 3.3, we adapt the conductance argument of Jerrum and Sinclair to "even-windable" functions, which are a slightly simpler version of windable functions. We study windable functions in Section 3.4. We study strictly terraced functions in Section 3.5. In Section 3.6 we establish Theorem 3.1 and Theorem 3.4. Finally, in Section 3.7 we discuss matchings circuits and establish Theorem 3.5 and Theorem 3.6.

3.2 Preliminaries

In this chapter, we will use bold face to distinguish between sets $S \subseteq J$ and the characteristic vector $\mathbf{S}$. Similarly the bold version of a relation $R \subseteq \{0,1\}^J$ is the corresponding zero-one-valued weight-function.

We will sometimes allow indexing sets to be partially enumerated in a certain way. This is for notational power: the enumerated indices are easy to refer to explicitly, while the unenumerated indices are easy to fix. For all positive integers k and all finite sets J, when $k + J$ is used as an indexing set it means the disjoint union of $\{1, \dots, k\}$ and J. Elements of $\{0,1\}^{k+J}$ will be denoted by $(x_1, \dots, x_k; \mathbf{y})$ where $x_1, \dots, x_k \in \{0,1\}$ and $\mathbf{y} \in \{0,1\}^J$.

The **distance** $|\{i \in J \mid x_i \neq y_i\}|$ between two configurations $\mathbf{x}, \mathbf{y} \in \{0,1\}^J$ will be denoted $\mathrm{d}(\mathbf{x}, \mathbf{y})$. We say F is **even**[3] if $\mathrm{d}(\mathbf{x}, \mathbf{y})$ is even for all $\mathbf{x}, \mathbf{y}$ with $F(\mathbf{x}), F(\mathbf{y}) > 0$.

For all $F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ and $\mathbf{y} \in \{0,1\}^J$ define the **flip of** F **by** $\mathbf{y}$ to be the weight-function $F'\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ defined by $F'(\mathbf{x}) = F(\mathbf{x} \oplus \mathbf{y})$ for all $\mathbf{x} \in \{0,1\}^J$. For all $i \in J$ define $\mathbf{e}_i \in \{0,1\}^J$ (where J is implicit) to be the characteristic vector of $\{i\}$.

[2] For a generalisation of the definition of FPRAS allowing approximate negative values.

[3] This may be confusing terminology - an even function can be non-zero on vectors of *odd* Hamming weight. But it is a common definition for delta-matroids.

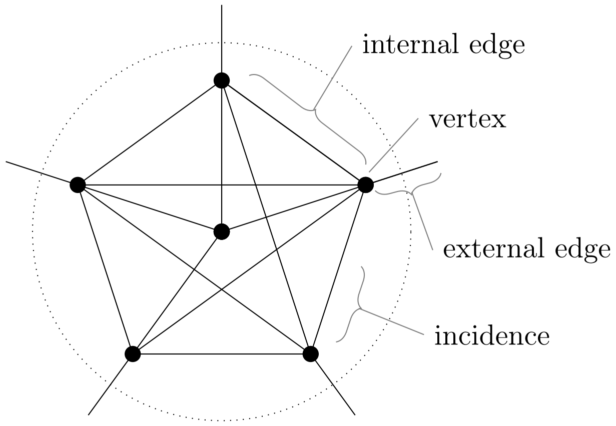

FIGURE 3.2: Terminology for graph fragments.

For all finite sets J we will use the relations

$$\begin{aligned}
\mathrm{Even}_J &= \{\mathbf{x} \in \{0,1\}^J \mid \sum_{i \in J} x_i \text{ is even}\} \\
\mathrm{Odd}_J &= \{\mathbf{x} \in \{0,1\}^J \mid \sum_{i \in J} x_i \text{ is odd}\} \\
\mathrm{NAE}_J &= \{\mathbf{x} \in \{0,1\}^J \mid 1 \le \sum_{i \in J} x_i \le |J| - 1\} \\
\mathrm{EvenNAE}_J &= \mathrm{Even}_J \cap \mathrm{NAE}_J
\end{aligned}$$

Even_J and Odd_J are **parity relations**. The last relation $\mathrm{EvenNAE}_J$ is only used for calculations (and only with $|J|$ even).

3.2.1 Circuits

In this chapter, circuits are a type of graph equipped with weight-functions at each vertex, and allowing external edges. A little care is needed to allow self-loops and asymmetric weight-functions.

A **graph fragment** G is specified by:

- a set J^G whose elements are called incidences
- a set V^G of vertices, and sets J_v^G, $v \in V^G$, that partition J^G
- a set $A^G \subseteq J^G$ whose elements are called external edges
- a partition E^G of $J^G \setminus A^G$ into pairs called internal edges

See Figure 3.2.

A **circuit** φ is graph fragment equipped with a **constraint** $F_v^\varphi \colon \{0,1\}^{J_v^\varphi} \to \mathbb{Q}_{\geq 0}$ for each vertex v. We can also use a relation $R \subseteq \{0,1\}^{J_v^\varphi}$ as a constraint by taking $F_v^\varphi = \mathbf{R}$. We will drop the superscript φ where there is only one graph or circuit in context.

G is **closed** if it has no external edges. Standard graph-theoretic terminology extends to graph fragments. In particular we will refer to connected graph fragments. An *edge* is either an internal edge or an external edge. A vertex v and an internal edge e are *incident* if J_v intersects e. If an internal edge e is uniquely identified by the vertices u, v it is incident to, we will denote e by uv.

Given a circuit φ, for any configuration $\mathbf{x}$ of J,

- $\mathbf{x}$ is an **assignment** (with respect to E) if $x_i = x_j$ for all $\{i, j\} \in E$.
- $\mathbf{x}|_{J_v}$ denotes the restriction of $\mathbf{x}$ to J_v.
- The **weight** of $\mathbf{x}$ is $\mathrm{wt}_\varphi(\mathbf{x}) = \prod_{v \in V} F_v(\mathbf{x}|_{J_v})$.

The **weight-function of** φ is the function $[\![\varphi]\!] : \{0,1\}^A \to \mathbb{Q}_{\geq 0}$ defined by

$$[\![\varphi]\!](\mathbf{x}) = \sum_{\mathbf{x}'} \mathrm{wt}_\varphi(\mathbf{x}') \qquad (\mathbf{x} \in \{0,1\}^A)$$

where the sum is over extensions of $\mathbf{x}$ to assignments $\mathbf{x}' \colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ with respect to E. If a weight-function F is equal to $[\![\varphi]\!]$, we will say that F **has** the circuit φ.

Another way to think of a circuit is as a "read-twice pps-formula", a special case of the pps-formulas of [22]. For example, consider an equation

$$F(x) = \sum_{y=0}^{1} \sum_{z=0}^{1} G(x, y) G(y, z) H(z) \qquad (x \in \{0, 1\}).$$

Note how on the right-hand-side, each bound (summed) variable appears exactly twice, and each free (unsummed) variable appears exactly once. Any equation of this form defines a circuit in a natural way: incidences correspond to the variable occurrences x, y, y, z, z; vertices correspond to terms $G(x, y), G(y, z), H(z)$; the sets J_v are scopes for each term; external edges correspond to free variables; and internal edges correspond to summed variables.

For any partition E of a finite set J into pairs, for all non-negative integers k, a k**-assignment** with respect to E is a configuration $\mathbf{x}$ of J such that $x_i = x_j$ for all but exactly k pairs $\{i, j\} \in E$. So an assignment is a 0-assignment. For all closed circuits φ and all integers $k \geq 0$ define

$$Z_k(\varphi) = \sum_{k\text{-assignments } \mathbf{x}} \mathrm{wt}_\varphi(\mathbf{x}).$$

So $Z_0(\varphi)$ is just $[\![\varphi]\!]$ (evaluated on the empty configuration).

In this chapter, we use an alternate definition of Holant problems:

Name. $\mathsf{Holant}(\mathcal{F})$
Instance. A closed circuit φ using copies of weight-functions in $\mathcal{F}$
Output. $[\![\varphi]\!]$

This is equivalent to the definition given in Section 1.3 of Holant problems as a read-twice #CSP, because each edge in a closed circuit is used exactly twice.

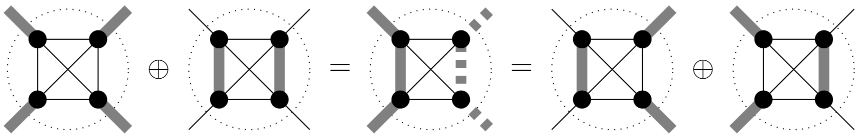

FIGURE 3.3: An example of constructing perfect matchings by symmetric differences. From left to right, M, M', $M \triangle M'$ (with P drawn in thick solid grey), $M \triangle P$, and $M' \triangle P$.

By substituting circuits, if F has a circuit using copies of weight-functions from a finite set $\mathcal{F}$, then $\mathsf{Holant}(\mathcal{F} \cup \{F\}) \leq_{\mathrm{AP}} \mathsf{Holant}(\mathcal{F})$. This justifies the focus on expressibility in this chapter.

3.3 Even-windable functions

3.3.1 Idea

Windability is an abstraction of a property of the distribution of perfect matchings in a graph with external edges. We will illustrate the idea briefly by the arity 4 case, where windability is already used implicitly in [70]. But higher-arity conditions are important for showing that windability is preserved by circuits.

Consider a graph G with four external edges e_1, e_2, e_3, e_4. For all x_1, x_2, x_3, x_4, let $F(x_1, x_2, x_3, x_4)$ be the number of perfect matchings in G that include the outgoing edges $\{e_i \mid x_i = 1\}$. So $F(0,0,0,0)F(1,1,1,1)$ is the number of pairs of perfect matchings (M, M') such that M includes all the external edges and M' includes none. But for any such pair (M, M'), the symmetric difference $M \triangle M'$ consists of cycles and paths, and the path starting at e_1 ends at either e_2, e_3, or e_4, depending on the choice of (M, M'). Thus $F(0,0,0,0)F(1,1,1,1)$ splits into three terms. Denote these numbers by $B((0,0,0,0),(1,1,1,1),M)$ where M is a partition of $\{1,2,3,4\}$ into pairs: either $\{\{1,2\},\{3,4\}\}$ or $\{\{1,3\},\{2,4\}\}$ or $\{\{1,4\},\{2,3\}\}$. For the clique example, Figure 3.3 (after numbering the external edges arbitrarily), we find that the value of $B((0,0,0,0),(1,1,1,1),\{\{1,2\},\{3,4\}\})$ (for example) is 1, because there is precisely one perfect matching of the clique giving a path from the first and second external edges. In a similar way we can define $B((1,1,0,0),(0,0,1,1),M)$.

When $M \triangle M'$ contains a path P from e_1 to e_2, the sets $M \triangle P$ and $M' \triangle P$ are also perfect matchings - see Figure 3.3. The only external edges in $M \triangle P$ are e_3 and e_4, while the only external edges in $M' \triangle P$ are e_1 and e_2. Thus $B((0,0,0,0),(1,1,1,1),\{\{1,2\},\{3,4\}\})$ equals $B((1,1,0,0),(0,0,1,1),\{\{1,2\},\{3,4\}\}$.

In this section, for simplicity, we will consider only even functions.

3.3.2 Definition

For any configuration $\mathbf{x} \in \{0,1\}^J$ define $\mathcal{M}_{\mathbf{x}}$ to be the set of partitions of $\{i \in J \mid x_i = 1\}$ into pairs. In particular, if $\sum_{i \in J} x_i$ is odd then $\mathcal{M}_{\mathbf{x}} = \emptyset$.

A function $F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ is **even-windable** (with witness B) if there exist values $B(\mathbf{x}, \mathbf{y}, M) \geq 0$ for all $\mathbf{x}, \mathbf{y} \in \{0,1\}^J$ and all $M \in \mathcal{M}_{\mathbf{x}\oplus\mathbf{y}}$, i.e. all partitions M of the set $\{i \in J \mid x_i \neq y_i\}$ into pairs, satisfying:

EW1. $F(\mathbf{x})F(\mathbf{y}) = \sum_{M\in\mathcal{M}_{\mathbf{x}\oplus\mathbf{y}}} B(\mathbf{x}, \mathbf{y}, M)$ for all $\mathbf{x}, \mathbf{y} \in \{0,1\}^J$, and

EW2. $B(\mathbf{x}, \mathbf{y}, M) = B(\mathbf{x} \oplus \mathbf{S}, \mathbf{y} \oplus \mathbf{S}, M)$ for all $\mathbf{x}, \mathbf{y} \in \{0,1\}^J$ and all $S \in M \in \mathcal{M}_{\mathbf{x}\oplus\mathbf{y}}$.

Note that in the second condition, S is a pair $\{i, j\}$ in M: we are swapping the values of x_i and y_i, and swapping the values of x_j and y_j. By swapping a sequence of pairs, EW2 is equivalent to

EW2'. $B(\mathbf{x}, \mathbf{y}, M) = B(\mathbf{x} \oplus \mathbf{S}_1 \oplus \cdots \oplus \mathbf{S}_k, \mathbf{y} \oplus \mathbf{S}_1 \oplus \cdots \oplus \mathbf{S}_k, M)$ for all $\mathbf{x}, \mathbf{y} \in \{0,1\}^J$ and all $S_1, \ldots, S_k \in M \in \mathcal{M}_{\mathbf{x}\oplus\mathbf{y}}$.

3.3.3 2-decompositions

Using pinnings, the even-windability conditions can be stated in a form that is sometimes easier to check. A function $H\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ has a **2-decomposition** if there are values $D(\mathbf{x}, M) \geq 0$, where $\mathbf{x}$ ranges over $\{0,1\}^J$ and M ranges over partitions of J into pairs, such that:

1. $H(\mathbf{x}) = \sum_M D(\mathbf{x}, M)$ for all $\mathbf{x}$, where the sum is over partitions of J into pairs, and
2. $D(\mathbf{x}, M) = D(\mathbf{x} \oplus \mathbf{S}, M)$ for all $\mathbf{x}, M$ and all $S \in M$.

In particular if $|J|$ is odd then the first condition forces H to be identically zero.

A function F is even-windable if and only if for all pinnings G of F the function $G\overline{G}$ has a 2-decomposition. For the forwards direction, given a witness B that F is even-windable, for each $I \subseteq J$ and each $\mathbf{p} \in \{0,1\}^{J\setminus I}$ define

$$D_{\mathbf{p}}(\mathbf{x}, M) = B((\mathbf{x}, \mathbf{p}), (\overline{\mathbf{x}}, \mathbf{p}), M) \quad (\mathbf{x} \in \{0,1\}^I, M \in \mathcal{M}_{\mathbf{I}})$$

to obtain a 2-decomposition $D_{\mathbf{p}}$ of $G\overline{G}$ where $G\colon \{0,1\}^I \to \mathbb{Q}_{\geq 0}$ is the pinning of F by $\mathbf{p}$. Indeed EW1 implies that for all $\mathbf{x} \in \{0,1\}^I$,

$$G\overline{G}(\mathbf{x}) = F(\mathbf{x}, \mathbf{p})F(\overline{\mathbf{x}}, \mathbf{p}) = \sum_{M\in\mathcal{M}_{\mathbf{I}}} B(\mathbf{x}, \mathbf{x} \oplus \mathbf{I}, M) = \sum_{M\in\mathcal{M}_{\mathbf{I}}} D_{\mathbf{p}}(\mathbf{x}, M),$$

while EW2 implies

$$D_{\mathbf{p}}(\mathbf{x}, M) = B((\mathbf{x}, \mathbf{p}), (\overline{\mathbf{x}}, \mathbf{p}), M) = B((\mathbf{x} \oplus \mathbf{S}, \mathbf{p}), (\overline{\mathbf{x}} \oplus \mathbf{S}, \mathbf{p}), M) = D_{\mathbf{p}}(\mathbf{x} \oplus \mathbf{S}, M)$$

for all $\mathbf{x} \in \{0,1\}^I$ and all $S \in M \in \mathcal{M}_{\mathbf{I}}$.

For the backwards direction, for each $I \subseteq J$ and each $\mathbf{p} \in \{0,1\}^I$, pick a 2-decomposition $D_{\mathbf{p}}$ of $G\overline{G}$ where G is the pinning of F by $\mathbf{p}$. For all $\mathbf{x}, \mathbf{y} \in \{0,1\}^J$, define $B(\mathbf{x},\mathbf{y},M) = D_{\mathbf{p}}(\mathbf{x}',M)$ where $\mathbf{p}$ is the restriction of $\mathbf{x}$ to $\{i \in J \mid x_i = y_i\}$ and $\mathbf{x}'$ is the restriction of $\mathbf{x}$ to $\{i \in J \mid x_i \neq y_i\}$. Then B witnesses that F is even-windable: EW1 is

$$F(\mathbf{x})F(\mathbf{y}) = G\overline{G}(\mathbf{x}') = \sum_M D_{\mathbf{p}}(\mathbf{x}',M) = \sum_M B(\mathbf{x},\mathbf{y},M),$$

where M ranges over partitions of $\{i \in J \mid x_i \neq y_i\}$ into pairs, while EW2 is

$$B(\mathbf{x},\mathbf{y},M) = D_{\mathbf{p}}(\mathbf{x}',M) = D_{\mathbf{p}}(\mathbf{x}' \oplus \mathbf{S},M) = B(\mathbf{x} \oplus \mathbf{S},\mathbf{y} \oplus \mathbf{S},M)$$

for all partitions M of $\{i \in J \mid x_i \neq y_i\}$ into pairs and all $S \in M$.

Lemma 3.7. *Let $F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ with $|J| \leq 3$. If F is even then F is even-windable.*

Proof. Let $G\colon \{0,1\}^I \to \mathbb{Q}_{\geq 0}$ be a pinning of F.

If $I = \emptyset$ define $D(\mathbf{x},\emptyset) = G\overline{G}(\mathbf{x})$ where $\mathbf{x} \in \{0,1\}^\emptyset$ is the empty configuration. Then $G\overline{G}(\mathbf{x}) = \sum_M D(\mathbf{x},M)$ where M ranges over the set $\{\emptyset\}$ of partitions of I into pairs, so D is a 2-decomposition of $G\overline{G}$.

If $|I| = 2$, let i,j be the elements of I and define $D(\mathbf{x},\{\{i,j\}\}) = G\overline{G}(\mathbf{x})$. For all $\mathbf{x} \in \{0,1\}^I$ we have $G\overline{G}(\mathbf{x}) = \sum_M D(\mathbf{x},M)$ where M ranges over the set $\{\{\{i,j\}\}\}$ of partitions of I into pairs, so D is a 2-decomposition of $G\overline{G}$.

If $|I|$ is 1 or 3 then $G(\mathbf{x})$ and $G(\overline{\mathbf{x}})$ cannot be simultaneously be non-zero because G is a pinning of the even function F, and $\sum_{i\in I} x_i \equiv |I| + \sum_{i \in I}(1-x_i) \pmod 2$. Thus $G\overline{G}$ is identically zero. There are also no partitions of I into pairs, so the empty function is a 2-decomposition of $G\overline{G}$. ☐

Lemma 3.8. $\mathbf{Even}_J$ *and* $\mathbf{Odd}_J$ *have a 2-decomposition whenever $|J|$ is even.* $\mathbf{Even}_J$ *and* $\mathbf{Odd}_J$ *are even-windable for any J.*

Proof. First consider $\mathbf{Even}_J$. Fix a partition N of J into pairs. Define

$$D(\mathbf{x},M) = \begin{cases} 1 & \text{if } M = N \text{ and } \sum_{i\in J} x_i \text{ is even} \\ 0 & \text{otherwise.} \end{cases}$$

Then for all $\mathbf{x} \in \{0,1\}^J$ we have $\mathbf{Even}_J(\mathbf{x}) = \sum_M D(\mathbf{x},M)$ (where the sum ranges over partitions M of J into pairs). Similarly for $\mathbf{Odd}_J$, define

$$D(\mathbf{x},M) = \begin{cases} 1 & \text{if } M = N \text{ and } \sum_{i\in J} x_i \text{ is odd} \\ 0 & \text{otherwise.} \end{cases}$$

Then for all $\mathbf{x} \in \{0,1\}^J$ we have $\mathbf{Odd}_J(\mathbf{x}) = \sum_M D(\mathbf{x},M)$.

Now consider a pinning $G\colon \{0,1\}^K \to \mathbb{Q}_{\geq 0}$ of $\mathbf{Even}_J$ or $\mathbf{Odd}_J$. If $|K|$ is odd then $G\overline{G}$ is identically zero, by the same argument used in Lemma 3.7. Otherwise, $G = G\overline{G}$ is either $\mathbf{Even}_K$ or $\mathbf{Odd}_K$, which we showed have 2-decompositions. ☐

The following argument gives a more difficult example of a 2-decomposition. It will be used later (in the proof of Lemma 3.17) to show that $\mathbf{NAE}_J$ is windable.

Lemma 3.9. *Let J be an finite set with $|J|$ even. Then $\mathbf{EvenNAE}_J$ has a 2-decomposition.*

Proof. For each subset $I \subseteq J$ of even order fix a partition M_I of I into pairs. Set

$$D(\mathbf{x}, M) = 2^{-k+2} \left| \{ I \subseteq J \mid |I| \text{ is even}, \sum_{i \in I} x_i \text{ and } \sum_{i \in J \setminus I} x_i \text{ are odd, and } M = M_I \cup M_{J \setminus I} \} \right| .$$

$S \in M_I \cup M_{J\setminus I}$ implies $S \subseteq I$ or $S \subseteq J \setminus I$. The conditions that $\sum_{i \in I} x_i$ and $\sum_{i \in J \setminus I} x_i$ are odd are therefore not affected by changing $\mathbf{x}$ to $\mathbf{x} \oplus \mathbf{S}$. Thus $D(\mathbf{x} \oplus \mathbf{S}, M) = D(\mathbf{x}, M)$ for all $S \in M$.

For any $\mathbf{x}$, if $\mathbf{EvenNAE}_J(\mathbf{x}) = 0$ then $D(\mathbf{x}, M) = 0$. If $\mathbf{EvenNAE}_J(\mathbf{x}) = 1$, pick i, j with $x_i = 0$ and $x_j = 1$. For each of the 2^{k-2} subsets $I' \subseteq J \setminus \{i, j\}$ there is a unique set $I'' \subseteq \{i, j\}$ such that the order of $I = I' \cup I''$ is even and such that $\sum_{i \in I} x_i$ and $\sum_{i \in J \setminus I} x_i$ are odd. There are thus 2^{k-2} such subsets I for each fixed $\mathbf{x}$, which gives $\sum_M D(\mathbf{x}, M) = 1$. So D is a 2-decomposition of $\mathbf{EvenNAE}_J$. $\square$

3.3.4 Expressibility

We will show that the weight-function of any circuit using even-windable functions is even-windable. We will use a certain graph associated with a choice of matching of incidences.

Let M and E each be a set of disjoint pairs of some set. Define $L_E(M)$ to be the multigraph on the vertex set $\bigcup_{S \in M} S$ with edge set the disjoint union of M and $\{\{i, j\} \in E \mid i, j \in \bigcup_{S \in M} S\}$ (so edges in $M \cap E$ give pairs of parallel edges in $L_E(M)$).

Note that for each vertex i of $L_E(M)$, the degree of i is two if $\{i, j\} \in E$ for some $j \in \bigcup_{S \in M} S$, and otherwise i has degree one. So $L_E(M)$ consists of paths and cycles.

We will use this graph later for the analysis of the near-assignments chain. For now, consider an assignment $\mathbf{x}$ of some circuit with internal edges E and external edges A, and let $M \in \mathcal{M}_{\mathbf{x}}$. For any i not in A with $x_i = 1$, the unique j with $\{i, j\} \in E$ satisfies $x_j = 1$. This means that $i \in \bigcup_{S \in M} S$ has degree 1 in $L_E(M)$ if and only if $i \in A$. So every path component of $L_E(M)$ ends in $\{i \in A \mid x_i = 1\}$, and every such i is at the end of a path. See Figure 3.4.

Lemma 3.10. *Let φ be a circuit using only weight-functions that are even-windable. The weight-function of φ is even-windable.*

Proof. Recall that V, J, J_v, A, E, F_v denote vertices, incidences, vertices' incidences, external edges, internal edges, and vertices' weight-functions. For each $v \in V$ pick a function B_v witnessing that F_v is even-windable.

Consider a set M' of disjoint pairs of J. We will say that M' *induces* the set of pairs $\{i, j\} \subseteq A$ such that there is a path from i to j in $L_E(M')$.

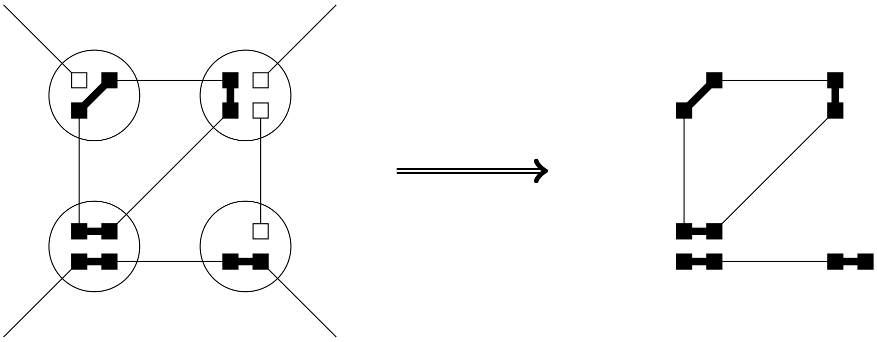

FIGURE 3.4: A circuit φ and $L_E(M)$ for some $M \in \mathcal{M}_{\mathbf{x}}$ where $\mathbf{x}$ is an assignment of $E = E^{\varphi}$. (In particular, the M drawn is a union of partitions $M_v \in \mathcal{M}_{\mathbf{x}|_{J_v}}$.) Circles represent vertices of the circuit. Squares are incidences $i \in J^{\varphi}$ of the circuit, and are filled black where $x_i = 1$. Elements of M are drawn as thick black lines. Elements $\{i, j\} \in E$ are drawn as thin lines.

For all $\mathbf{x}, \mathbf{y} \in \{0,1\}^A$ and all $M \in \mathcal{M}_{\mathbf{x}\oplus\mathbf{y}}$ define

$$B(\mathbf{x}, \mathbf{y}, M) = \sum_{\mathbf{x}', \mathbf{y}'} \sum_{\{M_v\} \text{ inducing } M} \prod_{v \in V} B_v(\mathbf{x}'|_{J_v}, \mathbf{y}'|_{J_v}, M_v)$$

where:

- $\sum_{\mathbf{x}', \mathbf{y}'}$ denotes the sum over assignments $\mathbf{x}'$ and $\mathbf{y}'$ extending $\mathbf{x}$ and $\mathbf{y}$ respectively.
- $\sum_{\{M_v\} \text{ inducing } M}$ denotes the sum over all choices of $M_v \in \mathcal{M}_{(\mathbf{x}'\oplus\mathbf{y}')|_{J_v}}$ for each $v \in V$, such that $\bigcup_{v\in V} M_v$ induces M.

For all $\mathbf{x}, \mathbf{y} \in \{0,1\}^A$ we have

$$\begin{aligned}
\sum_{M \in \mathcal{M}_{\mathbf{x}\oplus\mathbf{y}}} B(\mathbf{x}, \mathbf{y}, M) &= \sum_{M \in \mathcal{M}_{\mathbf{x}\oplus\mathbf{y}}} \sum_{\mathbf{x}', \mathbf{y}'} \sum_{\{M_v\} \text{ inducing } M} \prod_{v \in V} B_v(\mathbf{x}'|_{J_v}, \mathbf{y}'|_{J_v}, M_v) \\
&= \sum_{\mathbf{x}', \mathbf{y}'} \sum_{\{M_v\}} \prod_{v \in V} B_v(\mathbf{x}'|_{J_v}, \mathbf{y}'|_{J_v}, M_v) \\
&= \sum_{\mathbf{x}', \mathbf{y}'} \prod_{v \in V} F_v(\mathbf{x}'|_{J_v}) F_v(\mathbf{y}'|_{J_v}) \\
&= [\![\varphi]\!](\mathbf{x})\, [\![\varphi]\!](\mathbf{y}).
\end{aligned}$$

Here $\sum_{\{M_v\}}$ denotes the sum over *all* choices of $M_v \in \mathcal{M}_{(\mathbf{x}'\oplus\mathbf{y}')|_{J_v}}$ for each $v \in V$: the sum over M eliminates the condition that $\bigcup_{v\in V} M_v$ induces M.

Now fix $\mathbf{x}, \mathbf{y} \in \{0,1\}^A$ and $S = \{i, j\} \in M \in \mathcal{M}_{\mathbf{x}\oplus\mathbf{y}}$. For any choice of $\{M_v\}$ inducing M, there is a unique path component $P_{\{M_v\}}$ (also depending on $\mathbf{x}, \mathbf{y}, S$) from i to j in $L = L_E(\bigcup_{v\in V} \bigcup_{S \in M_v} S)$. By construction of L, the vertices of $P_{\{M_v\}}$ are a union of pairs $S \in \bigcup_{v\in V} M_v$. In particular, for each $v \in V$, the intersection $P_{\{M_v\}} \cap J_v$ is a union of pairs $S \in M_v$. Using EW2' we have

$$\begin{aligned} B(\mathbf{x},\mathbf{y},M) &= \sum_{\mathbf{x}',\mathbf{y}'} \sum_{\{M_v\} \text{ inducing } M} \prod_{v \in V} B_v(\mathbf{x}'|_{J_v}, \mathbf{y}'|_{J_v}, M_v) \\ &= \sum_{\mathbf{x}',\mathbf{y}'} \sum_{\{M_v\} \text{ inducing } M} \prod_{v \in V} B_v((\mathbf{x}' \oplus \mathbf{P}_{\{M_v\}})|_{J_v}, (\mathbf{y}' \oplus \mathbf{P}_{\{M_v\}})|_{J_v}, M_v) \\ &= B(\mathbf{x} \oplus \mathbf{S}, \mathbf{y} \oplus \mathbf{S}, M). \end{aligned}$$

So B witnesses that $[\![\varphi]\!]$ is even-windable. □

3.3.5 The near-assignments Markov chain

Throughout this subsection fix an even-windable weight-function $F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ and a partition E of J into pairs. This can be thought of as a circuit with one vertex. We will define and study the *near-assignments Markov chain* for (F, E).

Set $n = |J|$. For each $k \geq 0$ let Ω_k denote the set of k-assignments of J with respect to E that satisfy $F(\mathbf{x}) > 0$. The state-space is $\Omega = \Omega_0 \cup \Omega_2$. The transitions are Metropolis updates to states at distance two. More specifically, the transition probability from $\mathbf{x}$ to $\mathbf{y}$ is defined to be

$$P(\mathbf{x},\mathbf{y}) = \begin{cases} \frac{2}{n^2}\min(1, F(\mathbf{y})/F(\mathbf{x})) & \text{if } \mathrm{d}(\mathbf{x},\mathbf{y}) = 2 \\ 1 - \frac{2}{n^2}\sum_{\mathbf{y}'\colon\ \mathrm{d}(\mathbf{x},\mathbf{y}')=2}\min(1, F(\mathbf{y}')/F(\mathbf{x})) & \text{if } \mathbf{y} = \mathbf{x} \\ 0 & \text{otherwise.} \end{cases}$$

(We will not consider the initial state to be part of the Markov chain itself: the Markov chain is completely described by the matrix $P \in \mathbb{R}^{\Omega \times \Omega}$.) Define a probability distribution π on Ω by

$$\pi(\mathbf{x}) = F(\mathbf{x}) \Big/ \sum_{\mathbf{y} \in \Omega} F(\mathbf{y}) \qquad (\mathbf{x} \in \Omega).$$

By abuse of notation we will also denote $\sum_{\mathbf{x} \in X} \pi(\mathbf{x})$ by $\pi(X)$ for subsets $X \subseteq \Omega$. By adapting the arguments of [70], we will show:

Theorem 3.11. *For all $\mathbf{x} \in \Omega$ and all non-negative integers t, we have*

$$\frac{1}{2}\sum_{\mathbf{y} \in \Omega} |P^t(\mathbf{x},\mathbf{y}) - \pi(\mathbf{y})| \leq \frac{1}{2}\pi(\mathbf{x})^{-1/2}\exp(-t\pi(\Omega_0)^2/n^4)$$

Here P^t denotes the t'th matrix power. The factor of $\frac{1}{2}$ is convention: the left hand side is called the *total variation distance* of P^t from π.

We will use a congestion argument, with the following definitions. In this section a "**path**" is a directed path γ in the transition graph (the directed graph with vertex set Ω and an arc $(\mathbf{x},\mathbf{y})$ whenever $P(\mathbf{x},\mathbf{y}) > 0$). A **flow from** $X \subseteq \Omega$ **to** $Y \subseteq \Omega$ is a

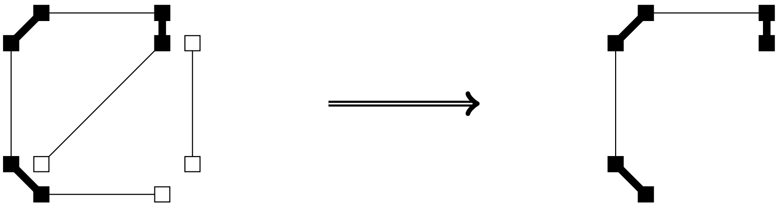

FIGURE 3.5: $L_E(M)$ for some 2-assignment-matching M. Squares are elements of J. Elements of M are drawn as thick black lines. Elements $\{i,j\} \in E$ are drawn as thin lines.

non-negative real-valued function f defined on paths which start in X and end in Y, satisfying

$$\sum_{\text{paths } \gamma \text{ from } x \text{ to } y} f(\gamma) = \pi(x)\pi(y) \qquad (x \in X, y \in Y).$$

The **congestion** of f is

$$\rho(f) = \max_{\text{transitions } (\mathbf{x},\mathbf{y})} \frac{1}{\pi(\mathbf{x})P(\mathbf{x},\mathbf{y})} \sum_{\text{paths } \gamma \text{ with } \gamma \ni (\mathbf{x},\mathbf{y})} f(\gamma).$$

These definitions of flows and ρ agree with [90, Section 4].

In the following arguments we will often use k-assignments (with respect to E) of the form $\mathbf{x} \oplus \mathbf{y}$ for $\mathbf{x} \in \Omega_{k_1}$ and $\mathbf{y} \in \Omega_{k_2}$. Note that we do not require $F(\mathbf{x}) > 0$ for k-assignments $\mathbf{x}$, though we do require $F(\mathbf{x}) > 0$ for $\mathbf{x} \in \Omega_k$ (so Ω_k, defined at the start of this section, is the set of "satisfying" k-assignments). For any non-negative integer k, a **k-assignment-matching** (with respect to E) is a set M of disjoint pairs of J such that exactly k edges $\{i,j\} \in E$ have exactly one endpoint, i or j, in $\bigcup_{S \in M} S$. In other words, the characteristic vector of $\bigcup_{S \in M} S$ is a k-assignment.

Consider a k-assignment-matching M. Recall the definition of $L_E(M)$ given in Section 3.3.4, which consists of cycles and paths. For any i with $i \in \bigcup_{S \in M} S$, the unique j with $\{i,j\} \in E$ satisfies $j \in \bigcup_{S \in M} S$, except for exactly k values i. Thus $L_E(M)$ has precisely $k/2$ path components. See Figure 3.5.

For all non-negative integers k define

$$Z_k = \sum_{\mathbf{x} \in \Omega_k} F(\mathbf{x}).$$

(This is $Z_k(\varphi)$ if we consider F as a one-vertex circuit φ.)

Lemma 3.12. *$Z_0 Z_4 \leq Z_2 Z_2$.*

Proof. We have

$$Z_0 Z_4 = \sum_{\substack{\mathbf{x} \in \Omega_0 \\ \mathbf{y} \in \Omega_4}} F(\mathbf{x})F(\mathbf{y}) = \sum_{\substack{\mathbf{x} \in \Omega_0 \\ \mathbf{y} \in \Omega_4}} \sum_{M \in \mathcal{M}_{\mathbf{x} \oplus \mathbf{y}}} B(\mathbf{x},\mathbf{y},M).$$

For each 4-assignment-matching M, pick a path component of $L_E(M)$ and let H_M be the set of vertices of this component. Let B be a function witnessing that F is even-windable. Each H_M is a union of pairs in M so by EW2',

$$Z_0 Z_4 = \sum_{\substack{\mathbf{x}\in\Omega_0\\ \mathbf{y}\in\Omega_4}} \sum_{M\in\mathcal{M}_{\mathbf{x}\oplus\mathbf{y}}} B(\mathbf{x}\oplus\mathbf{H}_M, \mathbf{y}\oplus\mathbf{H}_M, M).$$

Let $\mathbf{x}' = \mathbf{x}\oplus\mathbf{H}_M$ and $\mathbf{y}' = \mathbf{y}\oplus\mathbf{H}_M$. Note that $\mathbf{x}', \mathbf{y}' \in \Omega_2$, and that $(\mathbf{x}', \mathbf{y}', M)$ uniquely determines $(\mathbf{x}, \mathbf{y}, M)$, or in other words, the map $(\mathbf{x}, \mathbf{y}, M) \mapsto (\mathbf{x}', \mathbf{y}', M)$ is an injection. Therefore we get an upper bound by summing over all $\mathbf{x}', \mathbf{y}' \in \Omega_2$:

$$\begin{aligned} Z_0 Z_4 &\le \sum_{\substack{\mathbf{x}'\in\Omega_2\\ \mathbf{y}'\in\Omega_2}} \sum_{M\in\mathcal{M}_{\mathbf{x}'\oplus\mathbf{y}'}} B(\mathbf{x}', \mathbf{y}', M) \\ &= \sum_{\substack{\mathbf{x}'\in\Omega_2\\ \mathbf{y}'\in\Omega_2}} F(\mathbf{x}')F(\mathbf{y}') = Z_2 Z_2. \end{aligned}$$

□

Lemma 3.13. *Assume $Z_0 > 0$. There is a flow f_0 from Ω_0 to Ω with congestion at most $\frac{1}{2}n^3/\pi(\Omega_0)$, such that $f_0(\gamma) = 0$ for paths γ of length more than $n/2$.*

Proof. We will first construct a "winding" enumeration $S(M, 1), \ldots, S(M, |M|)$ of each 0- or 2-assignment-matching M. The property we will need is that for each $0 \le k \le |M|$, the characteristic vector of $S(M, 1) \cup \cdots \cup S(M, k)$ is a 0- or 2-assignment.

First define the final pair $T(M) = S(M, |M|)$ for all non-empty 0- or 2-assignment-matchings M as follows. If M is a non-empty 0-assignment-matching (so $L_E(M)$ consists of cycles), pick any vertex $i \in L_E(M)$. If M is a non-empty 2-assignment-matching, pick an endpoint i of the unique path component in $L_E(M)$. In either case let j be the unique index with $\{i, j\} \in M$, and set $T(M) = \{i, j\}$. In any case $L_E(M \setminus \{T(M)\}) = L_E(M) \setminus \{i, j\}$ has at most one path component.

So $M \setminus \{T(M)\}$ is a 0- or 2-assignment-matching. By induction on $|M| - k$ define

$$S(M, k) = T(M \setminus \{S(M, k+1), \ldots, S(M, |M|)\}).$$

So $M \setminus \{S(M, k+1), \ldots, S(M, |M|)\}$ is always a 0- or 2-assignment-matching. This completes the construction of $S(M, k)$.

For each $\mathbf{x} \in \Omega_0$ and $\mathbf{y} \in \Omega$ and $M \in \mathcal{M}_{\mathbf{x}\oplus\mathbf{y}}$, let $\gamma_{\mathbf{x},\mathbf{y},M}$ denote the path

$$\mathbf{x} = \mathbf{x}\oplus\mathbf{J}_{M,0} \to \mathbf{x}\oplus\mathbf{J}_{M,1} \to \cdots \to \mathbf{x}\oplus\mathbf{J}_{M,|M|} = \mathbf{y},$$

where $\mathbf{J}_{M,k}$ denotes the characteristic vector of $S(M, 1) \cup \cdots \cup S(M, k)$. Let B be a function witnessing that F is even-windable. For all paths γ from $\mathbf{x} \in \Omega_0$ to $\mathbf{y} \in \Omega$, define

$$f_0(\gamma) = \sum_{\substack{M \in \mathcal{M}_{\mathbf{x}\oplus\mathbf{y}} \\ \gamma = \gamma_{\mathbf{x},\mathbf{y},M}}} B(\mathbf{x}, \mathbf{y}, M)/(Z_0 + Z_2)^2.$$

f_0 is a flow from Ω_0 to Ω because for all $\mathbf{x} \in \Omega_0$ and $\mathbf{y} \in \Omega$ we have

$$\begin{aligned}\sum_{\text{paths } \gamma \text{ from } \mathbf{x} \text{ to } \mathbf{y}} f_0(\gamma) &= \sum_{M \in \mathcal{M}_{\mathbf{x}\oplus\mathbf{y}}} B(\mathbf{x}, \mathbf{y}, M)/(Z_0 + Z_2)^2 \\ &= F(\mathbf{x})F(\mathbf{y})/(Z_0 + Z_2)^2 \\ &= \pi(\mathbf{x})\pi(\mathbf{y}).\end{aligned}$$

The congestion of f_0 is, by definition,

$$\rho(f_0) = \max_{\text{transitions } (\mathbf{z}, \mathbf{z}')} \frac{1}{\pi(\mathbf{z})P(\mathbf{z}, \mathbf{z}')} \sum_{\text{paths } \gamma \text{ with } \gamma \ni (\mathbf{z}, \mathbf{z}')} f_0(\gamma).$$

But $\pi(\mathbf{z})P(\mathbf{z}, \mathbf{z}') = \frac{2}{n^2} \min(\pi(\mathbf{z}), \pi(\mathbf{z}'))$, so

$$\begin{aligned}\rho(f_o) &\le \max_{\mathbf{z} \in \Omega} \frac{n^2}{2 \cdot \pi(\mathbf{z})} \sum_{\text{paths } \gamma \text{ with } \gamma \ni \mathbf{z}} f_0(\gamma) \\ &= \max_{\mathbf{z} \in \Omega} \frac{n^2}{2F(\mathbf{z})(Z_0 + Z_2)} \sum_{\substack{\mathbf{x} \in \Omega_0 \\ \mathbf{y} \in \Omega}} \sum_{\substack{M \in \mathcal{M}_{\mathbf{x}\oplus\mathbf{y}} \\ \text{with } \mathbf{z} \in \gamma_{\mathbf{x},\mathbf{y},M}}} B(\mathbf{x}, \mathbf{y}, M)\end{aligned}$$

In the last summation, $\mathbf{z} \in \gamma_{\mathbf{x},\mathbf{y},M}$ implies $\mathbf{z} = \mathbf{x} \oplus \mathbf{J}_{M,k}$ for some k, so by EW2' we have $B(\mathbf{x}, \mathbf{y}, M) = B(\mathbf{z}, \mathbf{z} \oplus \mathbf{w}, M)$ where $\mathbf{w} = \mathbf{x} \oplus \mathbf{y}$. Thus,

$$\rho(f_0) \le \max_{\mathbf{z} \in \Omega} \frac{n^2}{2F(\mathbf{z})(Z_0 + Z_2)} \sum_{\text{0- and 2-assignments } \mathbf{w}} \sum_{\mathbf{x} \in \Omega_0} \sum_{\substack{M \in \mathcal{M}_{\mathbf{w}} \\ \text{with } \mathbf{z} \in \gamma_{\mathbf{x},\mathbf{x}\oplus\mathbf{w},M}}} B(\mathbf{z}, \mathbf{z} \oplus \mathbf{w}, M)$$

For each $(\mathbf{z}, \mathbf{w}, M)$ with $M \in \mathcal{M}_{\mathbf{w}}$, the only values of $\mathbf{x}$ such that $\mathbf{z} \in \gamma_{\mathbf{x},\mathbf{x}\oplus\mathbf{w},M}$ are the $|M| + 1$ values $\mathbf{z} \oplus \mathbf{J}_{M,0}, \ldots, \mathbf{z} \oplus \mathbf{J}_{M,|M|}$. Thus,

$$\begin{aligned}\rho(\Omega_0) &\le \max_{\mathbf{z} \in \Omega} \frac{n^2}{2F(\mathbf{z})(Z_0 + Z_2)} \sum_{\text{0- and 2-assignments } \mathbf{w}} (|M| + 1) \sum_{M \in \mathcal{M}_{\mathbf{w}}} B(\mathbf{z}, \mathbf{z} \oplus \mathbf{w}, M) \\ &\le \max_{\mathbf{z} \in \Omega} \frac{n^2(n/2 + 1)}{2F(\mathbf{z})(Z_0 + Z_2)} \sum_{\text{0- and 2-assignments } \mathbf{w}} F(\mathbf{z})F(\mathbf{z} \oplus \mathbf{w})\end{aligned}$$

Using $\mathbf{z} \oplus \mathbf{w} \in \Omega_0 \cup \Omega_2 \cup \Omega_4$,

$$\rho(\Omega_0) \le \frac{n^3}{2} \cdot \frac{Z_0 + Z_2 + Z_4}{Z_0 + Z_2}.$$

If $Z_2 = 0$ then by Lemma 3.12 we also have $Z_4 = 0$, so the congestion is at most $n^3/2$. Otherwise by Lemma 3.12 we have $Z_4/Z_2 \leq Z_2/Z_0$ and

$$\frac{Z_0 + Z_2 + Z_4}{Z_0 + Z_2} \leq 1 + \frac{Z_4}{Z_2} \leq 1 + \frac{Z_2}{Z_0} = 1 \Big/ \frac{Z_0}{Z_0 + Z_2} = 1/\pi(\Omega_0).$$

□

Lemma 3.14. *Assume $Z_0 > 0$. There is a flow f from Ω to Ω with congestion at most $n^3/\pi(\Omega_0)^2$, such that $f(\gamma) = 0$ for paths γ of length greater than n.*

Proof. As in [72], we will randomly route through Ω_0.

In this proof, let $\Gamma(\mathbf{y}, \mathbf{x})$ denote the set of paths that start at $\mathbf{y}$ and end at $\mathbf{x}$. For all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \Omega$, all $g \in \Gamma(\mathbf{y}, \mathbf{x})$ and all $g' \in \Gamma(\mathbf{y}, \mathbf{z})$, let $\overline{g}g'$ denote the path from $\mathbf{x}$ to $\mathbf{z}$ obtained by appending g' to the reverse of g. For all paths γ from $\mathbf{x} \in \Omega$ to $\mathbf{z} \in \Omega$, define

$$f(\gamma) = \sum_{\mathbf{y} \in \Omega_0} \sum_{\substack{g \in \Gamma(\mathbf{y}, \mathbf{x}) \\ g' \in \Gamma(\mathbf{y}, \mathbf{z}) \\ \overline{g}g' = \gamma}} f_0(g) f_0(g')/\pi(\mathbf{y})\pi(\Omega_0).$$

Then f is a flow from Ω to Ω: for all $\mathbf{x}, \mathbf{z} \in \Omega$ we have

$$\begin{aligned}
\sum_{\gamma \in \Gamma(\mathbf{x}, \mathbf{z})} f(\gamma) &= \sum_{\mathbf{y} \in \Omega_0} \sum_{\substack{g \in \Gamma(\mathbf{y}, \mathbf{x}) \\ g' \in \Gamma(\mathbf{y}, \mathbf{z})}} f_0(g) f_0(g')/\pi(\mathbf{y})\pi(\Omega_0) \\
&= \sum_{\mathbf{y} \in \Omega_0} \pi(\mathbf{x})\pi(\mathbf{y})\pi(\mathbf{y})\pi(\mathbf{z})/\pi(\mathbf{y})\pi(\Omega_0) \\
&= \pi(\mathbf{x})\pi(\mathbf{z}).
\end{aligned}$$

Letting $(\mathbf{w}, \mathbf{w}')$ denote an arbitrary transition, the congestion of f is

$$\begin{aligned}
\rho(f) &= \max_{(\mathbf{w}, \mathbf{w}')} \frac{1}{\pi(\mathbf{w}) P(\mathbf{w}, \mathbf{w}')} \sum_{\text{paths } \gamma \text{ with } \gamma \ni (\mathbf{w}, \mathbf{w}')} f(\gamma) \\
&= \max_{(\mathbf{w}, \mathbf{w}')} \frac{1}{\pi(\mathbf{w}) P(\mathbf{w}, \mathbf{w}')} \sum_{\substack{\mathbf{x}, \mathbf{z} \in \Omega \\ \mathbf{y} \in \Omega_0}} \sum_{\substack{g \in \Gamma(\mathbf{y}, \mathbf{x}) \\ g' \in \Gamma(\mathbf{y}, \mathbf{z}) \\ \text{such that } (\mathbf{w}, \mathbf{w}') \in \overline{g}g'}} \frac{f_0(g) f_0(g')}{\pi(\mathbf{y})\pi(\Omega_0)}.
\end{aligned}$$

By symmetry, half of the sum comes from $(\mathbf{w}, \mathbf{w}') \in g'$ terms:

$$\begin{aligned}
\rho(f) &= 2 \max_{(\mathbf{w},\mathbf{w}')} \frac{1}{\pi(\mathbf{w})P(\mathbf{w},\mathbf{w}')} \sum_{\substack{\mathbf{x},\mathbf{z}\in\Omega \\ \mathbf{y}\in\Omega_0}} \sum_{\substack{g\in\Gamma(\mathbf{y},\mathbf{x}) \\ g'\in\Gamma(\mathbf{y},\mathbf{z}) \\ \text{such that } (\mathbf{w},\mathbf{w}')\in g'}} \frac{f_0(g)f_0(g')}{\pi(\mathbf{y})\pi(\Omega_0)} \\
&= 2 \max_{(\mathbf{w},\mathbf{w}')} \frac{1}{\pi(\mathbf{w})P(\mathbf{w},\mathbf{w}')} \sum_{\substack{\mathbf{x},\mathbf{z}\in\Omega \\ \mathbf{y}\in\Omega_0}} \sum_{\substack{g'\in\Gamma(\mathbf{y},\mathbf{z}) \\ \text{such that } (\mathbf{w},\mathbf{w}')\in g'}} \frac{\pi(\mathbf{x})\pi(\mathbf{y})f_0(g')}{\pi(\mathbf{y})\pi(\Omega_0)} \\
&= 2 \max_{(\mathbf{w},\mathbf{w}')} \frac{1}{\pi(\mathbf{w})P(\mathbf{w},\mathbf{w}')} \sum_{\substack{\mathbf{z}\in\Omega \\ \mathbf{y}\in\Omega_0}} \sum_{\substack{g'\in\Gamma(\mathbf{y},\mathbf{z}) \\ \text{such that } (\mathbf{w},\mathbf{w}')\in g'}} f_0(g')/\pi(\Omega_0) \\
&= 2\rho(f_0)/\pi(\Omega_0) \\
&\leq n^3/\pi(\Omega_0)^2
\end{aligned}$$

by Lemma 3.13. □

The remaining task is to relate the congestion to Markov chain mixing.

Theorem 3.11. For all $\mathbf{x} \in \Omega$ and all non-negative integers t, we have

$$\frac{1}{2}\sum_{\mathbf{y}\in\Omega} |P^t(\mathbf{x},\mathbf{y}) - \pi(\mathbf{y})| \leq \frac{1}{2}\pi(\mathbf{x})^{-1/2}\exp(-t\pi(\Omega_0)^2/n^4).$$

Proof. The transition matrix P is reversible relative to π: it obeys the detailed balance condition

$$\pi(\mathbf{y})P(\mathbf{y},\mathbf{z}) = \pi(\mathbf{z})P(\mathbf{z},\mathbf{y}) \text{ for all } \mathbf{y},\mathbf{z} \in \Omega.$$

We have

$$P(\mathbf{x},\mathbf{x}) \geq 1 - \frac{2}{n^2}\binom{n}{2} \geq 1/n \qquad \text{for all } \mathbf{x} \in \Omega. \tag{3.2}$$

In particular, the Markov chain is aperiodic. Also, by Lemma 3.14 there exists a flow f from Ω to Ω, which implies that that the Markov chain is connected. This allows us to use the results from [90] and [44]. P has eigenvalues

$$1 = \lambda_0 > \lambda_1 \geq \ldots \lambda_{|\Omega|-1} \geq -1.$$

By Lemma 3.14 and [90, Corollary 6'],

$$\begin{aligned}
\lambda_1 &\leq 1 - \frac{1}{\rho(\Gamma)n} \\
&\leq 1 - \pi(\Omega_0)^2/n^4.
\end{aligned}$$

By (3.2) and equation 1 of [65] we have

$$-\lambda_{|\Omega|-1} \leq 1 - 2/n \leq \lambda_1.$$

By [44, Proposition 3],

$$\begin{aligned}\frac{1}{2}\sum_{\mathbf{y}\in\Omega}|P^t(\mathbf{x},\mathbf{y})-\pi(\mathbf{y})| &\le \frac{1}{2}\sqrt{\frac{1-\pi(\mathbf{x})}{\pi(\mathbf{x})}}\max(\lambda_1,-\lambda_{|\Omega|-1})^t \\ &\le \frac{1}{2}\pi(\mathbf{x})^{-1/2}\exp(-t\pi(\Omega_0)^2/n^4).\end{aligned}$$

□

3.4 Windable functions

In this section we extend the analysis of even-windable functions to windable functions. The definition of windability is a natural extension of even-windability, but turns out not to give much extra generality.

For all $F\colon \{0,1\}^J \to \mathbb{Q}_{\ge 0}$ define $F_\oplus\colon \{0,1\}^{1+J} \to \mathbb{Q}_{\ge 0}$ by

$$F_\oplus(p;\mathbf{x}) = \begin{cases} F(\mathbf{x}) & \text{if } p+\sum_{i\in J} x_i \text{ is even} \\ 0 & \text{otherwise}\end{cases} \qquad (p\in\{0,1\},\mathbf{x}\in\{0,1\}^J).$$

Lemma 3.15. $F\colon \{0,1\}^J \to \mathbb{Q}_{\ge 0}$ *is windable if and only if* $F_\oplus$ *is even-windable.*

Proof. ($\Rightarrow$) Pick an ordering of J. Consider a partition M of a subset $I\subseteq J$ into singletons $\{a_1\},\dots,\{a_k\}$ and pairs $S_1,\dots,S_\ell$. Define $\mu(M)$, when $|I|$ is even, to be the union of $\{S_1,\dots,S_\ell\}$ with a partition (depending only on M) of $\{a_1,\dots,a_k\}$ into pairs. Define $\mu(M)$, when $|I|$ is odd, to be the union of $\{S_1,\dots,S_\ell\}$ with a partition of $\{1,a_1,\dots,a_k\}$ into pairs. Let B be a witness that F is windable. For all $(p;\mathbf{x}),(q;\mathbf{y})\in\{0,1\}^{1+J}$ and all $M\in\mathcal{M}_{(p;\mathbf{x})\oplus(q;\mathbf{y})}$, define

$$B'((p;\mathbf{x}),(q;\mathbf{y}),M) = \begin{cases}\sum_{M'\colon \mu(M')=M} B(\mathbf{x},\mathbf{y},M') & \text{if } p+\sum_{i\in J}x_i \text{ and } q+\sum_{i\in J}y_i \text{ are even}\\ 0 & \text{otherwise}\end{cases}$$

For all $S\in M=\mu(M')$, if we let $S'=S\setminus\{1\}$ then $B(\mathbf{x}\oplus\mathbf{S}',\mathbf{y}\oplus\mathbf{S}',M')=B(\mathbf{x},\mathbf{y},M')$. So B' witnesses that $F_\oplus$ is even-windable.

($\Leftarrow$) For all sets M of disjoint pairs of $1+J$ define $\nu(M)$ to be $\{S\setminus\{1\}\mid S\in M\}$. Let B be a witness that $F_\oplus$ is windable. For all $\mathbf{x},\mathbf{y}\in\{0,1\}^J$ and all $M\in\mathcal{M}'_{\mathbf{x}\oplus\mathbf{y}}$, define

$$B'(\mathbf{x},\mathbf{y},M) = \sum_{p,q=0}^{1}\ \sum_{M'\colon \nu(M')=M} B((p;\mathbf{x}),(q;\mathbf{y}),M')$$

For all $S\in M=\nu(M')$, let $S'=S$ if $|S|=2$ and $S'=S\cup\{1\}$ otherwise. Then $B(\mathbf{x}\oplus\mathbf{S}',\mathbf{y}\oplus\mathbf{S}',M')=B(\mathbf{x},\mathbf{y},M')$. So B' witnesses that F is windable. □

Lemma 3.16. *Let* φ *be a circuit using only weight-functions that are windable. The weight-function of* φ *is windable.*

Proof. Replace each constraint F_v by $(F_v)_\oplus$, rename the new incidences p_v, $v \in V(\varphi)$, and add a constraint $\mathbf{Even}_{1+P}$ where $P = \{p_v \mid v \in V(\varphi)\}$. This produces a circuit $\varphi_\oplus$ with $[\![\varphi_\oplus]\!] = [\![\varphi]\!]_\oplus$. By Lemmas 3.15, 3.10, and 3.9 we find that $[\![\varphi_\oplus]\!]$ is even-windable. So by Lemma 3.15 again, $[\![\varphi]\!]$ is windable. □

Lemma 3.17. *For any J, the weight-functions $\mathbf{Even}_J$, $\mathbf{Odd}_J$, and $\mathbf{NAE}_J$ are windable.*

Proof. By Lemma 3.8 there is a witness B that $\mathbf{Even}_J$ is even-windable. Extending B by setting $B(\mathbf{x}, \mathbf{y}, M) = 0$ for all $M \in \mathcal{M}'_{\mathbf{x}\oplus\mathbf{y}} \setminus \mathcal{M}_{\mathbf{x}\oplus\mathbf{y}}$ we get a witness B' that $\mathbf{Even}_J$ is windable. Similarly, $\mathbf{Odd}_J$ is even-windable by Lemma 3.8, and it is therefore windable.

For $\mathbf{NAE}_j$, by Lemma 3.15 it suffices to show that the weight-function $(\mathbf{NAE}_J)_\oplus$ is even-windable. Let $I \subseteq 1 + J$, let $\mathbf{p} \in \{0,1\}^I$, let $K = (1 + J) \setminus I$ and let $G \colon \{0,1\}^K \to \mathbb{Q}_{\geq 0}$ be the pinning of $(\mathbf{NAE}_J)_\oplus$ by $\mathbf{p}$. We wish to show that $G\overline{G}$ has a 2-decomposition. If $|K|$ is odd then $G\overline{G}$ is identically zero so has a 2-decomposition. We can therefore assume that $|K|$ is even.

Let $c \in \{0,1\}$ be equal to $|J|$ modulo 2. $(\mathbf{NAE}_J)_\oplus$ is the weight-function corresponding to the relation $\mathrm{Even}_{1+J} \setminus \{(0;\underline{0}), (c;\underline{1})\}$. (Here $\underline{0}$ and $\underline{1}$ are the all-zeros and all-ones configurations of J). We first argue that in all cases, $G\overline{G}$ is either $\mathbf{Even}_K$ or $\mathbf{Odd}_K$, or a flip of $\mathbf{EvenNAE}_K$.

If $\sum_{i \in I} p_i$ is odd, then G takes the value 1 precisely on $\mathrm{Odd}_K \setminus X$ where X consists of at most one configuration $\mathbf{x} \in \mathrm{Odd}_K$. If $X = \emptyset$ then $G\overline{G} = \mathbf{Odd}_K$. If X is a singleton $\{\mathbf{x}\}$ then $G\overline{G}$ is the flip of $\mathbf{EvenNAE}_K$ by $\mathbf{x}$.

If $\sum_{i \in I} p_i$ is even, then G takes the value 1 precisely on $\mathrm{Even}_K \setminus X$ where X consists of at most two configurations in Even_K. If $|X| \leq 1$ we are done by the same argument as the previous paragraph: $G\overline{G}$ is either $\mathbf{Even}_K$ or a flip of $\mathbf{EvenNAE}_K$. If $|X| = 2$ then X consists of two configurations $\mathbf{x}, \mathbf{y}$ with $\mathrm{d}(\mathbf{x}, \mathbf{y}) = |J| + c$. But $|J| + c \leq |K| \leq |J| + 1$, and $|K|$ and $|J| + c$ are both even, so $|K| = |J| + c$. Thus $\mathbf{y} = \overline{\mathbf{x}}$, and again $G\overline{G}$ is a flip of $\mathbf{EvenNAE}_K$.

By Lemma 3.8 the weight-functions $\mathbf{Even}_K$ and $\mathbf{Odd}_K$ have 2-decompositions. So we only need to check the last case where $G\overline{G}$ is a flip, by $\mathbf{z} \in \{0,1\}^K$ say, of $\mathbf{EvenNAE}_K$. Let D be a 2-decomposition of $\mathbf{EvenNAE}_K$ given by Lemma 3.9. Define $D'(\mathbf{x}, M) = D(\mathbf{x} \oplus \mathbf{z}, M)$ for all $\mathbf{x} \in \{0,1\}^K$ and for all partitions M of K into pairs. For all $\mathbf{x} \in \{0,1\}^K$ we have $G\overline{G}(\mathbf{x}) = \mathbf{EvenNAE}_K(\mathbf{x} \oplus \mathbf{z}) = \sum_M D(\mathbf{x} \oplus \mathbf{z}, M) = \sum_M D'(\mathbf{x}, M)$, where M ranges over partitions of K into pairs. So D' is a 2-decomposition of $G\overline{G}$. □

3.5 Strictly terraced functions

3.5.1 Idea

To apply Theorem 3.11 to Holant problems, we want to bound the ratio of the weight of 2-assignments to the weight of 0-assignments in certain closed circuits. Given a closed circuit φ, consider breaking an edge. This gives an circuit φ' with two external edges The sum of $[\![\varphi']\!](1,0)$ and $[\![\varphi']\!](0,1)$ contributes to the weight of 2-assignments in φ, while

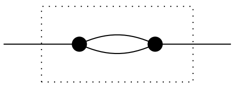

FIGURE 3.6: A circuit with weight-function F with $F(0,0) = 2$ and $F(1,1) = 1$ and $F(0,1) = F(1,0) = 0$. Vertices represent "exact-one" constraints $\{(1,0,0),(0,1,0),(0,0,1)\}$.

the sum of $[\![\varphi']\!](0,0)$ and $[\![\varphi']\!](1,1)$ contributes to the weight of 0-assignments in φ. So one idea (if we are trying to bound the ratio of the weight of 2-assignments of φ to the weight of 0-assignments) is to try to find a bound on ratios like $[\![\varphi']\!](0,1)/[\![\varphi']\!](0,0)$.

It is instructive to consider multiplication of two-by-two matrices. To see the relationship between multiplication of matrices and circuits (in the form of read-twice pps-formulas), for matrices M with rows and columns indexed by $\{0,1\}$, define $F_M \colon \{0,1\}^2 \to \mathbb{Q}_{\geq 0}$ by $F_M(i,j) = M_{i,j}$; then $F_{MN}(i,k) = \sum_j F_M(i,j)F_N(j,k)$.

Matrix multiplication can produce exponentially-large ratios: for any $x, y > 0$, we have

$$\begin{pmatrix} x & y \\ 0 & 1 \end{pmatrix}^n = \begin{pmatrix} x^n & y(x^{n-1} + \cdots + 1) \\ 0 & 1 \end{pmatrix}$$

and $x^n/1$ is exponentially large if $x > 1$.

In fact, the matrix $\left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ corresponds to the circuit depicted in Figure 3.6 using "exact-one" constraints $\{(1,0,0),(0,1,0),(0,0,1)\}$, which can be used to construct counterexamples to the bound (3.1) on nearly perfect matchings [14].

We might guess that exponentially-large ratios can only be produced by matrix multiplication when the zero entry in the matrix is surrounded by values that are different. And indeed this property of being "strictly terraced" turns out to give some control over ratios. For strictly terraced functions, the worst ratio in a weight-function is bounded by the sum of the worst ratios that can be obtained by mixing the individual functions with parity relations.

3.5.2 Definitions

A function $F \colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ is **strictly terraced** if

$$F(\mathbf{x}) = 0 \implies F(\mathbf{x} \oplus \mathbf{e}_i) = F(\mathbf{x} \oplus \mathbf{e}_j) \qquad \text{for all } \mathbf{x} \in \{0,1\}^J \text{ and all } i, j \in J.$$

For all weight-functions F, a **parity-weight-function of** F is a constant multiple of the weight-function of a circuit using one F constraint and such that all other constraints are parity relations. If F is not identically zero, define

$$\theta(F) = \max \left\{ \frac{F'(0)}{F'(1)} \;\middle|\; \begin{array}{c} F' \colon \{0,1\}^1 \to \mathbb{Q}_{\geq 0} \text{ is an arity 1} \\ \text{parity-weight-function of } F \text{ with} \\ F'(1) > 0 \end{array} \right\}.$$

We extend θ to all weight-functions F by setting $\theta(F) = 0$ if F is identically zero.

We can show that θ is well-defined using the following operation. For any circuit φ and any internal edge $e \in E^\varphi$ between incidences $i_u \in J_u$ and $i_v \in J_v$, with $u, v \in V$ (not necessarily distinct), define the **contraction of** φ **by** e to be the circuit φ' obtained by replacing u and v by a vertex w with incidences $(J_u \cup J_v) \setminus \{i_u, i_v\}$ and defining the constraint for this new vertex w to be the weight-function of the circuit with constraints F_u and F_v, edge e, and external edges $(J_u \cup J_v) \setminus \{i_u, i_v\}$.

Lemma 3.18. *Let φ be a connected circuit whose constraints are all parity constraints. The weight-function of φ is a constant multiple of a parity constraint.*

Proof. By induction on the number of edges of φ, it suffices to show that contracting a single edge of φ leaves only parity constraints (up to multiplication by constants).

Consider the case that the edge $\{i, j\}$ goes between distinct vertices, which are equipped with a $\mathbf{Even}_{\{i\}\cup I}$ constraint and a $\mathbf{Even}_{\{j\}\cup J}$ constraint. Then contraction gives a copy of $\mathbf{Even}_{I\cup J}$, because $\mathbf{Even}_{I\cup J}(\mathbf{x}, \mathbf{y}) = \sum_t \mathbf{Even}_{1+I}(t; \mathbf{x})\mathbf{Even}_{1+J}(t; \mathbf{y})$ for all $\mathbf{x} \in \{0,1\}^I$ and all $\mathbf{y} \in \{0,1\}^J$. Similarly $\mathbf{Odd}_{\{i\}\cup I}$ and $\mathbf{Odd}_{\{j\}\cup J}$ produce $\mathbf{Even}_{I\cup J}$, while $\mathbf{Even}_{\{i\}\cup I}$ and $\mathbf{Odd}_{\{j\}\cup J}$ produce $\mathbf{Odd}_{I\cup J}$.

If the edge $\{i, j\}$ is a loop on a vertex with constraint $\mathbf{Even}_{\{i,j\}\cup J}$, contracting $\{i, j\}$ produces $2\mathbf{Even}_J$. Similarly $\mathbf{Odd}_{\{i,j\}\cup J}$ produces $2\mathbf{Odd}_J$. $\square$

Note that contracting an edge does not affect the weight-function of a circuit. By contracting edges between parity relations, the circuits appearing in the definition of a parity-weight-function can be rewritten not to use any edges except external edges and edges incident to the F constraint. For fixed F there are therefore a finite number of equivalence classes of parity-weight-functions $F' \colon \{0,1\} \to \mathbb{Q}_{\geq 0}$ with $F'(1) > 0$, under the equivalence relation of multiplication by constants. Thus the maximum in the definition of $\theta(F)$ is taken over a finite set, which can be seen to be non-empty if F is not identically zero (if $F(\mathbf{x}) > 0$ for some $\mathbf{x} \in \mathrm{Even}_J$ then the function $F' \colon \{0,1\}^1 \to \mathbb{Q}_{\geq 0}$ defined by $F'(t) = \sum_{\mathbf{x}\in\{0,1\}^J} \mathbf{Odd}_1(t)F(\mathbf{x})\mathbf{Even}_J(\mathbf{x})$ satisfies $F'(1) > 0$, and if $F(\mathbf{x}) > 0$ for some $\mathbf{x} \in \mathrm{Odd}_J$ then the function $F' \colon \{0,1\}^1 \to \mathbb{Q}_{\geq 0}$ defined by $F'(t) = \sum_{\mathbf{x}\in\{0,1\}^J} \mathbf{Odd}_1(t)F(\mathbf{x})\mathbf{Odd}_J(\mathbf{x})$ satisfies $F'(1) > 0$).

Note that if G is a parity-weight-function of F, then $\theta(G) \leq \theta(F)$. In particular if G is a pinning of F then $\theta(G) \leq \theta(F)$. Also, since $\mathrm{Odd}_2 = \mathrm{NEQ} = \{(0,1), (1,0)\}$ is a parity relation, it is not important that we took $F'(0)/F'(1)$ rather than $F'(1)/F'(0)$ in the definition of θ.

3.5.3 Examples

A relation $R \subseteq \{0,1\}^J$ is **coindependent** if for all $\mathbf{x} \in \{0,1\}^J \setminus R$ we have $\mathbf{x} \oplus \mathbf{e}_i \in R$ for all indices i. For example, the disequality relation $\{(0,1), (1,0)\}$ is coindependent. Any coindependent relation R gives an example $\mathbf{R}$ of a strictly terraced weight-function.

Lemma 3.19. *For all finite sets J, the functions $\mathbf{Even}_J$, $\mathbf{Odd}_J$ and $\mathbf{NAE}_J$ are strictly terraced. Also, $\theta(\mathbf{Even}_J) = \theta(\mathbf{Odd}_J) = 0$, and $\theta(\mathbf{NAE}_J) \leq 3$.*

Proof. The first statement follows from the fact that the corresponding relations are coindependent. To show $\theta(\mathbf{Even}_J) = \theta(\mathbf{Odd}_J) = 0$, note that by Lemma 3.18 a parity-weight-function of a parity relation must be even.

Now we will show that $\theta(\mathbf{NAE}_J) \leq 3$. Consider a connected circuit φ with one external edge, such that φ uses one $\mathbf{NAE}_J$ constraint, and all other constraints are parity relations, with no internal edges between parity relations (this is without loss of generality, because we can contract any such edge). Assume that $[\![\varphi]\!]\,(0)$ and $[\![\varphi]\!]\,(1)$ are non-zero. We will show that $[\![\varphi]\!]\,(0) \leq 3\,[\![\varphi]\!]\,(1)$.

We can write

$$[\![\varphi]\!]\,(t) = \sum_{\mathbf{x}\in\mathrm{NAE}_J} \mathbf{R}(t;\mathbf{x})$$

where R is an affine subspace of $\mathrm{GF}(2)^{1+J}$. Since $[\![\varphi]\!]\,(0)$ and $[\![\varphi]\!]\,(1)$ are non-zero, the sets $R_0 = \{\mathbf{x} \mid (0;\mathbf{x}) \in R\}$ and $R_1 = \{\mathbf{x} \mid (1;\mathbf{x}) \in R\}$ are non-empty. Since R is an affine subspace, $|R_0| = |R_1|$, so

$$[\![\varphi]\!]\,(0) \leq |R_0| = |R_1| \leq [\![\varphi]\!]\,(1) + 2 \leq 3\,[\![\varphi]\!]\,(1).$$

$\square$

3.5.4 Properties

An important property we will use is that a strictly terraced function F is either identically zero or its support $\{\mathbf{x} \mid F(\mathbf{x}) > 0\}$ is coindependent. (If $F(\mathbf{x}) = 0$ and $F(\mathbf{y}) > 0$ for some $\mathbf{y}$, pick such a $\mathbf{y}$ with $d = \mathrm{d}(\mathbf{x},\mathbf{y})$ minimal. If $d > 1$, there are distinct indices i, j such that $x_i \neq y_i$ and $x_j \neq y_j$, so $F(\mathbf{y}\oplus\mathbf{e}_i) = F(\mathbf{y}\oplus\mathbf{e}_i\oplus\mathbf{e}_j) = 0$ by minimality of $\mathrm{d}(\mathbf{x},\mathbf{y})$, which means F is not strictly terraced: $F(\mathbf{y}\oplus\mathbf{e}_i) = 0$ but $F((\mathbf{y}\oplus\mathbf{e}_i)\oplus\mathbf{e}_i) \neq F((\mathbf{y}\oplus\mathbf{e}_i)\oplus\mathbf{e}_j)$.)

The Cartesian product of coindependent relations is in general not coindependent, for example $\{(0,1),(1,0)\} \times \{(0,1),(1,0)\}$ is not coindependent (set $\mathbf{x} = (0,0,0,0)$ and $i = 1$). Thus the class of strictly terraced functions is not closed under taking weight-functions of disconnected circuits.

Lemma 3.20. *Let φ be a connected circuit using strictly terraced weight-functions. Then $[\![\varphi]\!]$ is strictly terraced.*

Proof. We will argue by induction on the number of internal edges of φ. If there are no internal edges, then φ consists of a single constraint using a strictly terraced function F, and $[\![\varphi]\!] = F$. Otherwise, pick an internal edge e. We wish to argue that the function created by contracting e is strictly terraced. There are two cases.

(i) e is loop on a vertex v.

Let $F\colon \{0,1\}^{2+J} \to \mathbb{Q}_{\geq 0}$ be a copy of F_v, indexed so that the ends of e become enumerated indices. We wish to show that the function $H\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ defined by

$$H(\mathbf{x}) = \sum_{t=0}^{1} F(t,t;\mathbf{x}) \qquad (\mathbf{x} \in \{0,1\}^J)$$

is strictly terraced. Consider $\mathbf{x} \in \{0,1\}^J$ satisfying $H(\mathbf{x}) = 0$ and let $i,j \in J$. Since F is strictly terraced and $F(0,0;\mathbf{x}) = F(1,1;\mathbf{x}) = 0$, we have $F(0,0;\mathbf{x}\oplus\mathbf{e}_i) = F(0,0;\mathbf{x}\oplus\mathbf{e}_j)$ and $F(1,1;\mathbf{x}\oplus\mathbf{e}_i) = F(1,1,\mathbf{x}\oplus\mathbf{e}_j)$. Hence $H(\mathbf{x}\oplus\mathbf{e}_i) = H(\mathbf{x}\oplus\mathbf{e}_j)$.

(ii) e is incident to distinct vertices u and v.

Let $F\colon \{0,1\}^{1+I} \to \mathbb{Q}_{\geq 0}$ and $G\colon \{0,1\}^{1+J} \to \mathbb{Q}_{\geq 0}$ be copies of F_u and F_v respectively, reindexed so that the ends of e become the enumerated indices (and with I and J disjoint). We wish to show that the function $H\colon \{0,1\}^{I\cup J} \to \mathbb{Q}_{\geq 0}$ defined by

$$H(\mathbf{x},\mathbf{y}) = \sum_{t=0}^{1} F(t;\mathbf{x})G(t;\mathbf{y}) \qquad (\mathbf{x} \in \{0,1\}^I, \mathbf{y} \in \{0,1\}^J)$$

is strictly terraced. If F or G is identically zero then H is identically zero and therefore strictly terraced.

Otherwise consider $\mathbf{x} \in \{0,1\}^I$ and $\mathbf{y} \in \{0,1\}^J$ satisfying $H(\mathbf{x},\mathbf{y}) = 0$. Since F and G have coindependent support and $F(0;\mathbf{x})G(0;\mathbf{y}) + F(1;\mathbf{x})G(1;\mathbf{y}) = 0$, there exists $t \in \{0,1\}$ such that $F(t;\mathbf{x}) = G(1-t;\mathbf{y}) = 0$ and $F(1-t;\mathbf{x}), G(t;\mathbf{y}) > 0$. For all $i \in I$ we have

$$H(\mathbf{x}\oplus\mathbf{e}_i,\mathbf{y}) = F(t;\mathbf{x}\oplus\mathbf{e}_i)G(t;\mathbf{y}) = F(1-t;\mathbf{x})G(t;\mathbf{y}).$$

Similarly for $i \in J$ we have

$$H(\mathbf{x},\mathbf{y}\oplus\mathbf{e}_i) = F(1-t;\mathbf{x})G(1-t;\mathbf{y}\oplus\mathbf{e}_i) = F(1-t;\mathbf{x})G(t;\mathbf{y}).$$

Therefore for all $i,j \in I \cup J$ we have $H((\mathbf{x},\mathbf{y})\oplus\mathbf{e}_i) = H((\mathbf{x},\mathbf{y})\oplus\mathbf{e}_j)$. □

The following calculations bound ratios produced by certain circuits.

Lemma 3.21. *Let $F\colon \{0,1\}^{1+J}$ and $G\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$. Define $H(0), H(1)$ by*

$$H(t) = \sum_{\mathbf{x}\in\{0,1\}^J} F(t;\mathbf{x})G(\mathbf{x}).$$

Assume that F and G are strictly terraced and that $H(1) > 0$. Then

$$H(0) \leq (\theta(F) + \theta(G))H(1).$$

Proof. We will use induction on $|J|$. For the base case $J = \emptyset$ we have $H(0) \leq \theta(F)H(1)$ by definition of $\theta(F)$. So assume that J is non-empty.

For each $i \in J$ and each $c \in \{0,1\}$ define $F_{i,c}\colon \{0,1\}^{1+J\setminus\{i\}} \to \mathbb{Q}_{\geq 0}$ to be the pinning of F by taking i to c, and similarly define $G_{i,c}\colon \{0,1\}^{1+J\setminus\{i\}} \to \mathbb{Q}_{\geq 0}$ to be the pinning of g by taking i to c, and define

$$H_{i,c}(t) = \sum_{\mathbf{x}\in\{0,1\}^{J\setminus\{i\}}} F_{i,c}(t;\mathbf{x})G_{i,c}(\mathbf{x}) \qquad (t \in \{0,1\}).$$

Since pinnings are parity-weight-functions, $\theta(F_{i,c}) \leq \theta(F)$ and $\theta(G_{i,c}) \leq \theta(G)$. If there exists $i \in J$ such that $H_{i,0}(1)$ and $H_{i,1}(1)$ are non-zero, then by the induction hypothesis we have

$$H(0) = H_{i,0}(0) + H_{i,1}(0) \leq (\theta(F)+\theta(G))(H_{i,0}(1)+H_{i,1}(1)) = (\theta(F)+\theta(G))H(1)$$

as required. Otherwise, taking a choice for each i, we may assume that there exists $y \in \{0,1\}^J$ such that for all $i \in J$ we have $H_{i,1-y_i}(1) = 0$. So for each $i \in I$ the sets $R = \{\mathbf{x} \mid F_{i,1-y_i}(1;\mathbf{x}) > 0\}$ and $S = \{\mathbf{x} \mid G_{i,1-y_i}(\mathbf{x}) > 0\}$ are disjoint. R and S are pinnings of coindependent relations, so they are coindependent. For all $\mathbf{x} \in R$ we have $\mathbf{x} \notin S$, so $\mathbf{x}\oplus\mathbf{e}_i \in S$ for any i, and $\mathbf{x}\oplus\mathbf{e}_i \notin R$. Repeating this, we find that R consists of the configurations at even distance from $\mathbf{x}$, and S consists of the configurations at odd distance from $\mathbf{x}$. In other words, for each $i \in J$ there exists $c_i \in \{0,1\}$ such that

$$\left.\begin{aligned} F(1;\mathbf{x}) > 0 &\iff c_i + \sum_{j\in J} x_j \text{ is even, and} \\ G(\mathbf{x}) > 0 &\iff c_i + \sum_{j\in J} x_j \text{ is odd.} \end{aligned}\right. \qquad (\mathbf{x} \in \{0,1\}^J, x_i \neq y_i) \tag{3.3}$$

For any $i,j \in J$ there is some choice of $\mathbf{x} \in \{0,1\}^J$ with $x_i \neq y_i$ and $x_j \neq y_j$, so $c_i = c_j$. Thus there is a single choice of c such that (3.3) holds for all i taking $c_i = c$:

$$\left.\begin{aligned} F(1;\mathbf{x}) > 0 &\iff c + \sum_{j\in J} x_j \text{ is even, and} \\ G(\mathbf{x}) > 0 &\iff c + \sum_{j\in J} x_j \text{ is odd.} \end{aligned}\right. \qquad (\mathbf{x} \in \{0,1\}^J \setminus \{\mathbf{y}\}) \tag{3.4}$$

For any $\mathbf{x} \in \{0,1\}^J$ with $c + \sum_{i\in J} x_i$ even, and any distinct $i,j \in J$, we have $F(1;\mathbf{x}) = F(1;\mathbf{x}\oplus\mathbf{e}_i\oplus\mathbf{e}_j)$ because F is strictly terraced and either $F(1;\mathbf{x}\oplus\mathbf{e}_i)$ or $F(1;\mathbf{x}\oplus\mathbf{e}_j)$ is zero. Similarly for any $\mathbf{x} \in \{0,1\}^J$ with $c+\sum_{i\in J} x_i$ odd, and any distinct $i,j \in J$, we have $G(\mathbf{x}) = G(\mathbf{x}\oplus\mathbf{e}_i\oplus\mathbf{e}_j)$. This implies that there are constants $\lambda, \mu > 0$ such that

$$\left.\begin{aligned} F(1;\mathbf{x}) = \lambda &\text{ if } c + \sum_{i\in J} x_i \text{ is even, and} \\ G(\mathbf{x}) = \mu &\text{ if } c + \sum_{i\in J} x_i \text{ is odd.} \end{aligned}\right. \qquad (\mathbf{x} \in \{0,1\}^J). \tag{3.5}$$

If $c + \sum_{i\in J} y_i$ is odd then by (3.5) and (3.4), G is $\mu\mathbf{Even}_J$ (if $c = 1$) or $\mu\mathbf{Odd}_J$ (if $c = 0$). In this case G is a constant multiple of a parity relation. Considering H as the weight-function of a circuit with constraints F and G, we get $H(0) \leq \theta(F)H(1)$ by definition of θ. We may therefore assume that $c + \sum_{i\in J} y_i$ is even.

Define $F'(0), F'(1), G'(0), G'(1)$ by

$$\begin{aligned} F'(t) &= F(t;\mathbf{y}) \\ G'(t) &= \sum_{\mathbf{x}\in\{0,1\}^J} \mathbf{Odd}_{2+J}(t,c;\mathbf{x})G(\mathbf{x}) \end{aligned}$$

so

$$\begin{aligned} F'(0) &= F(0;\mathbf{y}) \\ F'(1) &= F(1;\mathbf{y}) = \lambda \\ G'(0) &= \mu 2^{|J|-1} \\ G'(1) &= G(\mathbf{y}) \end{aligned}$$

Since F' is a parity-weight-function of F we have $F(0;\mathbf{y})/\lambda \leq \theta(F)$. Since G' is a parity-weight-function of G we have $\mu 2^{|J|-1}/G(\mathbf{y}) \leq \theta(G)$. For all $\mathbf{x} \in \{0,1\}^J$ with $c + \sum_{i\in J} x_i$ odd, we have $F(1;\mathbf{x}) = 0$ and $F(1;\mathbf{x}\oplus\mathbf{e}_i) = \lambda$ (for any $i \in J$) and therefore $F(0;\mathbf{x}) = \lambda$ because F is strictly terraced. So

$$\frac{H(0)}{H(1)} = \frac{F(0;\mathbf{y})G(\mathbf{y}) + \lambda\mu 2^{|J|-1}}{\lambda G(\mathbf{y})} \leq \theta(F) + \theta(G).$$

□

Lemma 3.22. *Let φ be a circuit using strictly terraced weight-functions. Then*

$$\theta(\llbracket\varphi\rrbracket) \leq \sum_{v\in V^\varphi} \theta(F_v^\varphi).$$

Proof. We will argue by induction on the number k of constraints of φ that are not parity relations. The cases $k = 0$ and $k = 1$ follow from the definition of θ. Components of a circuit not connected to the external edges simply contribute a constant factor to the weight-function. And $\theta(F) = 0$ for parity-weight-functions F (Lemma 3.19). So for a given k, it suffices to show that $\llbracket\varphi(0)\rrbracket / \llbracket\varphi(1)\rrbracket \leq \sum_{v\in V^\varphi} \theta(F_v^\varphi)$ whenever:

- φ is a connected circuit with one external edge, with $\llbracket\varphi(1)\rrbracket > 0$, and
- φ uses strictly terraced weight-functions, at most k of which are not parity relations.

For the $k = 2$ case, if there is a loop on a vertex v, contract it. This changes the weight-function F_v, but the resulting weight-function is a parity-weight-function of F_v, so this process does not increase $\sum_{v\in V} \theta(F_v)$. And F_v is still strictly terraced by Lemma 3.20. Similarly, if there is an edge incident to distinct vertices u, v where F_u is a parity constraint, contract that edge. The weight-function F introduced by the contraction is a parity-weight-function of F_v, so this process does not increase $\sum_{v\in V} \theta(F_v)$. And again, F is strictly terraced by Lemma 3.20. Repeating this process we end up with a circuit

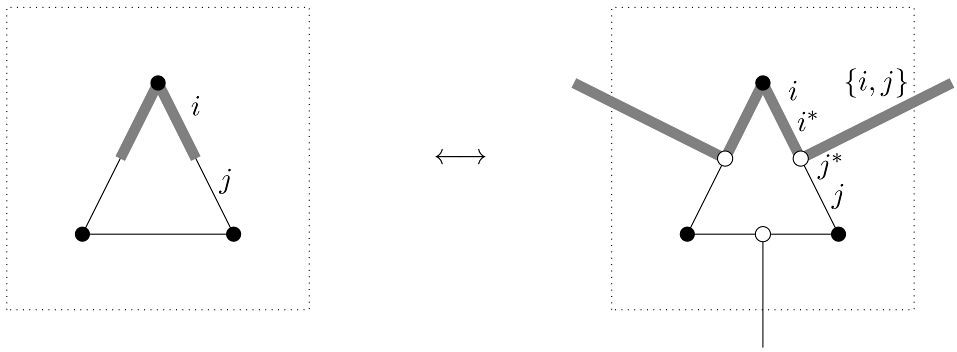

FIGURE 3.7: An illustration of the correspondence between arbitrary configurations in a closed circuit and assignments in a modified circuit. Solid circles are arbitrary constraints, empty circles are copies of Even_3. Thin black lines are incidences given the value 0, thick grey lines are incidences given the value 1.

with at most two vertices. If there is only one vertex we can appeal to the $k \le 1$ case, and otherwise we are done by Lemma 3.21.

For $k > 2$, contract any internal edge. From the $k \le 2$ case we know that $\sum_{v\in V} \theta(F_v)$ has not increased. This process does not change the weight-function of φ, and by Lemma 3.20 the constraint function introduced by the contraction is strictly terraced. $\square$

Lemma 3.23. *Let φ be a closed circuit using strictly terraced constraints, and assume that $Z_0(\varphi) > 0$. Then*

$$\frac{Z_2(\varphi)}{Z_0(\varphi)} \le \frac{1}{2}|E^\varphi|^2 \max\left(1, \sum_{v\in V^\varphi} \theta(F_v^\varphi)\right)^2. \tag{3.6}$$

Proof. We will consider a circuit ψ obtained by attaching $\mathbf{Even}_3$ to edges of φ as illustrated in Figure 3.7. In words: let J^* be a disjoint copy of J, consisting of an element i^* for each $i \in J$. Define ψ to have incidences $E \cup J \cup J^*$, external edges E, edges $\{i, i^*\}$ for each $i \in J$, vertex set $V \cup E$, the same constraints at each $v \in V$, and $F^\psi_{\{i,j\}} = \mathbf{Even}_{\{i^*,j^*,\{i,j\}\}}$ for all $\{i, j\} \in E$.

$Z_k(\varphi)$ is the sum of $[\![\psi]\!](\mathbf{x})$ over configurations $\mathbf{x}$ of E^φ with $\sum_{e\in E^\varphi} x_e = k$. By pinning, θ bounds the ratio $F(\mathbf{x} \oplus \mathbf{e}_i)/F(\mathbf{x})$ between the weights of neighbouring configurations of non-zero weight. Letting $\underline{0}$ denote the all-zeros vector, for all $i \ne j$ such that $[\![\psi]\!](\mathbf{e}_i) \ne 0$,

$$[\![\psi]\!](\mathbf{e}_i + \mathbf{e}_j) \le \theta(\psi)\,[\![\psi]\!](\mathbf{e}_i) \le \theta(\psi)^2\,[\![\psi]\!](\underline{0}).$$

If $[\![\psi]\!](\mathbf{e}_i) = 0$ we have $[\![\psi]\!](\mathbf{e}_i + \mathbf{e}_j) = [\![\psi]\!](\underline{0})$ because $[\![\psi]\!]$ is strictly terraced. Thus

$$Z_2(\varphi) \le Z_0(\varphi)\binom{|E^\varphi|}{2}\max(1, \theta(\psi))^2.$$

The result follows by applying Lemmas 3.19 and 3.22. $\square$

3.6 Proofs of Theorem 3.1 and Theorem 3.4

For the algorithmic results we will use a reduction of counting to sampling. This is a standard argument, which has been formalised in particular in [73, Theorem 6.4]. The important hypothesis is called self-reducibility and means that a problem can be expressed as a combination of not too many smaller subproblems. Rather than check this hypothesis (which can require a problem to be encoded in an unusual way), we will now describe the argument for the particular case of Holant problems, illustrating exactly what type of samples are needed.

The aim is to approximate $Z_0(\varphi)$ for a closed circuit φ with $Z_0(\varphi) > 0$. We will consider the unweighted case, by assuming that wt_φ takes values in $\{0, 1\}$, so $Z_0(\varphi) = |\Omega_0|$. First, the set of variables J is ordered, say as $1, \ldots, n$. For each $k = 1, \ldots, n$ in turn, the procedure takes a large number t of samples $\mathbf{z}^1, \ldots, \mathbf{z}^t$ sampled "almost uniformly" from the set $E_{k-1} = \{\mathbf{x} \in \Omega_0 \mid (x_1, \ldots, x_{k-1}) = (y_1, \ldots, y_{k-1})\}$ ($y_1, \ldots, y_{k-1}$ will have been chosen in steps $1, \ldots, k-1$). This is the important sampling step we need to be able to implement. Then y_k is defined to be the most common value in the list $z_k^1, \ldots, z_k^t$, and α_k is defined to be the proportion of these values equal to y_k. The output is $1/\alpha_1 \ldots \alpha_n$.

This approximation relies on the telescoping product

$$\frac{1}{Z_0(\varphi)} = \frac{|E_1|}{|E_0|} \frac{|E_2|}{|E_1|} \cdots \frac{|E_n|}{|E_{n-1}|} \approx \alpha_1 \cdots \alpha_n$$

where $E_n = \{\mathbf{y}\}$.

The calculations in [73, Theorem 6.4] show that this approximation gives an FPRAS as long as for any $\varepsilon > 0$ we can take a sample $\mathbf{z} \in E_k$ satisfying $(1+\varepsilon)^{-1}|E_k|^{-1} \leq \Pr(\mathbf{x} = \mathbf{z}) \leq (1+\varepsilon)|E_k|^{-1}$ for all $\mathbf{x} \in E_k$, in time polynomial in the size of φ and in $\log(1/\varepsilon)$.

The important requirement is that we can sample to within a small error from the uniform distribution on the sets E_k – holding some variables constant. This ability to hold variables constant is a type of self-reducibility. For an FPRAS, we also need to be able to test whether there are any satisfying assignments at all, that is, $Z_0(\varphi) > 0$.

Theorem 3.1. #ParityNAE has an FPRAS.

Proof. We are given a labelled graph, which is naturally a closed circuit φ using constraints of the form $\mathbf{Even}_J$, $\mathbf{Odd}_J$, and $\mathbf{NAE}_J$.

As described above, it suffices to show that we can test $Z_0(\varphi)$, and that we can sample uniformly from the subsets of Ω_0 where some variables are kept constant. But we can test whether $Z_0(\varphi) > 0$ in polynomial time by Cornuéjols' algorithm for the general factor problem [39]. And degree-1 parity relations can be used to fix variables to take a particular value, so it suffices to show how to sample uniformly from Ω_0 (without holding any variables constant).

We will use the near-assignments chain to sample from assignments of φ. Define $F\colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ by

$$F(\mathbf{x}) = \begin{cases} \mathrm{wt}_\varphi(\mathbf{x}), & \text{if } \sum_{i\in J} x_i \text{ is even} \\ 0 & \text{otherwise.} \end{cases}$$

(In fact, $\mathrm{wt}_\varphi(\mathbf{x})$ is always 0 or 1.)

By Lemma 3.17, all the constraints of φ are windable. By Lemma 3.16, wt_φ is windable. By Lemma 3.15, $(\mathrm{wt}_\varphi)_\oplus$ is even-windable. But F is a pinning of $(\mathrm{wt}_\varphi)_\oplus$ (setting the parity bit to zero). A pinning of an even-windable function is even-windable - this is immediate from the characterisation in terms of 2-decompositions given in Section 3.3.3.

We will use the notation π, Ω, Ω_0 from Theorem 3.11, for the near-assignment chain on the pair (F, E^φ).

Recall from Lemma 3.19 that $\theta(\mathbf{NAE}_J) \leq 3$ and $\theta(\mathbf{Even}_J) = \theta(\mathbf{Odd}_J) = 0$, and that all these weight-functions are strictly terraced. Let $R = 3|V|^2|E|^2$; by Lemma 3.23 we have $1/R \leq Z_0(\varphi)/Z_2(\varphi) \leq Z_0(\varphi)/(Z_0(\varphi) + Z_2(\varphi)) = \pi(\Omega_0)$.

By Cornuéjols' algorithm, mentioned above, we get an assignment $\mathbf{y}$ with $\mathrm{wt}_\varphi(\mathbf{y}) > 0$ and in particular $\pi(\mathbf{y}) \geq 2^{-|E|}$. Let $\delta > 0$ be an error parameter, which will be specified later. Applying Theorem 3.11, by simulating the near-assignments Markov chain of (F, E) for $t \geq (2|E|)^4 R^2(\log \frac{2R}{\delta} + |E| \log 2)$ steps we can take a sample $\mathbf{z}$ from near-assignments of φ such that

$$\frac{1}{2}\sum_{\mathbf{x}\in\Omega} |\Pr(\mathbf{x} = \mathbf{z}) - \pi(\mathbf{x})| \leq \delta/2R$$

Thus

$$\frac{1}{2}\sum_{\mathbf{x}\in\Omega_0} |\Pr(\mathbf{x} = \mathbf{z} | \mathbf{z} \in \Omega_0) - F(\mathbf{x})/Z_0(\varphi)| \leq \delta/2$$

We sample from Ω_0 by rejection sampling: run the simulation at least $2R\log\frac{2}{\delta}$ times and return the first sample in Ω_0. The probability that this fails is small (at most $(1-\frac{1}{2}\pi(\Omega_0))^{2R\log\frac{2}{\delta}} \leq \delta/2$). To get the condition that $(1+\varepsilon)^{-1}|\Omega_0|^{-1} \leq \Pr(\mathbf{x} = \mathbf{z}) \leq (1+\varepsilon)|\Omega_0|^{-1}$ for all $\mathbf{x} \in \Omega_0$, as required by [73, Theorem 6.4], we can take $\delta = \varepsilon/2^{|E|+1}$. $\square$

The reduction of counting to sampling in [73, Theorem 6.4] is stated in the setting of unweighted counting and uniform sampling. But the results generalise to weighted sums and non-uniform distributions. Instead of the uniform distribution on the sets E_k, we need to be able to sample to within a small error from the distribution on E_k proportional to wt_φ. The FPRAS described above would output $\mathrm{wt}_\varphi(\mathbf{z})/\alpha_1 \dots \alpha_n$ instead of $1/\alpha_1 \dots \alpha_n$. The sampling condition becomes $(1+\varepsilon)^{-1}\pi_k(\mathbf{x}) \leq \Pr(\mathbf{x} = \mathbf{z}) \leq (1+\varepsilon)\pi_k(\mathbf{x})$ where $\pi_k(\mathbf{x}) = \mathrm{wt}_\varphi(\mathbf{x})/\sum_{\mathbf{z}\in E_k} \mathrm{wt}_\varphi(\mathbf{z})$.

Theorem 3.4. Let $\mathcal{F}$ be the class of strictly terraced windable functions. Then

- $\mathcal{F}$ is closed under taking weight-functions of connected circuits
- $\mathcal{F}$ contains $\mathbf{Even}_k$, $\mathbf{Odd}_k$, and $\mathbf{NAE}_k$ for all $k \geq 1$
- for all finite subsets $\mathcal{F}' \subset \mathcal{F}$ there is an FPRAS for $\mathsf{Holant}(\mathcal{F}')$

Proof. The first statement is Lemma 3.16. The second statement is Lemma 3.17.

For the third statement, given $\mathcal{F}'$, let $\mathcal{F}'' = \mathcal{F}' \cup \{\mathbf{Even}_1, \mathbf{Odd}_1\}$. For the decision problem we can use Feder's algorithm for coindependent relations [52, Theorem 4]. We can use $\mathbf{Even}_1$ and $\mathbf{Odd}_1$ to fix the value a variable takes. Otherwise the argument proceeds as in the previous proof except taking R to be $|E|^2|V|^2 \max_{F\in\mathcal{F}'} \theta(F)$, and taking δ slightly smaller, for example $\delta = \varepsilon/2^{|E|+1}(M/m)^{|V|}$ where M is the maximum value taken by any function in $\mathcal{F}$ and m is the minimum non-zero value taken by any function in $\mathcal{F}$. We find that $\mathsf{Holant}(\mathcal{F}'')$, and therefore $\mathsf{Holant}(\mathcal{F}')$, has an FPRAS. □

3.7 Matchings circuits

Define a **matchings circuit** G to be a graph fragment equipped with:

- a non-negative rational edge-weight $w(e)$ for each internal edge e
- a non-negative rational fugacity $\lambda(v)$ for each vertex v

Note that in this definition the external edges are not given weights.

For any set of edges $F \subseteq A \cup E$, let $\deg_F(v)$ denote the number of edges in F incident to the vertex v. The weight of F is:

$$\mathrm{wt}_G(F) = \begin{cases} 0 & \text{if } \deg_F(v) \geq 2 \text{ for some vertex } v \\ \left(\prod_{\deg_F(v)=0} \lambda(v)\right)\left(\prod_{e\in F} w(e)\right) & \text{otherwise.} \end{cases}$$

The weight-function of G is the function $[\![G]\!] : \{0,1\}^A \to \mathbb{Q}_{\geq 0}$ where A is the set of external edges and

$$[\![G]\!]\,(\mathbf{x}) = \sum_{\substack{F\subseteq E \\ F\cap A = \{e\in A \mid x_e = 1\}}} \mathrm{wt}_\varphi(F).$$

As with circuits, if $F = [\![G]\!]$ we will say the F **has** the matchings circuit G.

For all $w \geq 0$ define $\mathrm{Edge}^w \colon \{0,1\}^2 \to \mathbb{Q}_{\geq 0}$ by

$$\mathrm{Edge}^w(i,j) = \begin{pmatrix} 1 & 0 \\ 0 & w \end{pmatrix}_{i,j}$$

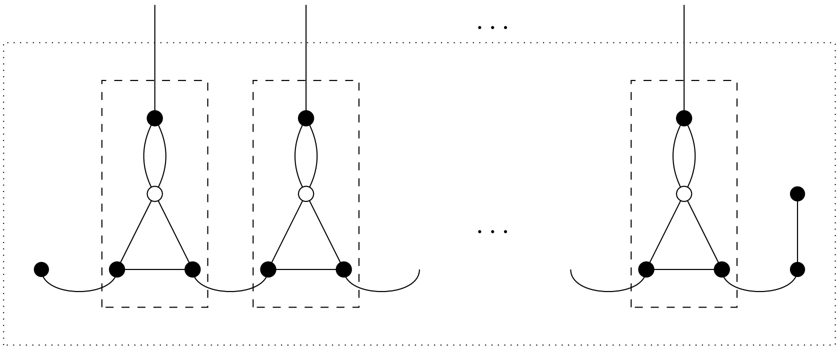

FIGURE 3.8: $2^{k-1}\mathrm{OR}_k$ matchings circuit. Hollow circles are vertices with fugacity 1. All other vertices have fugacity 0, and all edges have edge-weight 1.

where the matrix rows and columns are indexed by $\{0,1\}$. For all $\lambda \geq 0$ and all finite sets J define $\mathrm{Fugacity}_J^\lambda \colon \{0,1\}^J \to \mathbb{Q}_{\geq 0}$ by

$$\mathrm{Fugacity}_J^\lambda(\mathbf{x}) = \begin{cases} \lambda & \text{if } \sum_{i\in J} x_i = 0 \\ 1 & \text{if } \sum_{i\in J} x_i = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Given G, define a circuit by equipping each vertex v with the function $\mathrm{Fugacity}_{J_v}^{\lambda(v)}$, then subdividing each edge e and equipping the new vertex with the function $\mathrm{Edge}^{w(e)}$. The circuit clearly has the same weight-function as the matchings circuit. So matchings circuits are just a special type of circuit. We will use the same notation and terminology.

3.7.1 Example

Proposition 3.24. *For all finite sets J define $\mathrm{OR}_J = \{\mathbf{x} \in \{0,1\}^J \mid \sum_{i\in J} x_i > 0\}$. Then $\mathbf{OR}_J$ has a matchings circuit.*

Proof. We may assume $J = \{1,\ldots,k\}$. The matchings circuit is illustrated in Figure 3.8.

Define $F \colon \{0,1\}^3 \to \mathbb{Q}_{\geq 0}$ by

$$F(i,0,j) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}_{i,j}$$
$$F(i,1,j) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}_{i,j}$$

(with rows and columns indexed from zero.) Each of the smaller boxes shown in Figure 3.8 have the weight-function F (where external edges are numbered from left to right).

For all $x_1, \ldots, x_k \in \{0,1\}$,

$$\begin{aligned} \llbracket G \rrbracket (x_1, \ldots, x_k) &= \sum_{y_1,\cdots,y_{k-1}} F(1, x_1, y_1) F(y_1, x_2, y_2) \ldots F(y_{k-1}, x_k, 0) \\ &= \left(\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}^{k - x_1 + \cdots - x_k} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}^{x_1 + \cdots + x_k} \right)_{1,0} \\ &= 2^{k-1} \mathbf{OR}_k (x_1, \ldots, x_k). \end{aligned}$$

So the weight-function of G is $2^{k-1}\mathbf{OR}_k$. To deal with the scalar multiple, add an isolated vertex with fugacity $1/2^{k-1}$. □

In particular, let $k \geq 1$ be odd. Then $(\mathbf{OR}_k)_\oplus$ is a copy of $\mathbf{EvenOR}_{k+1}$ where EvenOR_{k+1} is defined as $\mathrm{Even}_{k+1} \cap \mathrm{OR}_{k+1}$. By Lemma 3.15, $\mathbf{EvenOR}_{k+1}$ is even-windable. Thus $\mathbf{EvenOR}_{k+1}\overline{\mathbf{EvenOR}_{k+1}} = \mathbf{EvenNAE}_{k+1}$ has a 2-decomposition. This gives an alternate proof of Lemma 3.9 which, via Lemma 3.17, shows that $\mathbf{NAE}_J$ is windable for all finite sets J. But this argument does not seem to show that $\mathbf{NAE}_J$ has a matchings circuit.

3.7.2 Approximate counting

Define

Name. #PM
Instance. A simple graph G
Output. The number of perfect matchings in G

The aim of this section is to establish Theorem 3.5, that $\mathsf{Holant}(\mathcal{F}) \leq_{\mathrm{AP}} \#\mathsf{PM}$ for any finite set $\mathcal{F}$ of weight-functions of matchings circuits, showing that matchings circuits are a natural choice of circuit for #PM. We will reduce via the following problem.

Name. #FugacityWeightedPM
Instance. A closed matchings circuit φ where fugacities and edge-weights are given as ratios of non-negative integers specified in binary
Output. $Z_0(\varphi)$

The fugacities and edge-weights can both be simulated using matchings circuits. A similar reduction appears in [59].

Lemma 3.25. *There is a polynomial-time algorithm which, given non-negative integers p, q specified in binary, outputs a matchings circuit $G_{p,q}$ whose fugacities are all* 0 *and whose edge-weights are all* 1*, and with two external edges such that*

$$\llbracket G_{p,q} \rrbracket (i,j) = \begin{pmatrix} p & 0 \\ 0 & q \end{pmatrix}_{i,j} \qquad \textit{for all } i,j \in \{0,1\}.$$

where we consider the rows and columns of the matrix to be indexed from zero.

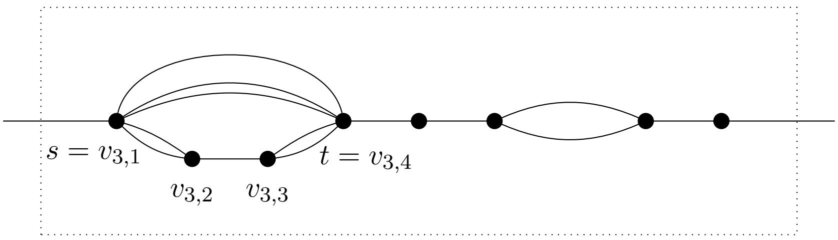

FIGURE 3.9: $G_{7,2}$, with one path in the copy of $G_{7,1}$ labelled. All fugacities are 0, all edge-weights are 1.

Proof. See Figure 3.9.

For all $p \geq 0$ there is a unique binary expansion $p = 2^{n_1} + \cdots + 2^{n_k}$, with $0 \leq n_1 < \cdots < n_k$. Define $G_{p,1}$ in the following way. Take two vertices s and t, each with one external edge. For each $1 \leq i \leq k$, if $n_i = 0$ add an edge between s and t, and otherwise add a path between s and t of length $2n_i - 1$, which we can denote $s = v_{i,1}, v_{i,2}, \ldots, v_{i,2n_i} = t$, and add a parallel edge in the odd positions: between $v_{i,2j-1}$ and $v_{i,2j}$ for each $1 \leq j \leq n_i$.

There is a unique perfect matching of $G_{p,1}$ that includes the external edges: it uses the edges in even position in each path, $v_{i,2j}v_{i,2j+1}$ for all $1 \leq i \leq k$ and all $1 \leq j < n_i$. The perfect matchings of $G_{p,1}$ that do not include the external edges are determined by a choice of $1 \leq i \leq k$ such that the i'th path uses edges in odd positions, and a choice of edge in each of the n_i odd positions in this path; there are 2^{n_i} choices for each i. So $G_{p,1}$ has the correct weight-function: $[\![G_{p,1}]\!]\,(i,j) = \left(\begin{smallmatrix} p & 0 \\ 0 & 1 \end{smallmatrix}\right)_{i,j}$.

Define H to be the circuit consisting of one vertex with fugacity zero, with two external edges. For $q \neq 1$ define $G_{p,q}$ to be serial composition of copies of $G_{p,1}$, H, $G_{q,1}$, and H, that is, we identify the second external edge of the i'th circuit with the first external edge of the $(i+1)$'th, for $i = 1, 2, 3$. The weight-function of $G_{p,q}$ is then given by the matrix

$$\begin{pmatrix} p & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} q & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} p & 0 \\ 0 & q \end{pmatrix}.$$

□

Lemma 3.26. *Given a matchings circuit G for a weight-function F, we can efficiently construct a matchings circuit G' for $F_\oplus$ (defined in Section 3.4) in which every vertex has fugacity zero. Conversely, given a matchings circuit G for $F_\oplus$, we can efficiently construct a matchings circuit G' for F.*

Proof. For the first statement, pick an enumeration $v_1, \ldots, v_n$ of the vertices of G. Form G' as follows. For each $1 \leq i \leq n$, add vertices a_i, b_i, c_i, edges a_ib_i,a_ic_i,b_ic_i with edge-weight 1, add an edge v_ia_i with weight $\lambda(v_i)$, and if $i < n$ add an edge c_ib_{i+1} with edge-weight 1. Set all the fugacities to zero and add an external edge at b_1. See Figure 3.10. Consider a matching $M \subseteq E$ of G. We will argue that there is a unique way to extend M to a perfect matching M' of G'.

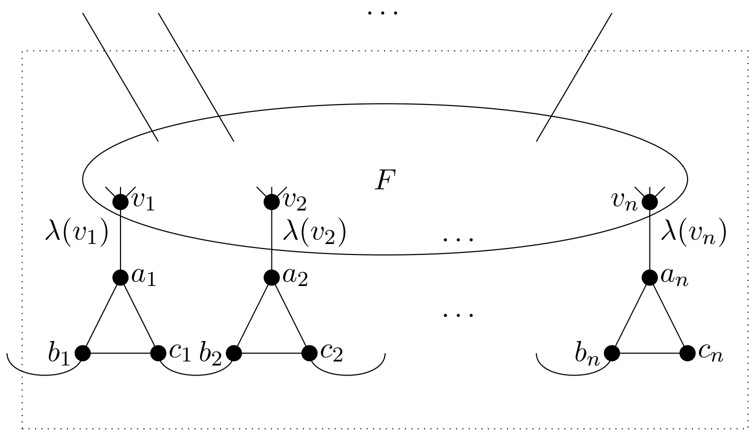

FIGURE 3.10: Illustration of a matching circuit for $F_\oplus$ built from a matchings circuit for F, as described in Lemma 3.26.

Let $U = \{i \mid \deg_M(v_i) = 0\}$ be the indices of unmatched vertices. Let $M_1 = \{a_i v_i \mid i \in U\}$. Note that the extension M' must include M_1, and if $i \notin U$ we cannot have $b_i c_i \in M'$. Consider the following subset P of external and internal edges: the external edge at b_1, edges $b_i c_i$ for all $i \in U$, and the edges $b_i a_i$ and $a_i c_i$ for all $i \notin U$. So P is a path, except that at one endpoint, P has an external edge b_1. Observe that there is a unique choice of perfect matching $M_2 \subseteq P$ along this path: the odd-numbered edges starting from the end of P not incident to the external edge b_1. (If P has an odd number of vertices then we get $b_1 \in M_2$, and otherwise $b_1 \notin M_2$.) Define $M' = M \cup M_1 \cup M_2$. Any extension of M to a perfect matching of G' would have to include M_1, and hence M_2, and so the extension is unique.

This gives a weight-preserving bijection between matchings M of G and perfect matchings M' of G'. Since G' has an even number of vertices, the sets M' must include an even number of external edges. Thus $[\![G']\!] = F_\oplus$.

The converse is easy: given a matchings circuit G for $F_\oplus$, add a vertex of fugacity 1 to the external edge 1 to get a matchings circuit for F. □

Lemma 3.27. #FugacityWeightedPM $\leq_{\mathrm{AP}}$ #PM

Proof. Given a matchings circuit G_1 with no external edges, we will construct a simple graph G with $C\,[\![G_1]\!]$ perfect matchings where C is an easily computed positive integer.

By Lemma 3.26 we get a matchings circuit G_2 such that $[\![G_2]\!] = [\![G_1]\!]_\oplus$. Deleting the external edge, we get a circuit G_3 with $[\![G_3]\!] = [\![G_1]\!]$. At each edge e of G_3, we have integers p_e, q_e such that the weight of e is p_e/q_e. Insert a copy of the circuit $G_{p,q}$ given by Lemma 3.25; this produces a circuit G_4 whose weight-function is $C\,[\![G_3]\!]$ where $C = \prod_{e \in E^{G_3}} q_e$, and where all fugacities are 0 and all edge-weights are 1. Forgetting the fugacities and edge-weights we get a multigraph with $C\,[\![G_3]\!]$ perfect matchings. To construct G, delete any loops and subdivide each edge into a path of length 3; this does not affect the number of perfect matchings. □

Theorem 3.5. If $\mathcal{F}$ is a finite set of weight-functions that have matchings circuits, then $\mathsf{Holant}(\mathcal{F}) \leq_{\mathrm{AP}} \#\mathsf{PM}$.

Proof. Pick a choice of matchings circuit G_F for each $F \in \mathcal{F}$. Given an instance ψ of $\mathsf{Holant}(\mathcal{F})$, for each vertex v the function F_v is a copy of some $F \in \mathcal{F}$; we can substitute G_F into ψ at v. This process gives a matchings circuit G' with the same weight-function as ψ. We can then appeal to Lemma 3.27. □

3.7.3 Expressive power

Lemma 3.28. *The weight-function of any matchings circuit is windable.*

Proof. By Lemma 3.26 and Lemma 3.15 it suffices to show that the weight-function of any matchings circuit where all fugacities are zero is even-windable. For all $w \geq 0$, Lemma 3.7 implies that Edge^w is even-windable. For all $\lambda \geq 0$ and all finite sets J, consider a pinning $G\colon \{0,1\}^I \to \mathbb{Q}_{\geq 0}$ of $\mathrm{Fugacity}_J^\lambda$. If $(G\overline{G})(\mathbf{x}) > 0$ for some $\mathbf{x}$ then $\sum_{i\in I} x_i$ and $\sum_{i\in I}(1-x_i)$ are at most 1, so $|I| \leq 2$. Thus $G\overline{G}$ has a 2-decomposition as in Lemma 3.7. □

To give circuits for low-arity functions we will apply linear programming duality in the form of Farkas' lemma. For a very short proof of Farkas' lemma, as well as a statement explicitly allowing a general ordered field, see [4]. For all $\mathbf{x}, \boldsymbol{\varphi} \in \mathbb{Q}^d$ denote the dot product $x_1\varphi_1 + \cdots + x_d\varphi_d$ by $\mathbf{x} \cdot \boldsymbol{\varphi}$.

Lemma 3.29. *Let $\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{y} \in \mathbb{Q}^d$. The following are equivalent:*

- $\mathbf{y} = c_1\mathbf{x}_1 + \cdots + c_k\mathbf{x}_k$ *for some* $c_1, \ldots, c_k \in \mathbb{Q}_{\geq 0}$
- $\mathbf{y} \cdot \boldsymbol{\varphi} \geq 0$ *for all* $\boldsymbol{\varphi} \in \mathbb{Q}^d$ *that satisfy* $\mathbf{x}_1 \cdot \boldsymbol{\varphi}, \ldots, \mathbf{x}_k \cdot \boldsymbol{\varphi} \geq 0$

Lemma 3.30. *Let $F\colon \{0,1\}^4 \to \mathbb{Q}_{\geq 0}$. Assume that $F(\overline{\mathbf{e}_1})$, $F(\overline{\mathbf{e}_2})$,$F(\overline{\mathbf{e}_3})$,$F(\overline{\mathbf{e}_4})$ are not all zero, and that $F(x_1,x_2,x_3,x_4) = 0$ whenever $x_1+x_2+x_3+x_4$ is even, and that for all $x_1,x_2,x_3,x_4 \in \{0,1\}$ we have*

$$\begin{array}{rl} & F(x_1,x_2,x_3,x_4)F(1-x_1,1-x_2,1-x_3,1-x_4) \\ \leq & F(x_1,x_2,1-x_3,1-x_4)F(1-x_1,1-x_2,x_3,x_4) \\ + & F(x_1,1-x_2,x_3,1-x_4)F(1-x_1,x_2,1-x_3,x_4) \\ + & F(x_1,1-x_2,1-x_3,x_4)F(1-x_1,x_2,x_3,1-x_4). \end{array}$$

Then F has a matchings circuit.

Proof. We will construct values $w(v_iv_j) \geq 0$ satisfying

$$\begin{aligned} F(\mathbf{e}_1) &= F(\overline{\mathbf{e}_2})w(v_3v_4) + F(\overline{\mathbf{e}_3})w(v_4v_2) + F(\overline{\mathbf{e}_4})w(v_2v_3) \\ F(\mathbf{e}_2) &= F(\overline{\mathbf{e}_3})w(v_4v_1) + F(\overline{\mathbf{e}_4})w(v_1v_3) + F(\overline{\mathbf{e}_1})w(v_3v_4) \\ F(\mathbf{e}_3) &= F(\overline{\mathbf{e}_4})w(v_1v_2) + F(\overline{\mathbf{e}_1})w(v_2v_4) + F(\overline{\mathbf{e}_2})w(v_4v_1) \\ F(\mathbf{e}_4) &= F(\overline{\mathbf{e}_1})w(v_2v_3) + F(\overline{\mathbf{e}_2})w(v_3v_1) + F(\overline{\mathbf{e}_3})w(v_1v_2) \end{aligned}$$

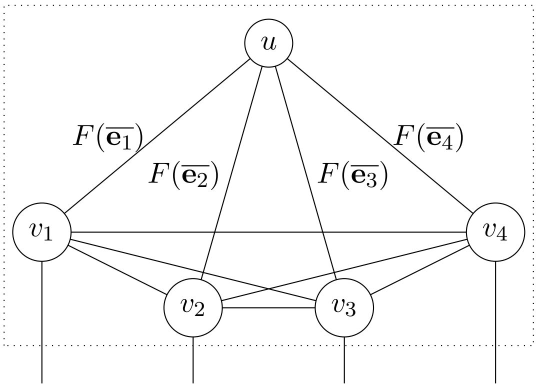

FIGURE 3.11: A weighted clique. All fugacities are zero, and $w(uv_i) = F(\overline{\mathbf{e}_i})$ for all i. The other edge-weights are to be determined.

(and also $w(v_iv_j) = w(v_jv_i)$.) This suffices because then $F = [\![G]\!]$ where G is the weighted clique illustrated in Figure 3.11. We need to show that the vector

$$\mathbf{y} = (F(\mathbf{e}_1), F(\mathbf{e}_2), F(\mathbf{e}_3), F(\mathbf{e}_4))$$

is a non-negative linear combination of the vectors $\mathbf{e}_i F(\overline{\mathbf{e}_j}) + \mathbf{e}_j F(\overline{\mathbf{e}_i})$ with $1 \le i < j \le 4$. By Farkas' lemma (Lemma 3.29), it suffices to show that $\mathbf{y} \cdot \boldsymbol{\varphi} \ge 0$ for all $\boldsymbol{\varphi} \in \mathbb{Q}^4$ satisfying

$$\varphi_i F(\overline{\mathbf{e}_j}) + \varphi_j F(\overline{\mathbf{e}_i}) \ge 0 \qquad \text{for all } 1 \le i < j \le 4. \tag{3.7}$$

If $\varphi_1, \varphi_2, \varphi_3, \varphi_4 \ge 0$ we are done. Otherwise $\varphi_i < 0$ for some i. By assumption $F(\overline{\mathbf{e}_j}) > 0$ for some j. If $j \ne i$, then (3.7) implies that $\varphi_j F(\overline{\mathbf{e}_i})$ is non-zero. In any case $F(\overline{\mathbf{e}_i}) > 0$. If $i = 1$ then

$$\begin{aligned}
F(\mathbf{e}_1)F(\overline{\mathbf{e}_1}) &\le F(\mathbf{e}_2)F(\overline{\mathbf{e}_2}) + F(\mathbf{e}_3)F(\overline{\mathbf{e}_3}) + F(\mathbf{e}_4)F(\overline{\mathbf{e}_4}) \\
-\varphi_1 F(\mathbf{e}_1)F(\overline{\mathbf{e}_1}) &\le -\varphi_1 F(\mathbf{e}_2)F(\overline{\mathbf{e}_2}) - \varphi_1 F(\mathbf{e}_3)F(\overline{\mathbf{e}_3}) - \varphi_1 F(\mathbf{e}_4)F(\overline{\mathbf{e}_4}) \\
-\varphi_1 F(\mathbf{e}_1)F(\overline{\mathbf{e}_1}) &\le \varphi_2 F(\mathbf{e}_2)F(\overline{\mathbf{e}_1}) + \varphi_3 F(\mathbf{e}_3)F(\overline{\mathbf{e}_1}) + \varphi_4 F(\mathbf{e}_4)F(\overline{\mathbf{e}_1}) \\
-\varphi_1 F(\mathbf{e}_1) &\le \varphi_2 F(\mathbf{e}_2) + \varphi_3 F(\mathbf{e}_3) + \varphi_4 F(\mathbf{e}_4)
\end{aligned}$$

Therefore $\mathbf{y} \cdot \boldsymbol{\varphi} \ge 0$. By symmetry the other cases, $i \ne 1$, are similar. $\square$

Theorem 3.6. Let $F\colon \{0,1\}^3 \to \mathbb{Q}_{\ge 0}$. The following are equivalent:

1. F is windable

2. For all $x_1, x_2, x_3 \in \{0,1\}$ we have

$$\begin{array}{cl} & F(x_1,x_2,x_3)F(1-x_1,1-x_2,1-x_3) \\ \leq & F(x_1,x_2,1-x_3)F(1-x_1,1-x_2,x_3) \\ + & F(x_1,1-x_2,x_3)F(1-x_1,x_2,1-x_3) \\ + & F(x_1,1-x_2,1-x_3)F(1-x_1,x_2,x_3) \end{array}$$

3. F has a matchings circuit

Proof. For notational convenience, in the following argument we will use a particular copy of $F_\oplus$. For all $x_1, x_2, x_3, x_4 \in \{0,1\}$ define

$$F'(x_1,x_2,x_3,x_4) = \begin{cases} F(x_1,x_2,x_3) & \text{if } x_1+x_2+x_3+x_4 \text{ is even} \\ 0 & \text{otherwise.} \end{cases}$$

($1 \implies 2$) Let B be a witness that F' is even-windable (using Lemma 3.15). Let $x_1, x_2, x_3 \in \{0,1\}$. Let $c \in \{0,1\}$ be the unique value such that $x_1+x_2+x_3+c$ is even. Then $(x_1,x_2,x_3,c) \oplus (1-x_1,1-x_2,1-x_3,1-c) = (1,1,1,1)$. Note that

$$\mathcal{M}_{(1,1,1,1)} = \{\{\{1,2\},\{3,4\}\},\{\{1,3\},\{2,4\}\},\{\{1,4\},\{2,3\}\}\}.$$

We have

$$\begin{aligned} & F(x_1,x_2,x_3)F(1-x_1,1-x_2,1-x_3) \\ & = F'(x_1,x_2,x_3,c)F'(1-x_1,1-x_2,1-x_3,1-c) \\ & = B((x_1,x_2,x_3,c),(1-x_1,1-x_2,1-x_3,1-c),\{\{1,2\},\{3,4\}\}) \\ & \qquad + B((x_1,x_2,x_3,c),(1-x_1,1-x_2,1-x_3,1-c),\{\{1,3\},\{2,4\}\}) \\ & \qquad + B((x_1,x_2,x_3,c),(1-x_1,1-x_2,1-x_3,1-c),\{\{1,4\},\{2,3\}\}) \\ & = B((x_1,x_2,1-x_3,1-c),(1-x_1,1-x_2,x_3,c),\{\{1,2\},\{3,4\}\}) \\ & \qquad + B((x_1,1-x_2,x_3,1-c),(1-x_1,x_2,1-x_3,c),\{\{1,3\},\{2,4\}\}) \\ & \qquad + B((x_1,1-x_2,1-x_3,c),(1-x_1,x_2,x_3,1-c),\{\{1,4\},\{2,3\}\}) \\ & \leq F'(x_1,x_2,1-x_3,1-c)F'(1-x_1,1-x_2,x_3,c) \\ & \qquad + F'(x_1,1-x_2,x_3,1-c)F'(1-x_1,x_2,1-x_3,c) \\ & \qquad + F'(x_1,1-x_2,1-x_3,c)F'(1-x_1,x_2,x_3,1-c) \\ & = F(x_1,x_2,1-x_3)F(1-x_1,1-x_2,x_3) \\ & \qquad + F(x_1,1-x_2,x_3)F(1-x_1,x_2,1-x_3) \\ & \qquad + F(x_1,1-x_2,1-x_3)F(1-x_1,x_2,x_3) \end{aligned}$$

($2 \implies 3$) We can assume that F is not identically zero (otherwise, take two vertices of fugacity 0, and attach four outgoing edges to one of them - the isolated vertex can never be matched). Pick $\mathbf{x} \in \{0,1\}^4$ with $F'(x_1, x_2, x_3, x_4) > 0$. Lemma 3.30 implies that the flip F'' of F' by $\mathbf{x} \oplus (1,1,1,0)$ has a matchings circuit. By subdividing the i'th outgoing edge for each i with $x_i = 1$, we get a matchings circuit for F'. By Lemma 3.26 we get a matchings circuit whose weight-function is F.

($3 \implies 1$) Lemma 3.28. □

In particular by Theorem 3.6 and Theorem 3.5, $\mathsf{Holant}(\{\mathbf{R}\}) \leq_{\mathrm{AP}} \#\mathsf{PM}$ where

$$R = \{(0,0,0), (1,0,0), (0,1,0), (1,0,1), (0,1,1)\}.$$

Chapter 4

Holant problems with arity three relations, and counting downsets

In this section we study some unweighted counting problems: first, degree-two #CSPs and Holant problems using arity 3 relations, and secondly, a restriction of #Downsets. We first discuss unweighted #CSPs without degree restrictions.

4.1 Introduction to bounded-degree unweighted #CSPs

There is a trichotomy for $\#\mathsf{CSP}(\Gamma)$ where Γ is a finite unweighted constraint language.

Proposition 4.1. *[48] Let Γ be a finite unweighted constraint language.*

- *If every relation in Γ is affine then $\#\mathsf{CSP}(\Gamma) \in \mathrm{FP}$.*
- *Otherwise, if $\Gamma \subseteq \mathsf{IMconj}$ then $\#\mathsf{CSP}(\Gamma) =_{\mathrm{AP}} \#\mathsf{BIS}$.*
- *Otherwise, $\#\mathsf{CSP}(\Gamma) =_{\mathrm{AP}} \#\mathsf{SAT}$.*

In particular $\#\mathsf{CSP}(\mathrm{NAND}) =_{\mathrm{AP}} \#\mathsf{SAT}$ and $\#\mathsf{CSP}(\mathrm{IMP}) =_{\mathrm{AP}} \#\mathsf{BIS}$. The following examples show that bounded-degree problems are unlikely to have a classification of the same form as Proposition 4.1 without some assumptions. Example 4.2 shows that $\#\mathsf{CSP}_{\leq 5}$ is different from $\#\mathsf{CSP}_{\leq 6}$.

Example 4.2. The problem $\#\mathsf{IS}_d$ of counting independent sets in graphs of maximum degree at most d has a (deterministic) FPRAS [100] for $d \leq 5$, while for $d \geq 6$, this problem has no FPRAS unless the complexity classes RP and NP are equal [92]. But $\#\mathsf{IS}_d$ is AP-equivalent to $\#\mathsf{CSP}_{\leq d}(\mathrm{NAND})$, where $\mathrm{NAND} = \{(0,0),(0,1),(1,0)\}$, by the correspondence mentioned in Section 1.3. ◇

Example 4.3 rules out any classification of constraint languages into a finite number of classes such that $\#\mathsf{CSP}_{\leq d}(\Gamma)$ can be classified up to AP-equivalence in a way that depends only on d and the class of Γ.

Example 4.3. A hypergraph is a set of vertices and a set of hyperedges, where a hyperedge is any set of vertices. The degree of a vertex v is the number of hyperedges including v, while the size of an hyperedge e is the number of vertices in e. An independent set of a hypergraph is a set of vertices I such that no hyperedge e is a subset of I. For all positive integers d, there is an FPRAS for the problem of counting independent sets in hypergraphs where every vertex has degree at most d and every hyperedge has size at least $2d + 1$ [9].

Unfortunately, the natural way to model hypergraph problems as a #CSP tends to give a subtly different problem - while the edges of a hypergraph are sets, the scope of a constraint may use the same variable more than once. However, we get the crude consequence that the problem $\#\mathsf{CSP}_{\leq d}(\mathrm{NAND}_{3d^2})$ has an FPRAS, where NAND_k denotes $\{0,1\}^k \backslash \{0\}^k$. Indeed, given an instance of $\#\mathsf{CSP}_{\leq d}(\mathrm{NAND}_{3d^2})$, consider the "primal constraint hypergraph" whose vertices are variables of the instance, and whose hyperedges are sets $\{v_1, \dots, v_{3d^2}\}$ for each constraint $\langle (v_1, \dots, v_{3d^2}), \mathrm{NAND}_{3d^2} \rangle$ (ignoring repeated hyperedges). Each hyperedge has size at least $3d \geq 2d + 1$, so we can count the number of independent sets in this hypergraph efficiently by [9].

Conversely $\#\mathsf{CSP}_{\leq d}(\mathrm{NAND}_{d/3})$ does not have an FPRAS, at least when d is a positive multiple of 6 and $\mathrm{RP} \neq \mathrm{NP}$. Recall from Example 4.2 that there is no FPRAS for $\#\mathsf{CSP}_{\leq 6}(\mathrm{NAND}_2)$ unless $\mathrm{RP} = \mathrm{NP}$. But given an instance of $\#\mathsf{CSP}_{\leq 6}(\mathrm{NAND}_2)$ we can replace each constraint $\langle (u, v), \mathrm{NAND}_2 \rangle$ by $\langle (u, \dots, u, v, \dots, v), \mathrm{NAND}_{d/3} \rangle$ – repeating each variable $d/6$ times – to get an instance of $\#\mathsf{CSP}_{\leq d}(\mathrm{NAND}_{d/3})$ with the same satisfying assignments. ◇

The problem $\#\mathsf{CSP}_{\leq d}(\Gamma)$ for $d \geq 3$ was studied in [51], giving a trichotomy for $d \geq 6$ for constraint languages that include constants:

Proposition 4.4. *[51, Theorem 24] Let Γ be a finite unweighted constraint language containing $\{(0)\}$ and $\{(1)\}$. Let $d \geq 6$.*

- *If every relation in Γ is affine then $\#\mathsf{CSP}_{\leq d}(\Gamma) \in \mathrm{FP}$.*
- *Otherwise, if $\Gamma \subseteq \mathsf{IMconj}$ then $\#\mathsf{CSP}_{\leq d}(\Gamma) =_{\mathrm{AP}} \#\mathsf{BIS}$.*
- *Otherwise, $\#\mathsf{CSP}_{\leq d}(\Gamma)$ has no FPRAS unless $\mathrm{RP} = \mathrm{NP}$.*

The aim of this chapter is to give an analogue of this result for $d = 2$. The distinction between degree-two #CSPs and Holant problems does not seem to be important — the only difference is that degree-two #CSPs allow variables to be used less than twice. But we state results in both settings in the hope that it is convenient for future research. Note that degree-two #CSPs can be viewed as read-twice #CSPs (where every variable must appear *exactly* twice):

Lemma 4.5. *[101, Proposition 9.2] $\#\mathsf{CSP}_{\leq 2}(R)$ is equivalent to $\mathsf{Holant}(R, \{0,1\}^1)$, under AP-reductions and under polynomial-time Turing reductions.*

Proof. (Note we have incorporated [101, Proposition 9.2] into our definitions to some extent, by taking Holant problems to be encoded like a constraint satisfaction problem rather than as a graph.)

Given an instance (V, C) of $\#\mathsf{CSP}_{\leq 2}(R)$ we can insert $2 - \deg_C(v)$ constraints $\langle (v), \{0,1\}^1 \rangle$ for each variable v, where $\deg_C(v)$ denotes the number of occurrences of v. This gives an instance of $\mathsf{Holant}(R, \{0,1\}^1)$ with the same partition function. Conversely, given an instance of $\mathsf{Holant}(R, \{0,1\}^1)$ we can delete the $\{0,1\}^1$ constraints to get an instance of $\#\mathsf{CSP}_{\leq 2}(R)$ with the same partition function. □

To emphasise the similarly between degree-two and read-twice problems, in this chapter we will write $\mathsf{Holant}(\mathcal{F})$ as $\#\mathsf{CSP}_{=2}(\mathcal{F})$.

4.2 Expressibility reductions

The reductions in this section are mostly a combination of previous results, and the following type of expressibility reduction. We will express a function F in the form

$$F(x_1, \dots, x_n) = \sum_{x_{n+1}, \dots, x_{n+m} \in \{0,1\}} \prod_{\langle (i_1, \dots, i_k), G \rangle \in C} G(x_{i_1}, \dots, x_{i_k}) \tag{4.1}$$

where on the right-hand-side each unsummed variable $x_1, \dots, x_n$ occurs exactly once and each summed variable $x_{n+1}, \dots, x_{n+m}$ occurs exactly twice.

If (4.1) holds, we can substitute each F constraint $\langle (v_1, \dots, v_n), F \rangle$ in a #CSP instance (V', C') by a copy of the constraints C, renaming the unsummed variables to $v_1, \dots, v_n$ and renaming the summed variables to new variables for each F constraint. This does not affect the partition function: if we let (V'', C'') denote the modified instance then

$$Z_{V',C'} = Z_{V'',C''}.$$

(In fact, each assignment $\sigma \colon V' \to \{0,1\}$ satisfies $\mathrm{wt}_{V',C'}(\sigma) = \sum \mathrm{wt}_{V'',C''}(\sigma')$ where the sum is over extensions of σ to $\sigma' \colon V'' \to \{0,1\}$. We formalise this argument further in Section 5.2.1.)

This gives both an AP-reduction and a polynomial-time Turing reduction, from $\#\mathsf{CSP}_{=2}(\mathcal{F} \cup \{F\})$ to $\#\mathsf{CSP}_{=2}(\mathcal{F})$, for any finite set $\mathcal{F}$ such that $G \in \mathcal{F}$ for all $\langle (i_1, \dots, i_k), G \rangle \in C$.

4.3 Exact evaluation

The problems $\#\mathsf{CSP}(\Gamma)$ in the last two cases in Proposition 4.1 are $\#\mathsf{P}$-complete; this was shown earlier by Creignou and Hermann [40]. But there turn out to be more tractable cases for degree two #CSPs and Holant problems.

In this section we will classify the complexity of exactly evaluating $\#\mathsf{CSP}_{=2}(\Gamma)$ where Γ is a set of relations of arity at most 3. We will do this by applying a dichotomy for symmetric Holant problems. Throughout this section, functions $\{0,1\}^k \to \mathbb{C}$ are called **signatures**, and $[f_0, \ldots, f_k]$ denotes the symmetric function $F\colon \{0,1\}^k \to \mathbb{C}$ defined by

$$F(x_1, \ldots, x_k) = f_{x_1+\cdots+x_k} \qquad (x_1, \ldots, x_k \in \{0,1\}).$$

We will use a dichotomy theorem for symmetric Holant problems stated in terms of certain classes of functions. A signature is **degenerate** if it is $[a^k, a^{k-1}b, a^{k-2}b^2, \ldots, b^k]$ for some $a, b \in \mathbb{C}$ and some integer $k \geq 0$. A symmetric signature is **of product type** if it is degenerate or $[a, 0, 0, \ldots, 0, b]$ or $[0, a, 0]$ for some $a, b \in \mathbb{C}$. Let $\mathcal{P}$ denote the class of signatures of product type. We will also need to refer to affine signatures. The symmetric affine signatures are listed below; see for example the discussion following Definition 2.7 in [31]. In each case λ is an arbitrary constant.

A1. $\lambda[1, 0, \ldots, 0, \pm 1]$
A2. $\lambda[1, 0, \ldots, 0, \pm\sqrt{-1}]$
A3. $\lambda[1, 0, 1, 0, \ldots, 0 \text{ or } 1]$
A4. $\lambda[1, -\sqrt{-1}, 1, -\sqrt{-1}, \ldots, -\sqrt{-1} \text{ or } 1]$
A5. $\lambda[0, 1, 0, 1, \ldots, 0 \text{ or } 1]$
A6. $\lambda[1, \sqrt{-1}, 1, \sqrt{-1}, \ldots, \sqrt{-1} \text{ or } 1]$
A7. $\lambda[1, 0, -1, 0, 1, 0, -1, 0, \ldots, 0 \text{ or } 1 \text{ or } -1]$
A8. $\lambda[1, 1, -1, -1, 1, 1, -1, -1, \ldots, 1 \text{ or } -1]$
A9. $\lambda[0, 1, 0, -1, 0, 1, 0, -1, \ldots, 0 \text{ or } 1 \text{ or } -1]$
A10. $\lambda[1, -1, -1, 1, 1, -1, -1, 1, \ldots, 1 \text{ or } -1]$

The dichotomy is stated in terms of signatures of the form $T^{\otimes k}F$. For any set of signatures $\mathcal{F}$ define $T\mathcal{F} = \{T^{\otimes k}F \mid k \geq 0 \text{ and } F\colon \{0,1\}^k \to \mathbb{C} \text{ is in } \mathcal{F}\}$.

Proposition 4.6. *[67] Let $\mathcal{F}$ be a set of symmetric signatures on Boolean variables with real values. Then $\#\mathsf{CSP}_{=2}(\mathcal{F})$ is $\#\mathsf{P}$-hard unless one of the following conditions hold.*

(i.) the arity of any non-degenerate signature in $\mathcal{F}$ is no more than 2

(ii.) $F \subseteq T\mathcal{A}$ for some orthogonal matrix $T \in \mathbb{C}^{2\times 2}$

(iii.) $F \subseteq T\mathcal{P}$ for some orthogonal matrix $T \in \mathbb{C}^{2\times 2}$

In these cases, $\#\mathsf{CSP}_{=2}(\mathcal{F})$ is computable in polynomial time.

(A matrix is **orthogonal** if it is the inverse of its transpose.) We will next show that these conditions simplify slightly if $\mathcal{F}$ consists of non-negative-valued signatures.

First, we will need to discuss a type of signature appearing in the third class in Proposition 4.6. A function $F = [f_0, \ldots, f_k]$ is a "generalised Fibonacci signature" if there exist $\alpha, \beta \in \mathbb{C}$ not both zero such that $\alpha f_{i-1} + \beta f_i - \alpha f_{i+1} = 0$ for all $1 \leq i \leq k-1$. These are mentioned in [67].

An important property is that a function is a generalised Fibonacci signature if and only if it is of the form $T^{\otimes k}[\lambda, 0, 0, \ldots, 0, \lambda']$ for some orthogonal matrix $T \in \mathbb{C}^{2\times 2}$. This property is used, for example, in the proof of [67, Corollary 4.6]; we will give a proof here for completeness.

For the forward direction, note that T is orthogonal so is either of the form $\left(\begin{smallmatrix} a & b \\ -b & a \end{smallmatrix}\right)$ or $\left(\begin{smallmatrix} -a & b \\ b & a \end{smallmatrix}\right)$. We have

$$\begin{aligned}\left(\begin{smallmatrix} a & b \\ -b & a \end{smallmatrix}\right)^{\otimes k}[\lambda, 0, 0, \ldots, 0, \lambda'] &= [\lambda a^k + \lambda' b^k, \lambda a^{k-1}(-b) + \lambda' a b^{k-1}, \ldots, \lambda(-b)^k + \lambda' a^k] \\ \left(\begin{smallmatrix} -a & b \\ b & a \end{smallmatrix}\right)^{\otimes k}[\lambda, 0, 0, \ldots, 0, \lambda'] &= [\lambda(-a)^k + \lambda' b^k, \lambda(-a)^{k-1} b + \lambda' a b^{k-1}, \ldots, \lambda b^k + \lambda' a^k]\end{aligned} \tag{4.2}$$

which both satisfy $abf_{i-1} + (a^2 - b^2)f_i - abf_{i+1} = 0$ for all $1 \le i \le k-1$.

Conversely given α, β not both zero, there are a, b not both zero satisfying $\alpha = ab$ and $\beta = a^2 - b^2$: if $\alpha = 0$ take $b = 0$ and pick a square root a of β and set $b = 0$, otherwise pick a zero X of $X^2 - (\beta/\alpha)X - 1$ (which ensures $\alpha X - \alpha/X = \beta$), pick a square root a of αX, and set $b = a/X$. Then the first equation of (4.2) gives the full two-dimensional space of solutions to the recurrence: $\alpha f_{i-1} + \beta f_i - \alpha f_{i+1} = 0$ for all $1 \le i \le k-1$.

Lemma 4.7. *Let $\mathcal{F}$ be a set of non-negative-valued symmetric signatures. If $\mathcal{F} \subseteq T\mathcal{A}$ or $\mathcal{F} \subseteq T\mathcal{P}$ for some orthogonal matrix $T \in \mathbb{C}^{2\times 2}$, then one of the following conditions holds.*

(i.) $\mathcal{F} \subseteq \mathcal{A}$

(ii.) $\mathcal{F} \subseteq \mathcal{P}$

(iii.) There exists $\mu \in \mathbb{R}$ such that every non-degenerate $[f_0, \ldots, f_k]$ in $\mathcal{F}$ satisfies $f_{i-1} + \mu f_i - f_{i+1} = 0$ for all $1 \le i \le k-1$

Proof. First consider the case that $\mathcal{F} \subseteq T\mathcal{P}$ for some orthogonal matrix T. Suppose that condition (ii.) fails, so T is not of the form $\left(\begin{smallmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{smallmatrix}\right)$ or $\left(\begin{smallmatrix} 0 & \pm 1 \\ \pm 1 & 0 \end{smallmatrix}\right)$. So T is of the form $\left(\begin{smallmatrix} a & b \\ -b & a \end{smallmatrix}\right)$ or $\left(\begin{smallmatrix} -a & b \\ b & a \end{smallmatrix}\right)$ with $a, b \neq 0$ and $a^2 + b^2 = 1$, and $\mathcal{F}$ consists of functions of the form $T^{\otimes 2}[0, \lambda, 0]$ or $T^{\otimes k}[\lambda, 0, 0, \ldots, 0, \lambda']$ (with $\lambda, \lambda' \in \mathbb{C}$). But $T^{\otimes 2}[0, \lambda, 0]$ cannot occur with $\lambda \neq 0$, because

$$\begin{aligned}\left(\begin{smallmatrix} a & b \\ -b & a \end{smallmatrix}\right)^{\otimes 2}[0, 1, 0] &= [2ab, a^2 - b^2, -2ab] \\ \left(\begin{smallmatrix} -a & b \\ b & a \end{smallmatrix}\right)^{\otimes 2}[0, 1, 0] &= [-2ab, b^2 - a^2, 2ab]\end{aligned}$$

and the values $2ab\lambda$ and $-2ab\lambda$ cannot both be non-negative. For $T^{\otimes k}[\lambda, 0, 0, \ldots, 0, \lambda']$, by (4.2) condition (iii.) is satisfied with $\mu = \frac{a^2-b^2}{ab}$; if this value is not real we can just take $\mu = 0$ because for any $[f_0, \ldots, f_k] \in \mathcal{F}$ the differences $f_{i-1} - f_{i+1} = \frac{a^2-b^2}{ab} f_i$ lie in $\frac{a^2-b^2}{ab}\mathbb{R} \cap \mathbb{R} = \{0\}$.

Now consider the case that $\mathcal{F} \subseteq T\mathcal{A}$ for some orthogonal matrix T. So $\mathcal{F}$ consists of functions $\lambda T^{\otimes k} F$ where $\lambda \in \mathbb{C}$ and F is in one the ten forms (A1)-(A10). If F is of the form (A2), (A4), or (A6), and F is not of the form (A1), (A3) or (A5), then $\sum_{x_1,\ldots,x_k=0}^{1} F(x_1,\ldots,x_k)F(x_1,\ldots,x_k)$ is zero. Since T is orthogonal,

$$\sum_{x_1,\ldots,x_k=0}^{1} F(x_1,\ldots,x_k)F(x_1,\ldots,x_k) = \sum_{x_1,\ldots,x_k=0}^{1} (T^{\otimes k}F)(x_1,\ldots,x_k)(T^{\otimes k}F)(x_1,\ldots,x_k)$$

but $T^{\otimes k}F$ takes non-negative values, so must be identically zero. Similarly if F is in one of the forms (A7)-(A10), and F is not of the form (A1), (A3) or (A5), then the arity $k-2$ function F' defined by $F'(x_1,\ldots,x_{k-2}) = \sum_{y=0}^{1} F(x_1,\ldots,x_{k-2},y,y)$ is identically zero. Thus $(T^{\otimes(k-2)}F')(x_1,\ldots,x_{k-2})$ is zero for all $x_1,\ldots,x_{k-2} \in \{0,1\}$. But by orthogonality of T,

$$(T^{\otimes(k-2)}F')(x_1,\ldots,x_{k-2}) = \sum_{y=0}^{1}(T^{\otimes k}F)(x_1,\ldots,x_{k-2},y,y)$$

for all $x_1,\ldots,x_{k-2} \in \{0,1\}$, so F is identically zero (or $F = [0,\lambda,0]$ for some λ, so F is of the form (A5)).

We have shown that for all $T^{\otimes k}F \in \mathcal{F}$, the function F is in one of the forms (A1), (A3) or (A5). Define $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$. Note that if T or TH is one of the eight matrices $\left(\begin{smallmatrix} \pm1 & 0 \\ 0 & \pm1 \end{smallmatrix}\right)$, $\left(\begin{smallmatrix} 0 & \pm1 \\ \pm1 & 0 \end{smallmatrix}\right)$ then $T^{\otimes k}F \in \mathcal{A}$ for any function F in the forms (A1), (A3), or (A5), which implies condition (i.). So we may assume T is not one of these eight matrices.

Both (A3) and (A5) are constant multiples of functions of the form $H^{\otimes k}[1,0,0,\ldots,0,\pm1]$, so $\mathcal{F}$ is contained in $T\mathcal{P} \cup TH\mathcal{P}$. We previously argued that if $\mathcal{F} \subseteq T\mathcal{P}$ then condition (ii.) or (iii.) holds. Similarly if $\mathcal{F} \subseteq TH\mathcal{P}$ then we are done. So we may assume that $\mathcal{F} \not\subseteq T\mathcal{P}$ and $\mathcal{F} \not\subseteq TH\mathcal{P}$. Note that all functions of the forms (A1), (A3) and (A5) of arity at most 2 are in $\mathcal{P}$. So $\mathcal{F}$ contains $S^{\otimes k}[\lambda,0,0,\ldots,0,\pm\lambda]$ where $S \in \{T, TH\}$ and $k \geq 3$. Since S is orthogonal, S is $\left(\begin{smallmatrix} a & b \\ -b & a \end{smallmatrix}\right)$ or $\left(\begin{smallmatrix} -a & b \\ b & a \end{smallmatrix}\right)$ where $a, b \neq 0$ and $a^2+b^2=1$. Let $G = S^{\otimes 4}[\lambda^2,0,0,0,\lambda^2]$. Let $F = S^{\otimes k}[\lambda,0,0,\ldots,0,\pm\lambda]$. By orthogonality of S, for all $y_1,\ldots,y_4 \in \{0,1\}$ we have

$$G(y_1,y_2,y_3,y_4) = \sum_{x_1,\ldots,x_{k-2}=0}^{1} F(x_1,\ldots,x_{k-2},y_1,y_2)F(x_1,\ldots,x_{k-2},y_3,y_4).$$

$F \in \mathcal{F}$ takes non-negative values, so $G(1,0,0,0), G(1,1,1,0) \geq 0$. But G is either

$$\left(\begin{smallmatrix} a & b \\ -b & a \end{smallmatrix}\right)^{\otimes 4}[\lambda^2,0,0,0,\lambda^2] = \lambda^2[a^4+b^4, -a^3b+ab^3, 2a^2b^2, -ab^3+a^3b, a^4+b^4] \text{ or}$$
$$\left(\begin{smallmatrix} -a & b \\ b & a \end{smallmatrix}\right)^{\otimes 4}[\lambda^2,0,0,0,\lambda^2] = \lambda^2[a^4+b^4, -a^3b+ab^3, 2a^2b^2, -ab^3+a^3b, a^4+b^4].$$

So $\lambda^2(-a^3b+ab^3)$ and $\lambda^2(-ab^3+a^3b)$ are non-negative, which implies that $a=0$ or $b=0$ or $a^2=b^2$. But this contradicts the assumption that T was not one of the eight matrices $\left(\begin{smallmatrix} \pm1 & 0 \\ 0 & \pm1 \end{smallmatrix}\right)$, $\left(\begin{smallmatrix} 0 & \pm1 \\ \pm1 & 0 \end{smallmatrix}\right)$. □

Lemma 4.7 simplifies the dichotomy slightly for non-negative-valued functions. Another way we will simplify the study of #CSPs is to restrict to studying constraint languages that do not contain Cartesian products. This is justified by the following argument.

Lemma 4.8. *Let $R \subseteq \{0,1\}^r$ and $S \subseteq \{0,1\}^s$ be non-empty relations, and let Γ be a set of relations. Then $\#\mathsf{CSP}_{=2}(\Gamma \cup \{R \times S\})$ is equivalent to $\#\mathsf{CSP}_{=2}(\Gamma \cup \{R, S\})$ under AP-reductions and under polynomial-time Turing reductions.*

Proof. We first give a reduction from $\#\mathsf{CSP}_{=2}(\Gamma \cup \{R\})$ to $\#\mathsf{CSP}_{=2}(\Gamma \cup \{R \times S\})$ where S is any non-empty relation. Consider an instance I of $\#\mathsf{CSP}_{=2}(\Gamma \cup \{R\})$ with k constraints using R. Let $2 \cdot I$ denote two disjoint copies of I. Let I' denote the instance of $\#\mathsf{CSP}_{=2}(\Gamma \cup \{R \times S\})$ obtained from $2 \cdot I$ by replacing each R constraint $c = \langle (v_{c,1}, \ldots, v_{c,r}), R\rangle$ and its corresponding copy $c' = \langle (v'_{c,1}, \ldots, v'_{c,r}), R\rangle$ with the constraints $\langle (v_{c,1}, \ldots, v_{c,r}, w_{c,1}, \ldots, w_{c,s}), R \times S\rangle$ and $\langle (v'_{c,1}, \ldots, v'_{c,r}, w_{c,1}, \ldots, w_{c,s}), R \times S\rangle$, where $w_{c,1}, \ldots, w_{c,s}$ are new variables.

$Z_{I'}$ gets k new factors of $\sum_{\mathbf{x} \in \{0,1\}^s} \mathbf{S}(\mathbf{x})\mathbf{S}(\mathbf{x}) = |S|$. So $Z_{I'} = |S|^k Z_{2 \cdot I} = |S|^k (Z_I)^2$. We can compute Z_I from $Z_{I'}$ by $Z_I = \sqrt{Z_{I'}/|S|^k}$. (For an AP-reduction, given error parameter ε, we can call the $Z_{I'}$ oracle with error parameter 2ε.) Repeating the same argument with the roles of R and S reversed, we get a reduction from $\#\mathsf{CSP}_{=2}(\Gamma \cup \{R, S\})$ to $\#\mathsf{CSP}_{=2}(\Gamma \cup \{R\} \cup \{R \times S\})$.

In the other direction we reduce $\#\mathsf{CSP}_{=2}(\mathcal{F} \cup \{R \times S\})$ to $\#\mathsf{CSP}_{=2}(\mathcal{F} \cup \{R, S\})$ by replacing each $\langle (v_1, \ldots, v_r, w_1, \ldots, w_s), R \times S\rangle$ constraint by a $\langle (v_1, \ldots, v_r), R\rangle$ constraint and a $\langle (w_1, \ldots, w_s), S\rangle$ constraint. $\square$

Proposition 4.9. *Let Γ be a set of relations of arity at most three, and assume that no relation in Γ is a Cartesian product of relations $R \subseteq \{0,1\}^I$ and $S \subseteq \{0,1\}^J$ with $|I|, |J| \geq 1$. Then $\#\mathsf{CSP}_{=2}(\Gamma)$ is $\#\mathsf{P}$-complete unless one of the following conditions hold.*

(i.) *every relation in Γ has arity at most two*

(ii.) *every relation in Γ is affine*

(iii.) $\Gamma \subseteq \{[1,0,1,1], [0,1,1], [1,0,1], [1,1], [0,1], [1,0]\}$

(iv.) $\Gamma \subseteq \{[1,1,0,1], [1,1,0], [1,0,1], [1,1], [0,1], [1,0]\}$

(See Figure 4.1.) In each of these cases, $\#\mathsf{CSP}_{=2}(\Gamma)$ can be computed in polynomial time.

Proof. $\#\mathsf{CSP}_{=2}(\Gamma)$ is clearly in $\#\mathsf{P}$. To show $\#\mathsf{P}$-hardness we use symmetrisation gadgets (this approach is used for similar purposes in [103]).

Let $\mathcal{F}$ be the set containing: the symmetric relations in Γ, and $[2,1,1]$ and $[1,1,2]$ if $\mathrm{IMP} = \{(0,0),(0,1),(1,1)\}$ is in Γ (or $\{(0,0),(1,0),(1,1)\}$ is in Γ), and for each arity 3

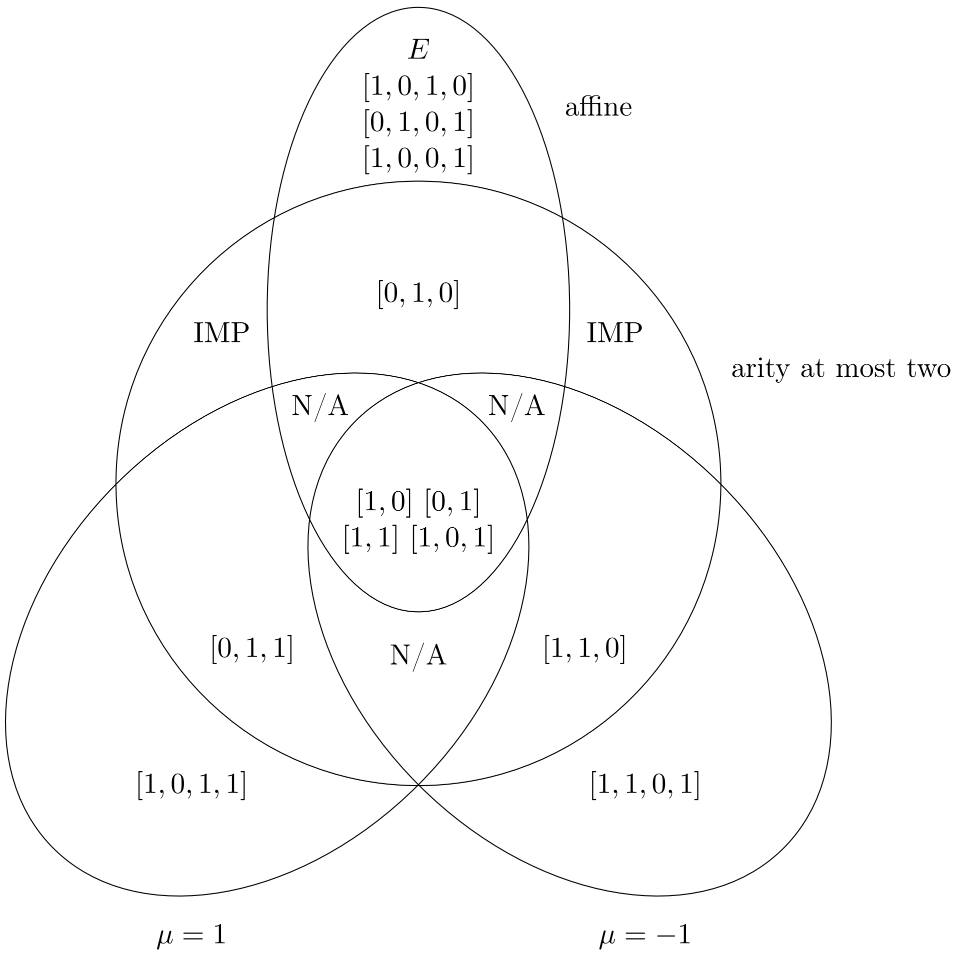

FIGURE 4.1: Tractable families of indecomposable relations of arity at most three. N/A means there are no indecomposable relations of arity at most three with the specified properties. IMP represents $\{(0,0),(0,1),(1,1)\}$ (and $\{(0,0),(1,0),(1,1)\}$), while E represents $\{(1,0,0),(0,1,1)\}$ (and $\{(0,1,0),(1,0,1)\}$ and $\{(0,0,1),(1,1,0)\}$).

relation R in Γ, the signatures $F_1, F_2, F_3 \colon \{0,1\}^3 \to \mathbb{C}$ and $F_4 \colon \{0,1\}^2 \to \mathbb{C}$ defined by

$$\begin{aligned}
F_1(x_1,x_2,x_3) &= \sum_{y_1,y_2,y_3 \in \{0,1\}} R(x_1,y_1,y_2)R(x_2,y_2,y_3)R(x_3,y_3,y_1) \\
F_2(x_1,x_2,x_3) &= \sum_{y_1,y_2,y_3 \in \{0,1\}} R(y_2,x_1,y_1)R(y_3,x_2,y_2)R(y_1,x_3,y_3) \\
F_3(x_1,x_2,x_3) &= \sum_{y_1,y_2,y_3 \in \{0,1\}} R(y_1,y_2,x_1)R(y_2,y_3,x_2)R(y_3,y_1,x_3) \\
F_4(x_1,x_2) &= \sum_{y_1,y_2 \in \{0,1\}} R(x_1,y_1,y_2)R(x_2,y_1,y_2)
\end{aligned}$$

F_4 and F_1 are represented in Figure 4.2.

These functions have cyclic symmetry, and any function on $\{0,1\}^2$ or $\{0,1\}^3$ with cyclic symmetry must be symmetric. We have

$$[2,1,1](x_1,x_2) = \sum_{y\in\{0,1\}} \mathrm{IMP}(x_1,y)\mathrm{IMP}(x_2,y) \text{ and}$$
$$[1,1,2](x_1,x_2) = \sum_{y\in\{0,1\}} \mathrm{IMP}(y,x_1)\mathrm{IMP}(y,x_2)$$

for all $x_1, x_2 \in \{0,1\}$, so there is an expressibility reduction from $\#\mathsf{CSP}_{=2}(\mathcal{F})$ to $\#\mathsf{CSP}_{=2}(\Gamma)$ (see Section 4.2). It remains to show that if Γ is not in any of the classes (i.) to (iv.) then $\#\mathsf{CSP}_{=2}(\mathcal{F})$ is not in any of the tractable classes given by Proposition 4.6. By Lemma 4.7 it suffices to show that $\mathcal{F} \not\subseteq \mathcal{A}$ and $\mathcal{F} \not\subseteq \mathcal{P}$ and there is no $\mu \in \mathbb{R}$ such that every non-degenerate symmetric signature $[f_0,\ldots,f_k]$ in $\mathcal{F}$ satisfies $f_{i-1} + \mu f_i - f_{i+1} = 0$ for all $1 \le i \le k-1$.

Here is how binary relations affect the classification. The binary relation $[1,1,0]$ is not affine or of product type, and forces $\mu = -1$, while the binary relation $[0,1,1]$ is not affine or of product type, and forces $\mu = 1$. On the other hand, $[0,1,0]$ is affine and of product type, but forces $\mu = 0$. If $\mathrm{IMP} \in \Gamma$ then $[2,1,1],[1,1,2] \in \mathcal{F}$, but these are not affine or of product type, and rule out any μ.

By considering each arity 3 relation, as shown in Figure 4.3, we can show that amongst arity 3 indecomposable relations R, the problem $\#\mathsf{CSP}_{=2}(R)$ is already $\#\mathsf{P}$-hard unless R is one of the following relations (possibly after permuting variables):

- $[1,0,0,1]$ and $\{(0,0,1),(1,1,0)\}$, which are affine and of product type; each rules out any μ.
- $[0,1,0,1]$ and $[1,0,1,0]$, which are affine but not of product type, and each forces $\mu = 0$.
- $[1,1,0,1]$, which is not affine or of product type, and forces $\mu = -1$.
- $[1,0,1,1]$, which is not affine or of product type, and forces $\mu = 1$.

Putting this all together verifies the $\#\mathsf{P}$-hard cases.

Indeed for the relations R marked F_i in Figure 4.3, $1 \le i \le 3$ the function $F_i = [f_0,f_1,f_2,f_3]$ defined above is already non-degenerate, not affine, and not of product type, and there is no μ such that $f_0 + \mu f_1 - f_2 = 0$ and $f_1 + \mu f_2 - f_3 = 0$. For the special cases marked * we find that F_4 is $[2,0,1]$ or $[1,0,2]$, which is not affine nor of product type and rules out any μ, while one of F_1, F_2, or F_3 is $[0,1,0,1]$ or $[1,0,1,0]$, which are non-degenerate signatures of arity greater than two. For example, in the case $R = \{(0,0,0),(0,1,1),(1,0,1)\}$ we get $F_4 = [2,0,1]$ and $F_3 = [1,0,1,0]$ in $\mathcal{F}$.

For tractability, for case (i.) we can use [27, Theorem 2.2], and for case (ii.) we can appeal to Proposition 4.1 (note that these results deal with asymmetric signatures). The third case consists of generalised Fibonacci signatures, $[f_0,\ldots,f_k]$ with $f_{i-1} + f_i - f_{i+1} = 0$. The last case consists of generalised Fibonacci signatures, $[f_0,\ldots,f_k]$ with

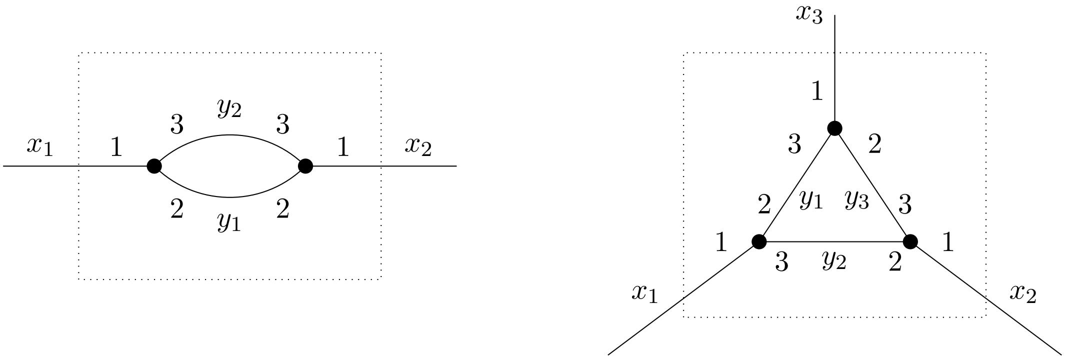

FIGURE 4.2: Symmetrisation gadgets F_4 and F_1. The numbers denote the position of a variable (drawn as an edge) in the constraint (drawn as a vertex).

$f_{i-1} - f_i - f_{i+1} = 0$. In cases (iii.) and (iv.) we can therefore appeal to Proposition 4.6. $\square$

		100	101	100 101	110	100 110	101 110	100 101 110	111	100 111	101 111	100 101 111	110 111	100 110 111	101 110 111	100 101 110 111
	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
000	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$*$	F_1	$\mathcal{P}$	F_1	F_3	F_1	F_2	F_1	F_2	F_1
001	$\times$	$\times$	$\times$	$\times$	$\mathcal{P}$	F_1	F_3	F_1	$\times$	$*$	$\times$	F_3	F_1	F_1	F_1	F_1
000,001	$\times$	$\times$	$\times$	$\times$	F_1	F_1	F_1	F_1	F_2	F_1	F_2	F_1	$\times$	F_1	F_1	$\times$
010	$\times$	$\times$	$\mathcal{P}$	F_1	$\times$	$\times$	F_2	F_1	$\times$	$*$	F_1	F_1	$\times$	F_2	F_1	F_1
000,010	$\times$	$\times$	F_1	F_1	$\times$	$\times$	F_1	F_1	F_3	F_1	$\times$	F_1	F_2	F_1	F_1	$\times$
001,010	$\times$	$*$	F_2	F_1	F_3	F_1	$\times$	F_1	$*$	$\mathcal{A}$	F_1	F_1	F_1	F_1	F_1	F_1
000,001,010	$\times$	F_1	F_1	F_1	F_1	F_1	F_1	$\times$	F_2	Fib	F_1	F_1	F_1	F_1	F_1	F_1
011	$\times$	$\mathcal{P}$	$\times$	F_3	$\times$	F_2	$*$	F_2	$\times$	F_1	$\times$	F_1	$\times$	F_1	F_1	F_1
000,011	$\times$	F_1	$*$	F_1	$*$	F_1	$\mathcal{A}$	F_2	F_1	$\times$	F_1	F_2	F_1	F_2	Fib	F_1
001,011	$\times$	F_2	$\times$	F_2	F_1	$\times$	F_1	F_1	$\times$	F_1	$\times$	F_1	F_1	F_1	F_1	$\times$
000,001,011	$\times$	F_1	F_2	F_1	F_1	F_1	F_1	F_1	F_1	F_2	F_1	$\times$	F_1	F_1	F_1	F_1
010,011	$\times$	F_3	F_1	$\times$	$\times$	F_2	F_1	F_1	$\times$	F_1	F_1	F_1	$\times$	F_1	F_1	$\times$
000,010,011	$\times$	F_1	F_1	F_1	F_3	F_1	F_1	F_1	F_1	F_2	F_1	F_1	F_1	$\times$	F_1	F_1
001,010,011	$\times$	F_2	F_1	F_1	F_1	F_1	F_1	F_1	F_1	F_2	F_1	F_1	F_1	F_1	$\times$	F_1
000,001,010,011	$\times$	F_1	F_1	$\times$	F_1	$\times$	F_1	F_1	F_1	F_1	$\times$	F_1	$\times$	F_1	F_1	$\times$

FIGURE 4.3: Case analysis of arity 3 Boolean relations. Each table entry represents the relation given by the tuples in the header of its row and its column. $\times$ means the relation is a Cartesian product of relations of arity at least 1. $\mathcal{P}$ means product type and affine, $\mathcal{A}$ means affine but not product type, and Fib means a generalised Fibonacci signature. F_1, F_2, F_3 refer to a symmetrisation gadget already proving #P-hardness, while the $*$ entries require the special symmetrisation argument.

4.4 Approximate evaluation

We now study approximate evaluation of degree-two #CSPs and Holant problems whose constraint language consists of a single arity 3 relation. The results are listed in Figure 4.4. The table only shows one member of each equivalence class under the symmetries of permuting variables and permuting the domain $\{0, 1\}$, because these operations do not affect the computational complexity. Also, the problems shown to be in FP by Proposition 4.9 are not shown. Each relation $R \subseteq \{0, 1\}^3$ is listed by writing ijk for each $(i, j, k) \in R$. For example the first row represents the relation $\{(0, 0, 1), (0, 1, 0), (1, 0, 0)\}$. In each row:

- "FPRAS" means that $\#\mathsf{CSP}_{\leq 2}(R)$ has an FPRAS
- "$\#\mathsf{X}$", where $\#\mathsf{X}$ is $\#\mathsf{PM}$, $\#\mathsf{BIS}$ or $\#\mathsf{SAT}$, means $\#\mathsf{CSP}_{\leq 2}(R) \leq_{\mathrm{AP}} \#\mathsf{X} \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(R)$
- "$\leq_{\mathrm{AP}} \#\mathsf{X}$" means $\#\mathsf{CSP}_{\leq 2}(R) \leq_{\mathrm{AP}} \#\mathsf{X}$

R								Complexity	
	100	010	001					$\#\mathsf{PM}$	Lemmas 4.14 & 4.15
000	100	010	001					FPRAS	Lemma 4.11
000			001	110					
	100		001	110				$\#\mathsf{SAT}$	Lemma 4.17
000	100		001	110					
	100	010	001	110				$\#\mathsf{SAT}$	Lemma 4.17
000	100	010	001	110				FPRAS	Lemma 4.11
000				110	101			$\#\mathsf{PM}$	Lemmas 4.14 & 4.16
000	100			110	101			FPRAS	Lemma 4.12
000		010		110	101				
	100		001	110	101			$\#\mathsf{SAT}$	Lemma 4.17
000	100	010		110	101				
000		010	001	110	101			$\leq_{\mathrm{AP}} \#\mathsf{PM}$	Lemma 4.14
	100	010	001	110	101			$\leq_{\mathrm{AP}} \#\mathsf{PM}$	Lemma 4.14
000	100			110	101	011			
000	100	010		110	101	011		FPRAS	Lemma 4.13
	100	010	001	110	101	011		FPRAS	Lemma 4.13
000	100	010	001	110	101	011		FPRAS	Lemma 4.13
000	100						111	$\#\mathsf{BIS}$	Lemma 4.17
000	100	010					111		
000	100				101		111	$\#\mathsf{BIS}$	Lemma 4.17
000	100	010		110			111	$\leq_{\mathrm{AP}} \#\mathsf{BIS}$	Proposition 4.1
000	100		001	110			111	$\leq_{\mathrm{AP}} \#\mathsf{PM}$	Lemma 4.14
000	100	010	001	110			111	FPRAS	Lemma 4.13
000		010	001	110	101		111	FPRAS	Lemma 4.13
000	100	010	001	110	101		111	FPRAS	Lemma 4.13

FIGURE 4.4: Summary of results for degree-two #CSPs and Holant problems.

Remark 4.10. The proof of Lemma 4.11 gives a deterministic FPRAS, sometimes called an FPTAS, for two rows.

4.4.1 FPRASes, and reductions from #CSPs to other problems

Lemma 4.11. *Let*

$$R = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$$
$$S = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1),(1,1,0)\}.$$

Then $\#\mathsf{CSP}_{\leq 2}(R,S)$ *has a FPRAS.*

Proof. Given an instance (V,C) of $\#\mathsf{CSP}_{\leq 2}(R,S)$, let G be the multigraph with vertex set V, three edges xy, xz, yz for each $\langle (x,y,z), R\rangle \in C$, and two edges xz, yz for each $\langle (x,y,z), S\rangle \in C$. Then G has maximum degree four, and there is a bijection $\sigma \mapsto \sigma^{-1}(\{1\})$ from satisfying assignments of (V,C) to independent sets of G. We can then use an FPRAS for the problem of counting independent sets in multigraphs of maximum degree at most five [100]. □

Lemma 4.12. *Let* $R = \{(0,0,0),(1,0,0),(1,1,0),(1,0,1)\}$. *Then* $\#\mathsf{CSP}_{\leq 2}(R)$ *has an FPRAS.*

Proof. We will reduce to computing the number of (not necessarily perfect) matchings in a multigraph, which has an FPRAS [70, Corollary 4.5]. For all integers $k \geq 1$ define

$$M_k(x_1,\dots,x_k) = \begin{cases} 1 & \text{if } x_1 + \cdots + x_k \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

Note that $M_1(0) = M_1(1) = 1$. There is an expressibility reduction from $\#\mathsf{CSP}_{\leq 2}(R)$ to $\#\mathsf{CSP}_{\leq 2}(\mathrm{NEQ}, M_3)$ where $\mathrm{NEQ} = \{(0,1),(1,0)\}$:

$$R(x,y,z) = \sum_{x' \in \{0,1\}} \mathrm{NEQ}(x,x') M_3(x',y,z).$$

By Lemma 4.5, $\#\mathsf{CSP}_{\leq 2}(\mathrm{NEQ}, M_3) \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(\mathrm{NEQ}, M_1, M_3)$. It remains to establish an AP-reduction from $\#\mathsf{CSP}_{=2}(\mathrm{NEQ}, M_1, M_3)$ to the problem of counting matchings in a multigraph. The important observation is that two M constraints joined by a NEQ constraint act like a single M constraint:

$$\sum_{x,y} M_{k+1}(x,x_1,\dots,x_k)\mathrm{NEQ}(x,y)M_{\ell+1}(y,y_1,\dots,y_\ell) = M_{k+\ell}(x_1,\dots,x_k,y_1,\dots,y_\ell)$$

for all $x_1,\dots,x_k,y_1,\dots,y_\ell \in \{0,1\}$. We will use this identity to eliminate constraints until we are only left with constraints over $\{M_k \mid k \geq 1\}$.

We are given an instance (V,C) of $\#\mathsf{CSP}_{=2}(\mathrm{NEQ}, M_1, M_3)$ and an error parameter ε. Set $(m_1, V_1, C_1) = (1, V, C)$. Given (m_i, V_i, C_i) such that:

- $Z_{V,C} = m_i Z_{V_i,C_i}$, and
- $\deg_{C_i}(v) = 2$ for all $v \in V_i$, and
- C_i consists of constraints over $\{\mathrm{NEQ}, \mathrm{PIN}_0, \mathrm{PIN}_1\} \cup \{M_k \mid k \geq 1\}$, where $\mathrm{PIN}_0 = \{(0)\}$ and $\mathrm{PIN}_1 = \{(1)\}$,

we will show that if C_i uses a NEQ, PIN_0, or PIN_1 constraint, then we can efficiently construct $(m_{i+1}, V_{i+1}, C_{i+1})$ with these same three properties, such that V_{i+1} is a strict subset of V_i. Note that $|C_i|$ is necessarily bounded by $2|V_i|$, so to get a polynomial time algorithm we only need to ensure that each step runs in polynomial time in $|V_i|$.

Consider the case where, for two variables $x, y \in V_i$, the constraint $\langle (x, y), \mathrm{NEQ} \rangle$ appears twice in C_i. Then we can set $V_{i+1} = V_i \setminus \{x, y\}$, set C_{i+1} to be C_i with the two $\langle (x, y), \mathrm{NEQ} \rangle$ constraints removed, and set $m_{i+1} = 2m_i$. This is correct because $Z_{V_i,C_i} = 2 \cdot Z_{V_{i+1},C_{i+1}}$. We denote this transformation by:

$$\textstyle\sum_{x,y} \mathrm{NEQ}(x, y)\mathrm{NEQ}(x, y) \to 2.$$

This notation means that the *summed* variables and constraints represented by the left-hand-side are deleted, and we multiply m_i by 2. In this case we do not introduce any new constraints. Since NEQ is a symmetric relation (as are PIN_0, PIN_1, and M_k), the order of variables does not matter: we can apply the same transformation if $\langle (x, y), \mathrm{NEQ} \rangle$ and $\langle (y, x), \mathrm{NEQ} \rangle$ are in C_i.

A more complicated case is when there is a constraint $\langle (x, y), \mathrm{NEQ} \rangle$, and x appears in a constraint using M_k for some $k \geq 1$, and y appears in a different constraint, using M_ℓ for some $\ell \geq 1$ (possibly with $k = \ell$). Set $V_{i+1} = V_i \setminus \{x, y\}$, let C_{i+1} be the result of deleting these three constraints using x or y then inserting a new constraint $\langle (x_1, \ldots, x_k, y_1, \ldots, y_\ell), M_{k+\ell} \rangle$, and set $m_{i+1} = m_i$. This is correct because $Z_{V_i,C_i} = Z_{V_{i+1},C_{i+1}}$. We denote this transformation by

$$\textstyle\sum_{x,y} M_{k+1}(x, x_1, \ldots, x_k)\mathrm{NEQ}(x, y)M_{\ell+1}(y, y_1, \ldots, y_\ell) \to M_{k+\ell}(x_1, \ldots, x_k, y_1, \ldots, y_\ell).$$

Again, we delete the summed variables and constraints represented by the left-hand-side, and the order of variables in a constraint does not matter. In this case we do not need to multiply m_i, but we do get a new constraint, represented by the expression on the right-hand-side.

If C_i has a $\langle (x), \mathrm{PIN}_0\rangle$ or $\langle (x), \mathrm{PIN}_1\rangle$ constraint for some $x \in V_i$, we can apply one of the following transformations, depending on which other constraint x appears in:

$$\begin{aligned}
\textstyle\sum_x \mathrm{PIN}_0(x)\mathrm{PIN}_0(x) &\to 1 \\
\textstyle\sum_x \mathrm{PIN}_1(x)\mathrm{PIN}_1(x) &\to 1 \\
\textstyle\sum_x \mathrm{PIN}_0(x)\mathrm{PIN}_1(x) &\to 0 \\
\textstyle\sum_x M_{k+1}(x, x_1, \dots, x_k)\mathrm{PIN}_0(x) &\to M_k(x_1, \dots, x_k) \\
\textstyle\sum_x M_{k+1}(x, x_1, \dots, x_k)\mathrm{PIN}_1(x) &\to \mathrm{PIN}_0(x_1)\dots\mathrm{PIN}_0(x_k) \\
\textstyle\sum_x \mathrm{PIN}_0(x)\mathrm{NEQ}(x, y) &\to \mathrm{PIN}_1(y) \\
\textstyle\sum_x \mathrm{PIN}_1(x)\mathrm{NEQ}(x, y) &\to \mathrm{PIN}_0(y)
\end{aligned}$$

If C_i has a $\langle (x, y), \mathrm{NEQ}\rangle$ constraint for some $x, y \in V_i$, we can apply one of the following transformations, depending on whether $x = y$, or otherwise depending on which other constraints x and y appear in:

$$\begin{aligned}
\textstyle\sum_x \mathrm{NEQ}(x, x) &\to 0 \\
\textstyle\sum_{x,y} \mathrm{NEQ}(x, y)\mathrm{NEQ}(x, y) &\to 2 \\
\textstyle\sum_{x,y} \mathrm{NEQ}(z, x)\mathrm{NEQ}(x, y)\mathrm{NEQ}(y, w) &\to \mathrm{NEQ}(z, w) \\
\textstyle\sum_{x,y} M_{k+1}(x, x_1, \dots, x_k)\mathrm{NEQ}(x, y)M_{\ell+1}(y, y_1, \dots, y_\ell) &\to M_{k+\ell}(x_1, \dots, x_k, y_1, \dots, y_\ell) \\
\textstyle\sum_{x,y} M_{k+1}(x, x_1, \dots, x_k)\mathrm{NEQ}(x, y)\mathrm{NEQ}(y, z) &\to M_{k+1}(z, x_1, \dots, x_k) \\
\textstyle\sum_{x,y} M_{k+2}(x, y, x_1, \dots, x_k)\mathrm{NEQ}(x, y) &\to 2\mathrm{PIN}_0(x_1)\dots\mathrm{PIN}_0(x_k)
\end{aligned}$$

(It is possible to show that some of these rules are unnecessary, but that argument seems to be much more complicated than just covering extra cases.)

The remaining case is that C_i does not contain a NEQ, PIN_0 or PIN_1 constraint. Let G be the multigraph whose vertex set is C_i and, for each variable $v \in C_i$, there is an edge between the two constraints in which v is used in C_i. The number of matchings in G is Z_{V_i,C_i}, and we wish to approximate $Z_{V,C} = m_i Z_{V_i,C_i}$. By [70, Corollary 4.4] we can find an approximation q of Z_{V_i,C_i} satisfying the FPRAS condition $\Pr(e^{-\varepsilon} Z_{V_i,C_i} \le q \le e^{\varepsilon} Z_{V_i,C_i}) \ge \frac{3}{4}$. We can then output $m_i q$. $\square$

We will use the results of Chapter 3 concerning windable strictly terraced functions. By Theorem 3.5, the condition for an arity three relation to be windable simplifies to:

There are at least four triples $(x, y, z) \in R$ with $(1-x, 1-y, 1-z) \in R$. (Wind)

Also note that an arity three relation R is strictly terraced whenever it is coindependent, that is, $(1-x, y, z), (x, 1-y, z), (x, y, 1-z) \notin R$ for all $(x, y, z) \in R$. Thus:

Lemma 4.13. *Let R be a coindependent relation satisfying (Wind). Then $\#\mathsf{CSP}_{\le 2}(R)$ has an FPRAS.*

Proof. The function U defined by $U(0) = U(1) = 1$ is windable and strictly terraced. $\#\mathsf{CSP}_{\leq 2}(R) =_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(R, U)$ by Lemma 4.5, and $\#\mathsf{CSP}_{=2}(R, U)$ has an FPRAS by Theorem 3.4. □

Lemma 4.14. *If $R \subseteq \{0, 1\}^3$ satisfies (Wind) then $\#\mathsf{CSP}_{\leq 2}(R) \leq_{\mathrm{AP}} \#\mathsf{PM}$.*

Proof. By Theorem 3.6, R has a matchings circuit. The function U defined by $U(0) = U(1) = 1$ has a matchings circuit (take a vertex of fugacity 1 with one external edge). The reduction is given by Theorem 3.5 and Lemma 4.5. □

4.4.2 Reductions from other problems to Holant problems

The following reduction is essentially due to Fisher [53].

Lemma 4.15. $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(\mathrm{PM}_3)$

Proof. The reduction is given a graph G and error parameter ε. For any degree-1 vertex v we can delete v and its neighbour; repeating this process gives a graph with the same number of perfect matchings and no vertices of degree 1. If G has a vertex of degree zero or if $|G|$ is odd, then G has no perfect matchings and we can output zero. So we may assume that G has minimum degree at least two and that $|G|$ is even. Replace any vertex of degree $k > 3$ by vertices of degree 2 and 3 as shown in Figure 4.5.

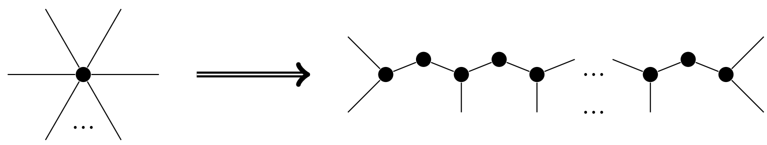

FIGURE 4.5: A vertex of degree $k > 3$ is transformed to vertices of degree two and three, preserving the total number of perfect matchings. In words, the vertex is replaced by a path $v_1 u_1 v_2 u_2 \ldots u_{k-3} v_{k-2}$, and the k edges that were incident to v are now incident to $v_1, v_1, v_2, v_3, \ldots, v_{k-3}, v_{k-2}, v_{k-2}$ respectively.

This produces a multigraph G' with the same number of perfect matchings as G, and where every vertex has degree 2 or 3, and with an even number of vertices. Any graph has an even number of odd-degree vertices, so G' has an even number of even-degree vertices. Enumerate the degree 2 vertices as $v_1, \ldots, v_{2k}$. For each $1 \leq i \leq k$ add new vertices x_i, y_i and edges $v_i x_i, v_{i+1} x_i, x_i y_i, y_i y_i$ to produce a new multigraph G''. Observe that for all i the edge $x_i y_i$ is in every perfect matching in G'', so G'' has the same number of perfect matchings as G.

Let C consist of one constraint $\langle (e_1, e_2, e_3), \mathrm{PM}_3 \rangle$ for each vertex in G'' with incident edges e_1, e_2, e_3 (possibly with repeats, corresponding to loops in G''). The reduction calls the oracle on $(E(G''), C)$ with error parameter ε. This is correct because number of perfect matchings in G'' is $Z_{E(G''),C}$. □

Lemma 4.16. *Let $R = \{(0, 0, 0), (0, 1, 1), (1, 0, 1)\}$. Then $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(R)$.*

Proof. Define $F\colon \{0,1\}^2 \to \mathbb{R}_{\geq 0}$ by $F(0,0)=1$ and $F(1,1)=2$ and $F(0,1)=F(1,0)=0$. It suffices to establish the following reductions:

$$\begin{aligned}\#\mathsf{PM} &\leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(\mathrm{PM}_3)\\ &\leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(R,\mathrm{PIN}_1)\\ &\leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(R,\{(1,1)\})\\ &\leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(R,F)\\ &\leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(R).\end{aligned}$$

The first reduction $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(\mathrm{PM}_3)$ is Lemma 4.15. The second is an expressibility reduction $\#\mathsf{CSP}_{=2}(\mathrm{PM}_3) \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(R,\mathrm{PIN}_1)$:

$$\mathrm{PM}_3(x,y,z) = \sum_{i,j\in\{0,1\}} R(x,i,j)R(i,y,z)\mathrm{PIN}_1(j) \qquad (x,y,z\in\{0,1\}).$$

The third reduction $\#\mathsf{CSP}_{=2}(R,\mathrm{PIN}_1) \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(R,\{(1,1)\})$ follows from Lemma 4.8.

For the fourth reduction $\#\mathsf{CSP}_{=2}(R,\{(1,1)\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(R,F)$ we are given an instance (V,C) of $\#\mathsf{CSP}_{=2}(R,\{(1,1)\})$ and error parameter ε, which we can assume is less than $1/2$. Let k be $|V|+1+\lceil\log_2\varepsilon^{-1}\rceil$: the least integer satisfying $2^{|V|-k}\leq\varepsilon/2$. Denote the constraints using $\{(1,1)\}$ by $\langle(x_1,y_1),\{(1,1)\}\rangle, \dots, \langle(x_m,y_m),\{(1,1)\}\rangle$. Let (V',C') denote the result of replacing, for each $1\leq i\leq m$, the constraint $\langle(x_i,y_i),\{(1,1)\}\rangle$ by

$$\langle(x_i,v_{i,1}),F\rangle, \langle(v_{i,1},v_{i,2}),F\rangle, \dots \langle(v_{i,k-1},y_i),F\rangle \tag{4.3}$$

where $v_{i,1},\dots,v_{i,k-1}$ are new variables. The reduction calls the oracle with error parameter $\varepsilon/2$ to obtain a value q, outputs zero if $q/2^{mk}<1/2$, and otherwise outputs $q/2^{mk}$.

For any $\sigma\colon V\to\{0,1\}$ let $\sigma'\colon V'\to\{0,1\}$ denote the extension of σ with $\sigma'(v_{i,1})=\cdots=\sigma'(v_{i,k-1})=\sigma(x_i)$ for each $1\leq i\leq m$. If σ is a satisfying assignment of (V,C), then the weight of σ is 1 and the weight of σ' is 2^{mk}. Summing over σ we get $Z_{V,C}\leq Z_{V',C'}/2^{mk}$. Conversely any assignment of (V',C') of positive weight is of the form σ' for some σ, and if σ is not a satisfying assignment of (V,C) then the weight of σ' is at most $2^{(m-1)k}$. Summing over σ we get

$$Z_{V,C}\leq Z_{V',C'}/2^{mk}\leq Z_{V,C}+2^{|V|-k}.$$

With probability at least 3/4, the value q satisfies

$$e^{-\varepsilon/2}Z_{V',C'}\leq q\leq e^{\varepsilon/2}Z_{V',C'}$$

If $Z_{V,C} = 0$ then $q/2^{mk} \leq e^{\varepsilon/2}2^{|V|-k} < 1/2$ so the reduction correct outputs zero. If $Z_{V,C} \geq 1$ then

$$\begin{aligned} e^{-\varepsilon/2}Z_{V,C} \leq q/2^{mk} &\leq e^{\varepsilon/2}(Z_{V,C} + 2^{|V|-k}) \\ &\leq e^{\varepsilon/2}(Z_{V,C} + \varepsilon/2) \\ &\leq e^{\varepsilon/2}Z_{V,C}(1+\varepsilon/2) \\ &\leq e^{\varepsilon}Z_{V,C} \end{aligned}$$

as required.

The fifth reduction $\#\mathsf{CSP}_{=2}(R,F) \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(R)$ is an expressibility reduction:

$$F(x,y) = \sum_{i,j\in\{0,1\}} R(i,j,x)R(i,j,y) \qquad (x,y \in \{0,1\})$$ □

Lemma 4.17. *Let* $R \subseteq \{0,1\}^3$*, and assume:*

- *for all* $(x,y,z) \in R$ *we have* $x \leq y$
- *either* $(0,0,0),(1,1,1) \in R$ *and* $(0,0,1),(1,1,0) \notin R$*, or* $(0,0,0),(1,1,1) \notin R$ *and* $(0,0,1),(1,1,0) \in R$

Then $\#\mathsf{CSP}_{=2}(R) =_{\mathrm{AP}} \#\mathsf{CSP}(R)$*, so the classification of Proposition 4.1 applies.*

Proof. The reduction $\#\mathsf{CSP}_{=2}(R) \leq_{\mathrm{AP}} \#\mathsf{CSP}(R)$ is obvious. For the other direction we will use the technique of 2-simulating equality from [51]. The reduction is given an instance (V,C) of $\#\mathsf{CSP}(R)$. For each $v \in V$ let $\deg_C(v)$ denote the total number of occurrences of v in C. Let V' consist of distinct variables $w_{v,i}, w'_{v,i}$ for each $v \in V$ and each $1 \leq i \leq \deg_C(v)$. Let C' consist of constraints:

- $\langle (w'_{v,i}, w'_{v,i+1}, w_{v,i}), R\rangle$ for each $v \in V$ and each $1 \leq i \leq \deg_C(v)$ where $w'_{v,\deg_C(v)+1}$ means $w'_{v,1}$, and
- $\langle (w_{v_1,i_1}, w_{v_2,i_2}, w_{v_3,i_3}), R\rangle$ for each constraint $\langle (v_1,v_2,v_3), R\rangle \in C$ such that this use of v_1 is the i_1'th occurrence of v_1, this use of v_2 is the i_2'th occurrence of v_2, and this use of v_3 is the i_3'th occurrence of v_3.

For each $v \in V$, the $\langle (w'_{v,i}, w'_{v,i+1}, w_{v,i}), R\rangle$ constraints force the variables $w'_{v,i}$ to take the same value for $1 \leq i \leq \deg_C(v)$. Given a satisfying assignment σ of (V,C) we get a satisfying assignment σ' of (V',C') by, for each $v \in V$ and each $1 \leq i \leq \deg_C(v)$, setting $\sigma'(w_{v,i}) = \sigma(v)$, and $\sigma'(w'_{v,i}) = \sigma(v)$ if $(0,0,0) \in R$, and $\sigma'(w'_{v,i}) = 1 - \sigma(v)$ if $(0,0,0) \notin R$. Conversely any satisfying assignment of (V',C') arises as σ' for some satisfying assignment σ of (V,C). Therefore (V',C') is an instance of $\#\mathsf{CSP}_{=2}(R)$ such that $Z_{V',C'} = Z_{V,C}$. □

4.5 Downsets in directed acyclic graphs of maximum degree three

As mentioned in the introduction (Example 1.1), one of the problems originally shown to be AP-equivalent to #BIS is #Downsets. The input of #Downsets is usually defined to be a partially ordered set. However, in this section we will think of the input as being a directed graph. A **downset** in a directed graph is a set of vertices D such that for all $u \in D$ and all arcs $u \to v$, we have $v \in D$. We can then consider the following problem, which is easily seen to be AP-equivalent to the usual definition as given in the introduction.

Name. #Downsets
Instance. A directed graph G.
Output. The number of downsets in G.

Downsets correspond to satisfying assignments of the constraints $\langle (u,v), \mathrm{IMP}\rangle$ for each $u \to v$, where $\mathrm{IMP} = \{(0,0),(0,1),(1,1)\}$. The restriction of #Downsets to directed graphs of maximum degree at most d is therefore AP-equivalent to $\#\mathsf{CSP}_{\leq d}(\mathrm{IMP})$. ("Degree" will always mean total degree: the sum of the in-degree and out-degree.) In particular the restriction to directed graphs of maximum degree at most 2 is in FP by Proposition 4.9, and the restriction to directed graphs of maximum degree at most 3 is AP-equivalent to #BIS by [51, Theorem 23].

However, the reductions used in [51] rely in a crucial way on cycles. Contracting a cycle to a single vertex does not affect the number of downsets. So #Downsets is AP-equivalent to its restriction to directed acyclic graphs, which we abbreviate as **dags**. In this section we will show that #Downsets remains AP-equivalent to #BIS even when restricted to dags of maximum degree three.

Remark 4.18. Consider the restriction of #Downsets to dags of depth two, that is, directed bipartite graphs with a bipartition $U \cup V$ such that all the arcs go from U to V. There is a bijection $I \mapsto I \triangle V$ from the set of independent sets of the underlying undirected graph to the set of downsets of the original graph. So the restriction of #Downsets to directed graphs of depth two is still AP-equivalent to #BIS. But there is an FPRAS for the restriction of #Downsets to dags of depth two and maximum degree five - we can use the equivalence just described for dags of depth two, then apply the Weitz's FPRAS for counting independent sets in graphs of maximum degree five [100].

Lemma 4.19. *Given an integer $n \geq 4$ specified in binary, we can build a dag G that has exactly n downsets, in polynomial time in $\log n$, such that G has a unique source and a unique sink, which are distinct and each have degree 1, and such that every vertex of G has degree at most three.*

Proof. We will first define certain graphs H_p, $p \geq 2$. Define H_2 to be the directed path of length one. For each integer $p \geq 2$, define H_{p+1} by adding three new vertices s, u, t to

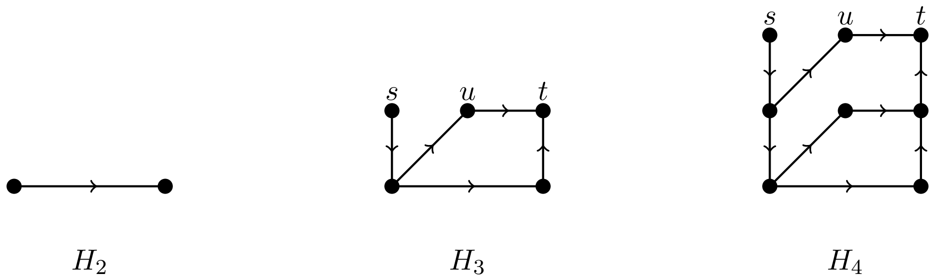

FIGURE 4.6: Three H_p graphs used in Lemma 4.19.

H_p and arcs $s \to s' \to u \to t$ and $t' \to t$ where s' denotes the source of H_p and t' denotes the sink of H_p; see Figure 4.6. The downsets of H_{p+1} are $\emptyset$, $V(H_p)$, $V(H_p) \cup \{u, t\}$, and $D \cup \{t\}$ and $D \cup \{u, t\}$ for each downset D of H_p that does not include the source s', that is, $D \neq V(H_p)$. Letting $h(p)$ denote the number of downsets in H_p we have $h(2) = 3$ and $h(p+1) = 2(h(p) - 1) + 3$. So $h(p) = 2^p - 1$ for all $p \geq 2$.

Let n' denote the greatest multiple of four with $n' \leq n$. Denote the binary expansion of n' by $2^{p_1} + \cdots + 2^{p_k}$, where $2 \leq p_1 < p_2 < \cdots < p_k$. Let $\ell = n - n' + 4k - 2$. Let G consist of a copy H'_i of H_{p_i} for each $1 \leq i \leq k$, as well as $\ell - k$ copies $H'_{k+1}, \ldots, H'_\ell$ of the directed edge H_2, except that for each $2 \leq i \leq \ell$ we identify the sink of H'_{i-1} with the source of H'_i. The downsets of G are $\emptyset$, $V(G)$, and $D \cup V(H'_{i+1}) \cup \cdots \cup V(H'_\ell)$ for each $1 \leq i \leq \ell$ and each downset D of H'_i that includes the sink of H'_i but not the source. Note we have not counted any downset twice. So the number of downsets in G is $2 + (h(p_1) - 2) + \cdots + (h(p_k) - 2) + (\ell - k)(h(2) - 2) = 2 + n' - 3k + \ell - k = n$. Note that the graphs H_m have $O(m)$ vertices, so G has $O((\log n)^2)$ vertices and can be constructed in polynomial time in $\log n$ as required. $\square$

Theorem 4.20. #Downsets *is AP-equivalent to its restriction to dags of maximum degree at most three.*

Proof. We will describe an AP-reduction from #BIS. Consider an instance G of #BIS, specified by vertex sets U and V and a set of edges $E \subseteq U \times V$, and an error parameter $0 < \varepsilon < 1$. We may assume that there are no isolated vertices. Let $w = \lceil (2/\varepsilon) 4^{|U|+|V|+|E|} \rceil$. For each $v \in U \cup V$ choose an enumeration $\{e_{v,1}, \cdots, e_{v,\deg(v)}\}$ of the edges incident to v, where $\deg(v)$ denotes the degree of v. Let G_v be a copy of the dag with exactly $w^{\deg(v)} + 2$ downsets given by Lemma 4.19. For each $e \in E$ let G_e be a copy of the dag with exactly $w + 2$ downsets given by Lemma 4.19. For each $x \in U \cup V \cup E$ let $s(x)$ denote the source of G_x and let $t(x)$ denote the sink of G_x. Construct G' by taking the disjoint union of the dags G_x with $x \in U \cup V \cup E$, together with arcs $t(u) \to s(e_{u,1}) \to \cdots \to s(e_{u,\deg(u)})$ for each $u \in U$ and $t(e_{v,1}) \to \cdots \to t(e_{v,\deg(v)}) \to s(v)$ for each $v \in V$. The reduction calls the oracle on G' with error parameter $\varepsilon/2$, divides by $w^{|E|}$, and returns the result.

Let Z denote the number of independent sets in G, and let Z' denote the number of downsets of G'. We will argue that

$$Z \leq w^{-|E|} Z' \leq Z + w^{-1} 4^{|U|+|V|+|E|}. \tag{4.4}$$

Assuming (4.4), the AP-reduction is correct: with probability at least 3/4 the value q returned by the oracle satisfies $e^{-\varepsilon/2} Z' \leq q \leq e^{\varepsilon/2} Z'$ which implies $e^{-\varepsilon} Z \leq w^{-|E|} q \leq e^{\varepsilon} Z$ as required.

Let X be the partial order with underlying set $\bigcup_{x \in U \cup V \cup E} \{s(u), t(u)\}$ where $x \leq y$ means there is a directed path from x to y in G'. Consider a downset Y of X, and a downset D of G' with $D \cap X = Y$. For each x, we have:

- $D \cap G_x = \emptyset$ if $t(x) \notin Y$, and
- $D \cap G_x = G_x$ if $s(x) \in Y$, and
- otherwise $s(x) \notin Y$ and $t(x) \in Y$, so $D \cap G_x$ can be any downset of G_x with $s(x) \notin D \cap G_x$ and $t(x) \in D \cap G_x$. There are w such sets if $x \in E$, and $w^{\deg(x)}$ such sets if $x \in U \cup V$.

So there are exactly $w^{c(Y)}$ downsets D of G' with $D \cap X = Y$, where

$$c(Y) = \sum_{\substack{v \in U \cup V \\ s(v) \notin Y \\ t(v) \in Y}} \deg(v) + \sum_{\substack{e \in E \\ s(e) \notin Y \\ t(e) \in Y}} 1 = \sum_{(u,v) \in E} \left(\sum_{\substack{s(u) \notin Y \\ t(u) \in Y}} 1 + \sum_{\substack{s(u,v) \notin Y \\ t(u,v) \in Y}} 1 + \sum_{\substack{s(v) \notin Y \\ t(v) \in Y}} 1 \right).$$

Note that the contribution from each $(u, v) \in E$ is at most 1, and $c(Y) = |E|$ if and only if for all $(u, v) \in E$ we have: $s(u) \notin Y$, and $t(u) \in Y \iff s(u, v) \in Y$, and $t(u, v) \in Y \iff s(v) \in Y$, and $t(v) \in Y$. For each independent set I of G let $Y_I \subseteq X$ be the set containing:

- $t(u), s(e_{u,1}), \ldots, s(e_{u,\deg(u)})$ for each $u \in U \cap I$, and
- $t(e_{v,1}), \ldots, t(e_{v,\deg(v)}), s(v)$ for each $v \in V \setminus I$, and
- $t(v)$ for each $v \in V$.

By the previous remarks we have $w^{c(Y_I)-|E|} = 1$, while $w^{c(Y)-|E|} \leq w^{-1}$ for any downset Y of X not of the form Y_I where I is an independent set of G. Since X has cardinality $2(|U| + |V| + |E|)$, it has at most $4^{|U|+|V|+|E|}$ downsets. This gives $Z \leq w^{-|E|} Z' \leq Z + w^{-1} 4^{|U|+|V|+|E|}$, which is (4.4). $\square$

Chapter 5

Degree-two #CSPs with variable weights

(This chapter is a revised version of [83] with a modified introduction.)

This chapter aims to study the computational complexity of approximately evaluating #CSPs where variables range over the Boolean domain $\{0, 1\}$, and we restrict the allowed constraints to a fixed constraint language Γ, and we restrict each variable to appear at most twice, but we allow instances to specify a weight for each value that each variable can take. These problems, degree-two #CSPs with variable weights, are an abstraction of the problem of counting perfect matchings in a graph with edge weights.

5.1 Introduction

Recall Proposition 4.4:

Proposition 4.4. [51, Theorem 24] Let Γ be a finite unweighted constraint language containing $\{(0)\}$ and $\{(1)\}$. Let $d \geq 6$.

- If every relation in Γ is affine then $\#\mathsf{CSP}_{\leq d}(\Gamma) \in \mathrm{FP}$.
- Otherwise, if $\Gamma \subseteq \mathsf{IMconj}$ then $\#\mathsf{CSP}_{\leq d}(\Gamma) =_{\mathrm{AP}} \#\mathsf{BIS}$.
- Otherwise, $\#\mathsf{CSP}_{\leq d}(\Gamma)$ has no FPRAS unless RP = NP.

Note the assumption that Γ contains the "pinning" relations $\{(0)\}$ and $\{(1)\}$. In other words, the instance is allowed to specify that configurations $\mathbf{x}$ with $\mathbf{x}(v) = i$ contribute to the output, for various pairs (v, i). We can interpret this as saying the instance may specify *variable weights* of zero or one: the number of configurations with $\mathbf{x}(v) = i$ is multiplied by a factor of 1, while other configurations are multiplied by a factor of 0. The main result of this chapter is a classification of degree-two #CSPs with unweighted constraint languages, where we allow the instance to specify arbitrary non-negative variable weights.

As mentioned in the introduction, we will be studying the following problem:

Name. $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma)$
Instance. A tuple (V, C, w) where:

- V is a finite set of variables.
- C is a finite set of constraints over Γ on V such that $\deg_C(v) \leq 2$ for all $v \in V$
- w is a function from V to $\mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$, where rationals are specified as ratios of binary integers.

Output. The sum of $\prod_{v \in V} w(v)_{\mathbf{x}(v)}$ over satisfying assignments $\mathbf{x}\colon V \to \{0,1\}$ of C, where we index the pair $w(v)$ from zero: $w(v)_{\mathbf{x}(v)}$ denotes the first element of $w(v)$ when $\mathbf{x}(v) = 0$, and the second element when $\mathbf{x}(v) = 1$.

Here and throughout the chapter we restrict to rational numbers, rather than some larger class of numbers, because rationals are easy to encode as input to a Turing machine.

Yamakami [103] investigated the problem $\mathsf{Holant}^*(\mathcal{F})$, where arbitrary complex-valued arity 1 functions are allowed as part of the instance. Because of the degree bound, variable weights are more invasive than arity 1 functions: the weights are applied to a variable, which is then used in two constraints, whereas an arity 1 function only affects a variable used in one other constraint. However, we are only allowing non-negative variable weights. So variable weights are neither a weaker assumption nor a stronger assumption than the "Holant^*" complex-valued arity 1 functions used in [103].

5.1.1 Delta-matroids and set family notation

To state the main theorem we need to define delta-matroids. Usually, delta-matroids are defined as set families. A *set family* $\mathcal{A}$ is a set of subsets of some finite set (called the ground set). Let $A \triangle B$ denote the symmetric difference of sets A and B. A set family $\mathcal{A}$ is a *delta-matroid* if for all $X, Y \in \mathcal{A}$ and all $i \in X \triangle Y$ there exists $j \in X \triangle Y$, not necessarily distinct from i, such that $X \triangle \{i, j\} \in \mathcal{A}$. For example, setting $X = \emptyset$ and $Y = \{1, 2, 3\}$ and $i = 1$, we see that $\{\emptyset, \{1, 2, 3\}\}$ is not a delta-matroid.

Delta-matroids were used by Feder [52], under the name of generalised matroids. Feder studied the decision version of $\#\mathsf{CSP}_{\leq 2}(\Gamma)$ — the problem of deciding if there is any satisfying assignment.[1] He showed that if the constraint language contains a non-delta-matroid, and the constant relations $\{(0)\}$ and $\{(1)\}$, then the graph constraint satisfaction is equivalent to the unbounded-degree constraint satisfaction problem [52, Theorem 4]. In fact, some of the NP-hardness implied by our main theorem is already implied by Feder's result. We will adapt this work to the setting of counting CSPs.

In this chapter we will call the corresponding relations delta-matroids:

Definition 5.1. We will denote $\{i \mid x_i \neq y_i\}$ by $\mathbf{x} \triangle \mathbf{y}$.

A relation R is a *delta-matroid* if for all $\mathbf{x}, \mathbf{y} \in R$, for all $i \in \mathbf{x} \triangle \mathbf{y}$ there exists $j \in \mathbf{x} \triangle \mathbf{y}$, not necessarily distinct from i, such that $\mathbf{x} \oplus \{i, j\} \in R$.

[1] Actually, Feder studied the variant where every variable is used exactly twice, rather than at most twice.

Note that, for example, $(1,0,0)\triangle(1,1,0) = \{2\}$, while $(1,0,0) \oplus \{1,2\} = (1,0,0) \oplus (1,1,0) = (0,1,0)$. The difference between $\triangle$ and $\oplus$ is whether a set or a configuration is returned.

In particular the perfect matchings relation $\mathrm{PM}_3 = \{(0,0,1),(0,1,0),(1,0,0)\}$ is a delta-matroid, while the arity 3 equality relation $\mathrm{EQ}_3 = \{(0,0,0),(1,1,1)\}$ is not.

5.1.2 Main result

We will refer to two more classes of relations in addition to those of Section 1.7.2.

Definition 5.2. $\mathsf{NEQconj}$ is the set of relations that can be written as a conjunction of equalities, disequalities, and constants.

A relation R is *basically binary* if it is a Cartesian product of relations of arity at most two: there is a partition of $\{1,\ldots,k\}$ into sets $D_1,\ldots,D_\ell$ of order at most two and relations $R_i \subseteq \{0,1\}^{D_i}$ for each $1 \le i \le \ell$ such that

$$R = \{\mathbf{x} \in \{0,1\}^k \mid \mathbf{x}|_{D_i} \in R_i \quad \text{for each} \quad 1 \le i \le \ell\},$$

where $\mathbf{x}|_{D_i}$ denotes the restriction of $\mathbf{x} \in \{0,1\}^k = \{0,1\}^{\{1,\ldots,k\}}$ from $\{1,\ldots,k\}$ to D_i.

For example $\{\mathbf{x} \in \{0,1\}^3 \mid x_1 \ne x_2 \text{ and } x_2 \ne x_3 \text{ and } x_3 = 1\}$ is in $\mathsf{NEQconj}$, and $\{\mathbf{x} \in \{0,1\}^4 \mid x_1 \le x_3 \text{ and } x_2 \le x_4\}$ is basically binary, but $\mathrm{EQ}_3 = \{(0,0,0),(1,1,1)\}$ is not basically binary.

With these definitions we state the main result of this chapter.

Theorem 5.3. *Let Γ be a finite unweighted constraint language. If every relation in Γ is basically binary, or if $\Gamma \subseteq \mathsf{NEQconj}$, then $\#\mathsf{CSP}^{\ge 0}_{\le 2}(\Gamma)$ is in* FP*. Otherwise:*

- *If every relation in Γ is a delta-matroid then $\#\mathsf{PM} \le_{\mathrm{AP}} \#\mathsf{CSP}^{\ge 0}_{\le 2}(\Gamma)$.*
- *If some relation in Γ is not a delta-matroid and $\Gamma \subseteq \mathsf{IMconj}$, then $\#\mathsf{BIS} =_{\mathrm{AP}} \#\mathsf{CSP}^{\ge 0}_{\le 2}(\Gamma)$.*
- *If some relation in Γ is not a delta-matroid and $\Gamma \not\subseteq \mathsf{IMconj}$ then $\#\mathsf{SAT} =_{\mathrm{AP}} \#\mathsf{CSP}^{\ge 0}_{\le 2}(\Gamma)$.*

Theorem 5.3 says that every degree-two Boolean #CSP with variable weights either lies between $\#\mathsf{PM}$ and $\#\mathsf{SAT}$, or is AP-equivalent to $\#\mathsf{BIS}$, or is in FP. Under the assumption that $\#\mathsf{PM}$ and $\#\mathsf{BIS}$ do not have an FPRAS, this classifies completely which problems have an FPRAS.

This is quite a different situation from the corresponding decision problems, considered in [41]. For degree-two decision CSP there is no known dichotomy, and there are many tractable problems using delta-matroids.

Even if $\#\mathsf{PM}$ does have an FPRAS, it might be that there are delta-matroids R such that $\#\mathsf{CSP}^{\ge 0}_{\le 2}(R)$ does not have an FPRAS. Perhaps Theorem 5.3 could be refined to say more about which of these scenarios can be ruled out.

An approximation algorithm necessarily gives a decision algorithm: if a #CSP has an FPRAS, then there is a (randomised) polynomial-time algorithm that determines whether a satisfying assignment exists. The following example shows that the converse does not hold for the problems considered in Theorem 5.3.

Example 5.4. Let R be the relation $\{\mathbf{x} \in \{0,1\}^3 \mid x_1, x_2 \leq x_3\}$. By Schaefer's classification [89] (or reduction to 2-SAT), there is a polynomial-time algorithm that determines whether a list of constraints over $\{R, \mathrm{NAND}\}$ has a satisfying assignment. R is not a delta-matroid, because there is no j such that $(1,1,1) \oplus \{3,j\} \in R$. Also, $R \in \mathsf{IMconj}$, and NAND $\notin \mathsf{IMconj}$. By Theorem 5.3 we have $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(R) =_{\mathrm{AP}} \#\mathsf{BIS}$ and $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(R, \mathrm{NAND}) =_{\mathrm{AP}} \#\mathsf{SAT}$. $\diamond$

5.1.3 Weighted #CSPs and other generalisations

We now discuss some variants of Theorem 5.3. Here we will not restrict to unweighted constraint languages, and we will extend the notation to generalise the degree restrictions and to allow restricted sets of variable weights. For all $K \subseteq \mathbb{N}$ (recall $0 \in \mathbb{N}$), all $W \subseteq \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$, and all finite weighted constraint languages $\mathcal{F}$ define:

Name. $\#\mathsf{CSP}^{W}_{K}(\mathcal{F})$
Instance. A triple (V, C, w) where

- V is a finite set of variables.
- C is a list of constraints over $\mathcal{F}$ on V such that $\deg_C(v) \in K$ for all $v \in V$
- w is a function from V to W, with rationals represented as ratios of binary integers.

Output. The value

$$Z_{V,C,w} = \sum_{\mathbf{x}\colon V \to \{0,1\}} \mathrm{wt}_{V,C,w}(\mathbf{x})$$

where

$$\mathrm{wt}_{V,C,w}(\mathbf{x}) = \left(\prod_{v \in V} w(v)_{\mathbf{x}(v)}\right)\left(\prod_{\langle (s_1,\ldots,s_k),F\rangle \in C} F(\mathbf{x}(s_1),\ldots,\mathbf{x}(s_k))\right).$$

For readability we will use the following **shorthand for K and W**: if K is omitted then $K = \mathbb{N}$ (no degree restriction), if K is "$= d$" then $K = \{d\}$, and if K is "$\leq d$" then $K = \{1,\ldots,d\}$; if W is omitted then $W = \{(1,1)\}$ (no weighting), if W is "≥ 0" then $W = \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$ (arbitrary variable weights). Also, when w is omitted from (V, C, w) it means the unique function $V \to \{(1,1)\}$ — this convention is only used in proofs.

Note that if $\mathcal{F} = \Gamma$ is a set of relations, then $\#\mathsf{CSP}(\Gamma)$, $\#\mathsf{CSP}_{\leq d}(\Gamma)$ and $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma)$ all agree with their earlier definitions.

Example 5.5. Let B be an arity 2 weight-function that is symmetric ($B(0,1) = B(1,0)$).

The problem $\#\mathsf{CSP}^{\{(1,\lambda)\}}(B)$ is then the problem of computing the partition function of a two-state spin system as discussed in Chapter 2. Examples are the Ising model ($B(0,0) = B(1,1)$) and the hardcore gas model ($B = \mathrm{NAND}$). In the ferromagnetic case $B(0,0)B(1,1) \geq B(0,1)B(1,0)$ the problem $\#\mathsf{CSP}(B)$ has an FPRAS [64]. If $B(0,0)B(1,1) < B(0,1)B(1,0)$, then $\#\mathsf{CSP}^{\{(1,\lambda)\}}(B)$ corresponds to the partition function of an anti-ferromagnetic spin system. The research on these problems is a rare example of a (near) dichotomy for approximate counting for weighted problems. See [79] for details. In particular, consider the following problem:

Name. $\#\mathsf{HC}_d(\lambda)$
Instance. A simple graph in which every vertex has degree d.
Output. The sum of $\lambda^{|I|}$ over all independent sets I of G.

For all integers $d \geq 3$ and all rationals $\lambda < (d-1)^{d-1}/(d-2)^d$, the problem $\#\mathsf{HC}_d(\lambda)$ has an FPRAS (in fact an FPTAS). But for $\lambda > (d-1)^{d-1}/(d-2)^d$, there does not exist an FPRAS for $\#\mathsf{HC}_d(\lambda)$ unless $\mathrm{RP} = \mathrm{NP}$ [92, Theorem 1]. Using the correspondence mentioned in Section 1.3, the partition function of the hardcore model with fugacity λ for regular graphs of degree d is AP-equivalent to (a restriction of) $\#\mathsf{CSP}^{\{(1,\lambda)\}}_{=d}(\mathrm{NAND})$. $\diamond$

Thus, in terms of #CSPs:

Proposition 5.6. *[93] For all rationals $\lambda > 4$, there is no FPRAS for $\#\mathsf{CSP}^{\{(1,\lambda)\}}_{=3}(\mathrm{NAND})$ unless* $\mathrm{RP} = \mathrm{NP}$.

The paper [22] studies weighted #CSPs where all (non-negative valued) arity 1 weight-functions are assumed to be in the constraint language. To make this precise, rather than allowing variable weights, the statement of [22, Lemma 16] involves inserting finite sets of arity 1 functions into a given finite constraint language. The classification is in terms of LSM, as defined in the introduction to the thesis, and another class which we will call WNEQ.

Definition 5.7. F' is a *simple weighting* of F if F' is of the form

$$F'(\mathbf{x}) = \lambda F(\mathbf{x}) U_1(x_1) \cdots U_k(x_k)$$

with $\lambda, U_1(0), U_1(1), \ldots, U_k(0), U_k(1) \in \mathbb{Q}_{\geq 0}$. WNEQ is the set of simple weightings of weight-functions in $\mathsf{NEQconj}$.

Proposition 5.8. *[22, Lemma 16] Let $\mathcal{F}$ be a finite weighted constraint language.*

- *If $\mathcal{F} \subseteq \mathsf{WNEQ}$ then $\#\mathsf{CSP}(\mathcal{F})$ is in* FP.
- *Otherwise, there is a finite set of S of weight-functions $\{0,1\}^1 \to \mathbb{Q}_{\geq 0}$ such that*
  - *$\#\mathsf{BIS} \leq_{\mathrm{AP}} \#\mathsf{CSP}(\mathcal{F} \cup S)$, and*
  - *if furthermore $\mathcal{F} \not\subseteq \mathsf{LSM}$ then $\#\mathsf{SAT} =_{\mathrm{AP}} \#\mathsf{CSP}(\mathcal{F} \cup S)$.*

(This statement takes the liberty of restricting to rational numbers, which is justified because the relevant constructions in [22] only use field operations. The original statement in [22] only gives an FPRAS for the first case, but membership in FP follows by inspecting the algorithm used in the proof when $\mathcal{F}$ is rational-valued, or by the algorithm of [29]. The set WNEQ is the same as the set "$\langle \mathrm{NEQ}, \mathcal{B}_1 \rangle$" used in [22], by [22, Remark 14].)

Much like Theorem 5.3 merely gives a reduction from $\#\mathsf{PM}$ in some cases and not an equivalence, Proposition 5.8 merely gives a reduction from $\#\mathsf{BIS}$ in some cases and not an equivalence. The truth could be that $\#\mathsf{BIS}$ has an FPRAS but there exists $F \in \mathsf{LSM}$ such that there is no FPRAS for $\#\mathsf{CSP}(F)$.

5.1.4 Terraced weight-functions

How might Theorem 5.3 generalise to weighted constraint languages? The #CSP result stated here as Proposition 5.8 suggests that WNEQ and LSM can be seen as a generalisation of $\mathsf{NEQconj}$ and IMconj. Basically binary weight-functions can be defined by replacing Cartesian products by tensor products: a function $F\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ is basically binary if there is a partition of $\{1,\ldots,k\}$ into sets $D_1,\ldots,D_\ell$ of order at most two and functions $F_i\colon \{0,1\}^{D_i} \to \mathbb{Q}_{\geq 0}$ for each $1 \leq i \leq \ell$ such that

$$F(\mathbf{x}) = \prod_{i=1}^{\ell} F_i(\mathbf{x}|_{D_i}) \qquad (\mathbf{x} \in \{0,1\}^k).$$

Then we have the following generalisations of the classes used in Theorem 5.3:

Unweighted	Weighted
basically binary	basically binary
$\mathsf{NEQconj}$	WNEQ
IMconj	LSM
delta-matroid	?

Terraced weight-functions give one weighted version of delta-matroids.

Definition 5.9. A *partial configuration* $\mathbf{p}$ of V is an element of $\{0,1\}^U$ for some subset $U \subseteq V$, which we call $\operatorname{dom} \mathbf{p}$.

Two weight-functions $F, G\colon \{0,1\}^V \to \mathbb{Q}_{\geq 0}$ are *parallel* if there are λ, μ not both zero such that for all $\mathbf{x} \in \{0,1\}^V$ we have $\lambda F(\mathbf{x}) = \mu G(\mathbf{x})$. Note that if either F or G is identically zero then F and G are parallel.

A weight-function F of arity k is *terraced* if for all partial configurations $\mathbf{p}$ such that the pinning $F(\mathbf{p}, \cdot)$ is identically zero, for all $i, j \in \operatorname{dom} \mathbf{p}$ the pinnings $F(\mathbf{p} \oplus \{i\}, \cdot)$ and $F(\mathbf{p} \oplus \{j\}, \cdot)$ are parallel.

For example, $\mathrm{EQ}_3 = \{(0,0,0),(1,1,1)\}$ is not terraced: define $\mathbf{p}$ by $p_1 = 0$ and $p_2 = 1$, and set $i = 1$ and $j = 2$.

A relation is terraced if and only if it is a delta-matroid (Lemma 5.27). Terraced weight-functions are thus a weighted version of delta-matroids that allow a generalisation of Theorem 5.3. First, we define the *support* $\mathrm{supp}(F)$ of a weight-function $F\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ to be the relation $\{\mathbf{x} \in \{0,1\}^k \mid F(\mathbf{x}) \neq 0\}$.

Theorem 5.10. *Let $\mathcal{F}$ be a finite weighted constraint language.*

(i.) *If $\mathcal{F} \subseteq \mathsf{WNEQ}$ or every weight-function in $\mathcal{F}$ is basically binary, then $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathcal{F})$ is in* FP.

(ii.) *Otherwise, if there is a non-terraced weight-function in $\mathcal{F}$, then we have a similar classification to Proposition 5.8: $\#\mathsf{BIS} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F})$ and if $\mathcal{F} \not\subseteq \mathsf{LSM}$ then $\#\mathsf{SAT} =_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F})$.*

(iii.) *Otherwise (when neither of the two conditions above hold), if there is a weight-function in $\mathcal{F}$ whose support is not basically binary, then $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F})$.*

Under the assumption that #PM and #BIS do not have an FPRAS, this theorem classifies when there is an FPRAS, except for the case of terraced weight-functions whose support is basically binary.

Example 5.11. Theorem 5.10 says nothing about the problem $\#\mathsf{CSP}^{\geq 0}_{=2}(F)$ where $F\colon \{0,1\}^3 \to \mathbb{Q}_{\geq 0}$ is defined by

$$
\begin{aligned}
F(0,0,0) &= 0 & \qquad F(1,0,0) &= 0 \\
F(0,0,1) &= 1 & \qquad F(1,0,1) &= 1 \\
F(0,1,0) &= 1 & \qquad F(1,1,0) &= 1 \\
F(0,1,1) &= 1 & \qquad F(1,1,1) &= 2.
\end{aligned}
\qquad \diamond
$$

Note that the reductions in Theorem 5.10 are slightly sharper than Theorem 5.3 in one respect. They are stated for $\#\mathsf{CSP}^{\geq 0}_{=2}$, or Holant problems with variable weights: every variable must be used exactly twice.

5.1.5 Other results

The conclusion of Theorem 5.10 can be extended in some cases to allow finite sets of variable weights:

Definition 5.12. A weight-function F is *IM-terraced* if it satisfies the definition of a terraced function whenever "$\mathbf{p}, i, j$" satisfy $p_i \neq p_j$. In full: for all partial configurations $\mathbf{p}$ such that the pinning $F(\mathbf{p}, \cdot)$ is identically zero, for all $i, j \in \mathrm{dom}\,\mathbf{p}$ such that $p_i \neq p_j$, the pinnings $F(\mathbf{p} \oplus \{i\}, \cdot)$ and $F(\mathbf{p} \oplus \{j\}, \cdot)$ are parallel.

The "IM" prefix is meant to suggest that we are ruling out powerful IMconj-like relations, such as $\{(0,0,0),(1,1,1)\}$.

Theorem 5.13. *Let $\mathcal{F}$ be a finite weighted constraint language. Assume $\mathcal{F} \not\subseteq \mathsf{WNEQ}$, and that not every weight-function in $\mathcal{F}$ is basically binary, and that not every weight-function in $\mathcal{F}$ is terraced. (This is the same setting as the #*BIS *and #*SAT *reductions in Theorem 5.10.)*

Unless all the following conditions hold, there is a finite set $W \subseteq \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$ *such that* $\#\mathsf{BIS} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{=2}(\mathcal{F})$*, and if furthermore* $\mathcal{F} \not\subseteq \mathsf{LSM}$ *then* $\#\mathsf{SAT} =_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{=2}(\mathcal{F})$.

(i.) *Every weight-function* $F \in \mathcal{F}$ *is IM-terraced.*

(ii.) *Either the support of every weight-function* F *in* $\mathcal{F}$ *is closed under meets —* $F(\mathbf{x}), F(\mathbf{y}) \neq 0 \implies F(\mathbf{x} \wedge \mathbf{y}) \neq 0$ *— or the support of every weight-function* F *in* $\mathcal{F}$ *is closed under joins —* $F(\mathbf{x}), F(\mathbf{y}) \neq 0 \implies F(\mathbf{x} \vee \mathbf{y}) \neq 0$.

(iii.) *No pinning of the support of a weight-function* $F \in \mathcal{F}$ *is a copy of* EQ_2*, that is, there are no* $\mathbf{x}, i, j$ *satisfying* $x_i = x_j$ *and* $F(\mathbf{x}), F(\mathbf{x} \oplus \{i, j\}) \neq 0$ *and* $F(\mathbf{x} \oplus \{i\}) = F(\mathbf{x} \oplus \{j\}) = 0$.

An interesting class of relations not covered by Theorem 5.13 is the class of monotone relations, for example the relation

$$R = \{(0,0,1), (0,1,1), (1,0,1), (1,1,0), (1,1,1)\}.$$

This is not a delta-matroid: there is no $j \in (0,0,1) \triangle (1,1,0)$ such that $(0,0,1) \oplus \{3, j\} \in R$.

A simple type of reduction we will use is to substitute one constraint by other constraints. We will formalise this process by "K-formulas", a bounded-degree refinement of the pps-formulas used in [22]. The important property is that if G can be expressed by a K-formula over F then $\#\mathsf{CSP}^{W}_{K}(F, G) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{K}(F)$ for any set $\{(1,1)\} \subseteq W \subseteq \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$ — see Lemma 5.21.

An important feature is that Theorem 5.10 cannot be improved by this type of reduction alone. Suppose that some non-terraced function G could be expressed by some terraced function F. Then by Theorem 5.10 and substitution (Lemma 5.21) we would know $\#\mathsf{CSP}^{\geq 0}_{=2}(F) =_{\mathrm{AP}} \#\mathsf{SAT}$, at least if F is neither basically binary nor in $\mathsf{LSM} \cup \mathsf{WNEQ}$. The following result shows that this cannot happen.

Theorem 5.14. *No non-terraced function can be defined by a* (≤ 2)*-formula that only uses terraced functions.*

Feder proved the analogous result for delta-matroids [52, Theorem 4]; a similar result was given by Bouchet and Cunningham [13, Theorem 2.2].

For higher degrees, the situation is quite simple as long as $\mathcal{F}$ contains a weight-function with non-degenerate support, where a relation is *degenerate* if it is a Cartesian product of arity 1 relations.

Theorem 5.15. *Let $\mathcal{F}$ be a finite weighted constraint language and assume that not every weight-function in $\mathcal{F}$ has degenerate support. There exists a finite set of variable weights W such that $\#\mathsf{CSP}^{\geq 0}(\mathcal{F})$ has an FPRAS if and only if $\#\mathsf{CSP}^{W}_{\leq 3}(\mathcal{F})$ has an FPRAS.*

So under these assumptions, by Proposition 5.8, the tractable cases are just what can be computed exactly (unless #BIS has an FPRAS). On the other hand, we show that the tractable region has positive measure, loosely speaking, for all $d \geq 2$:

Theorem 5.16. *Let $d, k \geq 2$. Let F be an arity k weight-function with values in the range $[1, \frac{d(k-1)+1}{d(k-1)-1})$. Then $\#\mathsf{CSP}^{\geq 0}_{\leq d}(F)$ has an FPRAS.*

In Section 5.8 we show that infinite sets of variable weights are necessary in Theorem 5.10, at least unless $\#\mathsf{PM}$ has an FPRAS:

Theorem 5.17. *Let $\mathrm{AtMostOne}_3 = \{\mathbf{x} \in \{0,1\}^3 \mid x_1 + x_2 + x_3 \leq 1\}$. Let W be a finite subset of $\mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$. Then $\#\mathsf{CSP}^{W}_{\leq 2}(\mathrm{AtMostOne}_3)$ has an FPRAS.*

Note that by Theorem 5.3, $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathrm{AtMostOne}_3)$.

5.1.6 Other definitions and notation

The *tensor product* $F \otimes G\colon \{0,1\}^{U \cup V} \to \mathbb{Q}_{\geq 0}$ of $F\colon \{0,1\}^U \to \mathbb{Q}_{\geq 0}$ and $G\colon \{0,1\}^V \to \mathbb{Q}_{\geq 0}$, where U and V are assumed to be disjoint, is defined by $(F \otimes G)(\mathbf{x}, \mathbf{y}) = F(\mathbf{x})G(\mathbf{y})$ for all $\mathbf{x} \in \{0,1\}^U$ and $\mathbf{y} \in \{0,1\}^V$. So if F and G are relations, then $F \otimes G$ is just the Cartesian product and will be denoted $F \times G$. Consider a weight-function $F\colon \{0,1\}^V \to \mathbb{Q}_{\geq 0}$ that is not identically zero. F is *decomposable* if it is the tensor product of at least two weight-functions of arity at least one. Otherwise F is *indecomposable*.

We will sometimes write partial configurations using the notation $\{i \mapsto p_i \mid i \in \operatorname{dom} p\}$. For example $\{i \mapsto c\}$ denotes the unique function $\{i\} \to \{c\}$.

A partial configuration $\mathbf{p}$ is non-empty if $\operatorname{dom} \mathbf{p}$ is non-empty. For a property P of weight-functions, a weight-function F is *pinning-minimal subject to P* if F satisfies P and every pinning of F by a non-empty partial configuration does not satisfy P. A *weight-function pair* is a pair (F, G) where $F, G\colon \{0,1\}^V \to \mathbb{Q}_{\geq 0}$ for some V. We say (F, G) is *pinning-minimal subject to not being parallel* if: F is not parallel to G, and $F(\mathbf{p}, \cdot)$ is parallel to $G(\mathbf{p}, \cdot)$ for any non-empty partial configuration $\mathbf{p}$.

5.2 Reductions

This section establishes some reductions between #CSPs.

We will often implicitly use the fact that $\#\mathsf{CSP}^{W}_{K}(\mathcal{F}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W'}_{K'}(\mathcal{F}')$ whenever $K \subseteq K'$ and $W \subseteq W'$ and $\mathcal{F} \subseteq \mathcal{F}'$. The reduction is to pass through the input to the oracle and return the result.

5.2.1 K-formulas

Another basic reduction uses a simple type of gadget we will call a K-formula, similar to the T-constructability of [103] and refining the pps-definability of [22]. A natural type of gadget for $\#\mathsf{CSP}_K^W(\mathcal{F})$ would allow variable weights. However, this amount of generality is not needed in the reductions later. We will be more restrictive, defining K-formulas without variable weights.

Definition 5.18. Let $\mathcal{F}$ be a weighted constraint language (not necessarily finite). For any $K \subseteq \mathbb{N}$, a *K-formula over $\mathcal{F}$* of arity k is a tuple $\varphi = (V, C, v_1, \ldots, v_k)$ where:

- V is a finite set of variables.
- C is a list of constraints over $\mathcal{F}$ on V.
- $v_1, \ldots, v_k$ are distinct elements of V, called the *external variables*.
- The degree of each internal (i.e. not external) variable is in K.
- Each external variable has degree 1, unless $K = \mathbb{N}$.

φ defines a functions $[\![\varphi]\!] : \{0,1\}^k \to \mathbb{Q}_{\geq 0}$:

$$[\![\varphi]\!] (\mathbf{x}(v_1), \ldots, \mathbf{x}(v_k)) = \sum_{\mathbf{x}|_{V \setminus \{v_1, \ldots, v_k\}}} \prod_{\langle (s_1, \ldots, s_k), F \rangle \in C} F(\mathbf{x}(s_1), \ldots, \mathbf{x}(s_k))$$

for all $\mathbf{x} \colon \{v_1, \ldots, v_k\} \to \{0,1\}$, where the sum is over all extensions of $\mathbf{x}$ to functions $V \to \{0,1\}$.

We will usually specify a K-formula implicitly as a summation of a product. For example, a $\{1,2\}$-formula $(\{x,y,z\}, \{\langle (y,z), F\rangle, \langle (x,y,z), G\rangle\}, x)$ could be specified by:

$$[\![\varphi]\!] (x) = \sum_{y,z} F(y,z) G(x,y,z).$$

The same conventions for the K in $\#\mathsf{CSP}_K$ also apply to the K in K-formulas: $(\leq d)$-formulas are $\{1, \ldots, d\}$-formulas, and $(= d)$-formulas are $\{d\}$-formulas. The degree restriction on the external variables allows substitutions, defined below.

Definition 5.19. For all $K \subseteq \mathbb{N}$, all $W \subseteq \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$ containing $(1,1)$, all sets $\mathcal{F}$ of weight-functions, all K-formulas $\varphi = (V_\varphi, C_\varphi, v_1, \ldots, v_k)$ over $\mathcal{F}$, all weight-functions F of arity k, and all instances $I = (V_I, C_I, w_I)$ of $\#\mathsf{CSP}_K^W(\mathcal{F} \cup \{F\})$, define the *substitution* $I[\varphi/F] = (V_{I[\varphi/F]}, C_{I[\varphi/F]}, w_{I[\varphi/F]})$ as follows.

- $V_{I[\varphi/F]}$ is the disjoint union of V_I and $U_\varphi \times C'$ where U_φ is the set of internal variables of φ, and C' is the set of constraints in C_I that use F. (If there are repeated constraints, C_I should be a set of labels for these constraints.)

- $C_{I[\varphi/F]} = (C_I \setminus C') \cup C''$ where C'' has one constraint $\langle (U_{c,d,1}, \dots, U_{c,d,k_d}), G_d\rangle$ for each $c = \langle (s_{c,1}, \dots, s_{c,k}), F\rangle \in C'$ and each $d = \langle (u_{d,1}, \dots, u_{d,k_d}), G_d\rangle \in C_\varphi$, where

$$U_{c,d,i} = \begin{cases} (u_{d,i}, c) & \text{if } u_{d,i} \notin \{v_1, \dots, v_k\} \\ s_{c,j} & \text{otherwise, where } j \text{ satisfies } u_{d,i} = v_j. \end{cases}$$

- $w_{I[\varphi/F]}(v) = w_I(v)$ for all $v \in V_I$, and $w_{I[\varphi/F]} = (1,1)$ for all $(u,t) \in U_\varphi \times C'$.

The condition that external variables of φ have degree 1 (when $K \neq \mathbb{N}$) again means that $I[\varphi/F]$ is an instance of $\#\mathsf{CSP}_K^W(\mathcal{F})$.

Lemma 5.20. *If $F = [\![\varphi]\!]$ then $Z_I = Z_{I[\varphi/F]}$.*

Proof. We will use the notation from the definition of $I[\varphi/F]$. Let

$$b(\mathbf{x}) = \left(\prod_{v \in V_I} w(v)_{\mathbf{x}(v)}\right)\left(\prod_{\langle (s_1,\dots,s_m),H\rangle \in C_I \setminus C'} H(\mathbf{x}(s_1), \dots, \mathbf{x}(s_m))\right).$$

Then,

$$\begin{aligned} Z_I &= \sum_{\mathbf{x}\colon V_I \to \{0,1\}} \left(b(\mathbf{x}) \prod_{c \in C'} F(\mathbf{x}(s_{c,1}), \dots, \mathbf{x}(s_{c,k}))\right) \\ &= \sum_{\mathbf{x}\colon V_I \to \{0,1\}} \left(b(\mathbf{x}) \prod_{c \in C'} \sum_{\mathbf{x}|_{U_\varphi \times \{c\}}} \prod_{d \in C_\varphi} G_d(\mathbf{x}(U_{c,d,1}), \dots, \mathbf{x}(U_{c,d,k_d}))\right) \end{aligned}$$

where $\sum_{\mathbf{x}|_{U_\varphi \times \{c\}}}$ denotes the sum over all extensions of $\mathbf{x}$ to functions $V \cup (U_\varphi \times \{c\}) \to \{0,1\}$. By the distributivity of multiplication over summation,

$$\begin{aligned} Z_I &= \sum_{\mathbf{x}\colon V_I \cup (U_\varphi \times C') \to \{0,1\}} \left(b(\mathbf{x}|_{V_I}) \prod_{c \in C'} \prod_{d \in C_\varphi} G_d(\mathbf{x}(U_{c,d,1}), \dots, \mathbf{x}(U_{c,d,k_d}))\right) \\ &= Z_{I[\varphi/F]}. \end{aligned}$$

□

Lemma 5.21. *Let $\mathcal{F}$ be a finite weighted constraint language. Let $(1,1) \in W \subseteq \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$. Let $K \subseteq \mathbb{N}$. Let φ be a K-formula. Then $\#\mathsf{CSP}_K^W(\mathcal{F} \cup \{F\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}_K^W(\mathcal{F})$ where $F = [\![\varphi]\!]$.*

Proof. Given I, the reduction calls the oracle on $I[\varphi/F]$, without changing the error parameter, and returns the result. This is a correct AP-reduction by Lemma 5.20. □

5.2.2 h-maximisation

We now discuss a reduction which is an important step in the proof of Theorem 5.3: it shows that we can delete configurations in certain sense.

Definition 5.22. For all $k \geq 0$, all $h \in \mathbb{Z}^k$ and all $F\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ not identically zero, define the *h-maximisation* $F_{h-\max}\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ *of* F by setting

$$F_{h-\max}(\mathbf{x}) = \begin{cases} F(\mathbf{x}) & \text{if } \sum_{i=1}^k x_i h_i = \max_{\mathbf{y}\in\mathrm{supp}(F)} \sum_{i=1}^k y_i h_i, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 5.23. *Let $\mathcal{F}$ be a finite weighted constraint language. Let $W = \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$ or $W = \{(2^a, 2^b) \mid a, b \in \mathbb{Z}\}$. Then for every $F \in \mathcal{F}$, and every $h \in \mathbb{Z}^k$ where k is the arity of F, we have*

$$\#\mathsf{CSP}_K^W(\mathcal{F} \cup \{F_{h-\max}\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}_K^W(\mathcal{F}).$$

Proof. The reduction is given an instance (V, C, w) of $\#\mathsf{CSP}_K^W(\mathcal{F} \cup \{F_{h-\max}\})$ and error parameter $\varepsilon > 0$ which we can assume is less than $1/2$. We wish to compute a value Z such that $e^{-\varepsilon} Z_{V,C,w} \leq Z \leq e^{\varepsilon} Z_{V,C,w}$.

Let $s = |V| + |C|$. Let M be the maximum over: the value 1, the values taken by weight-functions in $\mathcal{F}$, and the values $w(v)_i$ for all v, i. Let m be the minimum over: the value 1, the non-zero values taken by weight-functions in $\mathcal{F}$, and the values $w(v)_i$ for all v, i such that $w(v)_i \neq 0$. Let $n = \lceil |V| + s \log_2 M - \log_2(m^s \varepsilon/4) \rceil$. Note that n is polynomially bounded in the size of the input, and $2^{|V| + s\log_2 M - n} \leq (\varepsilon/4) m^s$. Let $H = \max_{\mathbf{y}\in\mathrm{supp}(F)} \sum_{i=1}^k y_i h_i$. Define $F_n\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ by

$$F_n(\mathbf{x}) = F(\mathbf{x}) 2^{n(\sum_{i=1}^k x_i h_i - H)} \qquad (\mathbf{x} \in \{0,1\}^k).$$

Note that for all $\mathbf{x}$ either: $\sum_{i=1}^k x_i h_i = H$ so $F_n(\mathbf{x}) = F_{h-\max}(\mathbf{x}) = F(\mathbf{x})$, or $\sum_{i=1}^k x_i h_i < H$ so $F_{h-\max}(\mathbf{x}) = 0$ and $F_n(\mathbf{x}) \leq M 2^{-n}$. Let C_n be the result of replacing $F_{h-\max}$ by F_n in C.

$Z_{V,C_n,w}$ can be approximated using the oracle as follows. First, let C' be the result of replacing $F_{h-\max}$ by F in C. Then define $w'\colon V \to W$ by $w'(v)_0 = w(v)_0$ and $w'(v)_1 = w(v)_1 2^{nh(v)}$ where $h(v)$ is the sum of h_i over all pairs $(\langle (s_1, \ldots, s_k), F_{h-\max}\rangle, i) \in C \times \mathbb{N}$ with $s_i = v$. So every time that v is used in position i in an $F_{h-\max}$ constraint, we get a contribution of h_i to $h(v)$. Then $Z_{V,C',w'} = Z_{V,C_n,w} 2^{nHt}$ where t is the number of constraints in C using $F_{h-\max}$. Call the oracle on (V, C', w') with error parameter $\varepsilon/2$ and divide the result by 2^{nHt} to obtain a value Z' such that $e^{-\varepsilon/2} Z_{V,C_n,w} \leq Z' \leq e^{\varepsilon/2} Z_{V,C_n,w}$ with probability at least $3/4$.

We will argue that the following algorithm is a correct AP-reduction: compute Z' as above; if $Z' < m^s/4$ then output $Z = 0$, and otherwise output $Z = Z'$. It suffices to show that $e^{-\varepsilon/2} Z_{V,C_n,w} \leq Z' \leq e^{\varepsilon/2} Z_{V,C_n,w}$ implies $e^{-\varepsilon} Z_{V,C,w} \leq Z \leq e^{\varepsilon} Z_{V,C,w}$.

For all configurations $\mathbf{x}$, if $\mathrm{wt}_{V,C,w}(\mathbf{x}) \neq \mathrm{wt}_{V,C_n,w}(\mathbf{x})$ then $\mathrm{wt}_{V,C,w}(\mathbf{x}) = 0$ and $\mathrm{wt}_{V,C_n,w}(\mathbf{x}) \leq M^s 2^{-n}$. Hence

$$|Z_{V,C,w} - Z_{V,C_n,w}| \leq 2^{|V| + s\log_2 M - n} \leq m^s(\varepsilon/4).$$

If $Z_{V,C,w} = 0$ then $Z' \leq 2Z_{V,C_n,w} \leq 2m^s(\varepsilon/4) < m^s/4$, so the algorithm correctly outputs zero. Otherwise, $Z_{V,C,w} \neq 0$, so $Z' > Z_{V,C_n,w}/2 \geq (Z_{V,C,w} - m^s(\varepsilon/4))/2 \geq m^s/4$. In this case $|Z_{V,C,w} - Z_{V,C_n,w}| \leq Z_{V,C,w}(\varepsilon/4)$, and since $e^{-\varepsilon/2} \leq 1 - \varepsilon/4$ we have

$$\begin{aligned}(1-\varepsilon/4)Z_{V,C,w} \leq Z_{V,C_n,w} &\leq (1+\varepsilon/4)Z_{V,C,w}\\ e^{-\varepsilon/2}Z_{V,C,w} \leq Z_{V,C_n,w} &\leq e^{\varepsilon/2}Z_{V,C,w}\\ e^{-\varepsilon}Z_{V,C,w} \leq Z' &\leq e^{\varepsilon}Z_{V,C,w}.\end{aligned}$$ □

5.2.3 Other reductions

We will use variable-weighted versions of known reductions.

Lemma 5.24. *Let $\mathcal{F}$ be a finite weighted constraint language.*

(i.) *If $\mathcal{F} \subseteq \mathsf{WNEQ}$ then $\#\mathsf{CSP}^{\geq 0}(\mathcal{F}) \in \mathrm{FP}$ [29]. If every weight-function in $\mathcal{F}$ is basically binary then $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathcal{F}) \in \mathrm{FP}$ [27, Theorem 2.2].*

(ii.) *If $\mathcal{F} \subseteq \mathsf{IMconj}$ then $\#\mathsf{CSP}^{\geq 0}(\mathcal{F}) \leq \#\mathsf{BIS}$.*

(iii.) $\#\mathsf{CSP}^{\geq 0}(\mathcal{F}) \leq_{\mathrm{AP}} \#\mathsf{SAT}$.

(iv.) $\#\mathsf{CSP}^{\geq 0}(\mathcal{F}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathcal{F} \cup \{\mathrm{EQ}_3\})$.

Proof. (i.) is a straightforward modification of the cited algorithms to include variable weights. In (ii.) and (iii.), by scaling we can assume all the variable weights and values taken by weight-functions are in fact integers.

For (ii.), we have $\#\mathsf{CSP}(\mathrm{IMP}, \mathrm{PIN}_0, \mathrm{PIN}_1) \leq_{\mathrm{AP}} \#\mathsf{BIS}$ by Proposition 4.1 ([48, Theorem 3]), so it suffices to show that $\#\mathsf{CSP}^{\geq 0}(\mathrm{IMP}, \mathrm{PIN}_0, \mathrm{PIN}_1) \leq_{\mathrm{AP}} \#\mathsf{CSP}(\mathrm{IMP}, \mathrm{PIN}_0, \mathrm{PIN}_1)$. Since there is no degree bound, arity 1 functions can be used as variable weights. The construction in [22, Proposition 25] defines any positive-integer-valued arity 1 weight-function F in polynomial time in $\lceil \log(F(0)) + \log(F(1)) \rceil$, as a $\mathbb{N}$-formula. So we can simulate any variable weight.

For (iii.), the problem of evaluating a #CSP, with explicit integer-valued weight-functions as part of the input, is in $\#\mathsf{P}$ and hence AP-reduces to $\#\mathsf{SAT}$ - see the remarks in Section 3 of [46].

(iv.) is called "2-simulating equality" in [51]. Since this reduction is an important part of Theorem 5.3, we will give details here.

For all $k \geq 3$ let φ_k be the $(=2)$-formula over $\{\mathrm{EQ}_3\}$ on the variable set $\{u_1, \ldots, u_k, v_1, \ldots, v_k\}$ with external variables $(v_1, \ldots, v_k)$ and constraints $\langle (u_{i-1}, v_i, u_i), \mathrm{EQ}_3 \rangle$ for $1 \leq i \leq k$, where u_0 means u_k. Then $[\![\varphi_k]\!] = \mathrm{EQ}_k$.

The reduction is given an instance (V, C, w) of $\#\mathsf{CSP}^{\geq 0}(\Gamma)$ and an error parameter ε. For each variable $v \in V$ of degree $\deg_C(v) > 2$, replace all but one of its occurrences in C by distinct new variables $v_2, \ldots, v_{\deg_C(v)}$ and add a new constraint $\langle (v, v_2, \ldots, v_{\deg_C(v)}), \mathrm{EQ}_{\deg_C(v)} \rangle$. Call the new variable set V' and the new constraint list

C'. Extend w to $w' \colon V' \to \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$ by setting $w(v_i) = (1,1)$ for all $v_i \in V' \setminus V$. Then $Z_{V',C',w'} = Z_{V,C,w}$. Let (V'', C'', w'') be the repeated substitution

$$(V', C', w')[\varphi_4/\mathrm{EQ}_4] \dots [\varphi_D/\mathrm{EQ}_D]$$

where $D = \max_{v \in V} \deg_C(v)$. This substitution can easily be computed in polynomial time. By Lemma 5.20 we have $Z_{V'',C'',w''} = Z_{V',C',w'} = Z_{V,C,w}$, so the reduction can just call the oracle on $Z_{V'',C'',w''}$ with error parameter ε and return the result. □

Pinning is very useful throughout.

Lemma 5.25. *Let $\mathcal{F}$ be a finite weighted constraint language. Let $F' \colon \{0,1\}^m \to \mathbb{Q}_{\geq 0}$ be a copy of a pinning of a weight-function $F \in \mathcal{F}$. Then $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathcal{F} \cup \{F', \mathrm{PIN}_0, \mathrm{PIN}_1\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathcal{F})$.*

Proof. F' is $[\![\varphi]\!]$, for some $(=2)$-formula φ over $\{F, \mathrm{PIN}_0, \mathrm{PIN}_1\}$. Specifically, let k be the arity of F. Permuting F' if necessary, there are indices $1 \leq i_1 < \dots < i_m \leq k$ and constants $x_i \in \{0,1\}$ for $i \in \{1, \dots, k\} \setminus \{i_1, \dots, i_m\}$ such that

$$F'(x_{i_1}, \dots, x_{i_m}) = F(x_1, \dots, x_k) \qquad \text{for all } x_{i_1}, \dots, x_{i_m} \in \{0,1\}.$$

Thus $F' = [\![\varphi]\!]$ where φ is the $(=2)$-formula on the variable set $\{v_1, \dots, v_k\}$, with external variables $(v_{i_1}, \dots, v_{i_m})$, consisting of a constraint $\langle (v_1, \dots, v_k), F \rangle$, and a constraint $\langle (v_i), \mathrm{PIN}_{x_i} \rangle$ for each $i \in \{1, \dots, k\} \setminus \{i_1, \dots, i_m\}$.

Therefore, by substitution (Lemma 5.21), it suffices to show $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathcal{F} \cup \{\mathrm{PIN}_0, \mathrm{PIN}_1\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathcal{F})$. Given an instance (V, C, w) of $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathcal{F} \cup \{\mathrm{PIN}_0, \mathrm{PIN}_1\})$, let C' be the list of constraints in C not using PIN_0 or PIN_1, and define $w' \colon W \to \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$ by

$$w'(v)_i = \begin{cases} 0 & \text{if } \langle (v), \mathrm{PIN}_{1-i} \rangle \in C, \\ w(v)_i & \text{otherwise.} \end{cases}$$

Then $\mathrm{wt}_{V,C,w}$ agrees with $\mathrm{wt}_{V,C',w'}$ on all configurations, so $Z_{V,C,w} = Z_{V,C',w'}$, and (V, C', w') is an instance of $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathcal{F})$. □

We can convert between finite sets of variable weights and simple weightings. This is useful for Theorems 5.13 and 5.15.

Lemma 5.26. *Let K be a finite non-empty set of integers. Let $\mathcal{F}$ be a finite weighted constraint language.*

1. *Let $\mathcal{G}$ be a finite set of simple weightings of weight-functions in $\mathcal{F}$. There is a finite set $W \subseteq \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$ such that $\#\mathsf{CSP}_K(\mathcal{G}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^W_K(\mathcal{F})$.*
2. *For all finite sets $W \subseteq \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$ there is a finite set $\mathcal{G}$ of simple weightings of weight-functions in $\mathcal{F}$ such that $\#\mathsf{CSP}^W_K(\mathcal{F}) \leq_{\mathrm{AP}} \#\mathsf{CSP}_K(\mathcal{G})$.*

Proof. (1.) Each $G \in \mathcal{G}$ can be expressed as $G(\mathbf{x}) = \lambda_G F_G(\mathbf{x}) \prod_{j=1}^{k_G} U_{G,j}(x_j)$ for some $F_G \in \mathcal{F}$ and some $\lambda_G, U_{G,1}(0), U_{G,1}(1), \ldots, U_{G,k_G}(0), U_{G,k_G}(1) \geq 0$ where k_G is the arity of G. Given a function n taking pairs (G, j), $G \in \mathcal{G}$, $1 \leq j \leq k_G$, to integers $0 \leq n_{G,j} \leq \max(K)$, define

$$w_n = \left(\prod_{G \in \mathcal{G}, 1 \leq j \leq k_G} U_{G,j}(0)^{n_{G,j}}, \prod_{G \in \mathcal{G}, 1 \leq j \leq k_G} U_{G,j}(1)^{n_{G,j}} \right) \in \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}.$$

Let W be the set of pairs w_n where n ranges over all $(\max(K)+1)^{\sum_{G \in \mathcal{G}} k_G}$ choices of the integers $n_{G,j}$.

Given an instance (V, C) of $\#\mathsf{CSP}_K(\mathcal{G})$, let C' be the list with one constraint $\langle (v_1, \ldots, v_k), F_G \rangle$ for each constraint $\langle (v_1, \ldots, v_k), G \rangle \in C$. Define $w \colon V \to W$ by $w(v) = w_{n(x)}$, where $n(x)_{G,j}$ is the number of constraints in C using G and using v in the j position: $\langle (v_1, \ldots, v_k), G \rangle \in C$ with $v_j = v$. This ensures that $\mathrm{wt}_{V,C}(\mathbf{x}) = \mu \cdot \mathrm{wt}_{V,C',w}(\mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^V$, where $\mu = \prod_{\langle (v_1,\ldots,v_k), G \rangle \in C} \lambda_G$. So the reduction can just query the oracle with (V, C', w), passing the error parameter to the oracle, and divide by μ.

(2.) For all $F \colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ and all $w \in (W \cup \{(1,1)\})^k$, define

$$G_{F,w}(\mathbf{x}) = F(\mathbf{x}) \prod_{i=1}^{k} w(i)_{x_i} \qquad (\mathbf{x} \in \{0,1\}^k).$$

Let $\mathcal{G}$ be the set of weight-functions of the form $G_{F,w}$ with $F \in \mathcal{F}$.

We are given an instance (V, C, w) of $\#\mathsf{CSP}_K^W(\mathcal{F})$. Let $V' = \{v \in V \mid \deg_C(v) > 0\}$. Let $g \colon V' \to C$ be any map taking each variable $v \in V'$ to the index of a constraint with v in its scope: if $g(v) = \langle (v_1, \ldots, v_k), G \rangle$ then $v = v_i$ for some i. Let C' be the list with one constraint $\langle (v_1, \ldots, v_k), G_{F,w_F} \rangle$ for each constraint $c = \langle (v_1, \ldots, v_k), F \rangle \in C$, where

$$w_F(i) = \begin{cases} w(v_i) & \text{if } g(v_i) = c, \text{ and} \\ (1,1) & \text{otherwise.} \end{cases}$$

This ensures that $\mathrm{wt}_{V',C,w}(\mathbf{x}) = \mathrm{wt}_{V',C'}(\mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^{V'}$, so $Z_{V,C,w} = Z_{V',C'} \cdot \mu$ where $\mu = \prod_{v \in V \setminus V'} (w(v)_0 + w(v)_1)$, so the reduction can just call the $\#\mathsf{CSP}_K(\mathcal{G})$ oracle on $Z_{V',C'}$, passing through the error parameter, then multiply by the result by μ. $\square$

5.3 Main theorem

We first establish some useful properties of delta-matroids.

Lemma 5.27. *A relation is a delta-matroid if and only if it is terraced.*

(Recall that a relation R is a *delta-matroid* if for all $\mathbf{x}, \mathbf{y} \in R$ and for all $i \in \mathbf{x} \triangle \mathbf{y}$ there exists $j \in \mathbf{x} \triangle \mathbf{y}$, not necessarily distinct from i, such that $\mathbf{x} \oplus \{i, j\} \in R$. A weight-function F is *terraced* if for all partial configurations $\mathbf{p}$ of V and all $i, j \in \operatorname{dom} \mathbf{p}$, if $F(\mathbf{p}, \cdot)$ is identically zero then $F(\mathbf{p} \oplus \{i\}, \cdot)$ and $F(\mathbf{p} \oplus \{j\}, \cdot)$ are parallel.)

Proof. Let R be a delta-matroid. Let $\mathbf{p}$ be a partial configuration such that $R(\mathbf{p}, \cdot)$ is empty and let $i, j \in \operatorname{dom} \mathbf{p}$ be indices such that $R(\mathbf{p} \oplus \{i\}, \cdot)$ and $R(\mathbf{p} \oplus \{j\}, \cdot)$ are non-empty. We will show that $R(\mathbf{p} \oplus \{i\}, \cdot) = R(\mathbf{p} \oplus \{j\}, \cdot)$. By symmetry it suffices to show that for all $\mathbf{x} \in R(\mathbf{p} \oplus \{i\}, \cdot)$ we have $\mathbf{x} \in R(\mathbf{p} \oplus \{j\}, \cdot)$. Pick $\mathbf{y} \in R(\mathbf{p} \oplus \{j\}, \cdot)$. By the delta-matroid property applied to $((\mathbf{p} \oplus \{i\}, \mathbf{x}), (\mathbf{p} \oplus \{j\}, \mathbf{y}), i)$ there exists d, such that $x_d \neq y_d$ or $d \in \{i, j\}$, and such that $(\mathbf{p} \oplus \{i\}, \mathbf{x}) \oplus \{i, d\}$ is in R. Since $R(\mathbf{p}, \cdot)$ is empty we have $d = j$ and hence $\mathbf{x} \in R(\mathbf{p} \oplus \{j\}, \cdot)$.

Conversely let R be a relation which is terraced when considered as a weight-function. For all $\mathbf{x}, \mathbf{y} \in R$ and all $d \in \mathbf{x} \triangle \mathbf{y}$ we wish to show that $\mathbf{x} \oplus \{d, d'\} \in R$ for some $d' \in \mathbf{x} \triangle \mathbf{y}$. Let $\mathbf{y}' \in R$ satisfy $\{d\} \subseteq \mathbf{x} \triangle \mathbf{y}' \subseteq \mathbf{x} \triangle \mathbf{y}$ with $|\mathbf{x} \triangle \mathbf{y}'|$ minimal. If $\mathbf{x} \triangle \mathbf{y}' = \{d\}$ we can take $d' = d$. Otherwise pick $d' \in (\mathbf{x} \triangle \mathbf{y}') \setminus \{d\}$. Let $\mathbf{p}$ be the restriction of $\mathbf{x} \oplus \{d\}$ to $\{d, d'\} \cup \{i \mid x_i = y'_i\}$. Configurations $\mathbf{z} \in R(\mathbf{p}, \cdot)$ satisfy $\{d\} \subseteq \mathbf{x} \triangle (\mathbf{p}, \mathbf{z}) \subseteq (\mathbf{x} \triangle \mathbf{y}') \setminus \{d'\}$, but $|\mathbf{x} \triangle (\mathbf{p}, \mathbf{z})| < |\mathbf{x} \triangle \mathbf{y}'|$ contradicts the choice of $\mathbf{y}'$; therefore $R(\mathbf{p}, \cdot)$ is empty. And $R(\mathbf{p} \oplus \{d\}, \cdot)$ and $R(\mathbf{p} \oplus \{d'\}, \cdot)$ contain the restrictions of $\mathbf{x}$ and $\mathbf{y}$ respectively (to $(\mathbf{x} \triangle \mathbf{y}') \setminus \{d, d'\}$). Since R has a terraced weight-function, $R(\mathbf{p} \oplus \{d\}, \cdot) = R(\mathbf{p} \oplus \{d'\}, \cdot)$ so $\mathbf{x} \oplus \{d, d'\} \in R$. $\square$

The following argument is useful for studying pinnings.

Lemma 5.28. *Let (F, G) be a weight-function pair that is pinning-minimal subject to not being parallel. Then* $\operatorname{supp}(F) \cup \operatorname{supp}(G) = \{\mathbf{x}, \overline{\mathbf{x}}\}$ *where* $\mathbf{x} \in \operatorname{supp}(F)$ *and* $\overline{\mathbf{x}} \in \operatorname{supp}(G)$*.*

Proof. First we give another characterisation of when a weight-function pair is parallel. For any $F, G \colon \{0,1\}^V \to \mathbb{Q}_{\geq 0}$ consider the two-by-$2^{|V|}$ matrix M, with columns indexed by $\{0,1\}^V$, defined by $M_{1,\mathbf{x}} = F(\mathbf{x})$ and $M_{2,\mathbf{x}} = G(\mathbf{x})$. The weight-function pair is non-parallel if and only if M has row rank two, hence if and only if M has column rank two, and hence if and only if there exist $\mathbf{x}, \mathbf{y}$ such that the two-by-two submatrix

$$M(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} F(\mathbf{x}) & F(\mathbf{y}) \\ G(\mathbf{x}) & G(\mathbf{y}) \end{pmatrix}$$

has linearly independent rows.

Now let (F, G) be a weight-function pair that is pinning-minimal subject to not being parallel. For any $(\mathbf{x}, \mathbf{y})$ such that $M(\mathbf{x}, \mathbf{y})$ has linearly independent rows, let $\mathbf{p} = \{i \mapsto x_i \mid x_i = y_i\}$. Then $F(\mathbf{p}, \cdot)$ is not parallel to $G(\mathbf{p}, \cdot)$. But (F, G) is pinning-minimal subject to not being parallel, so $\mathbf{y} = \overline{\mathbf{x}}$.

There exists some $\mathbf{x}$ such that $M(\mathbf{x}, \overline{\mathbf{x}})$ has linearly independent columns. For all $\mathbf{z}$ such that $F(\mathbf{z})$ or $G(\mathbf{z})$ is non-zero, $(F(\mathbf{z}), G(\mathbf{z})) \in \mathbb{R}^2$ cannot be a multiple of both $(F(\mathbf{x}), G(\mathbf{x}))$ and $(F(\overline{\mathbf{x}}), G(\overline{\mathbf{x}}))$, so either $M(\mathbf{x}, \mathbf{z})$ has linearly independent columns or $M(\mathbf{z}, \overline{\mathbf{x}})$ has linearly independent columns. By the previous paragraph, $\mathbf{z} = \overline{\mathbf{x}}$ or $\mathbf{z} = \overline{\overline{\mathbf{x}}} = \mathbf{x}$. Hence $\operatorname{supp}(F) \cup \operatorname{supp}(G) \subseteq \{\mathbf{x}, \overline{\mathbf{x}}\}$. Finally, since F and G are not parallel, if one of $F(\mathbf{x})$ and $G(\overline{\mathbf{x}})$ is zero, then $F(\overline{\mathbf{x}})$ and $G(\mathbf{x})$ are both non-zero. $\square$

To get a reduction from #PM to a degree-two #CSP whose constraint language consists of delta-matroids, we will use certain pinnings.

Lemma 5.29. *Let R be a delta-matroid that is not basically binary. There is a copy $R'\colon \{0,1\}^3 \to \mathbb{Q}_{\geq 0}$ of an arity 3 pinning of R such that for some $h\colon \{1,2,3\} \to \mathbb{Z}$ and some $\mathbf{x} \in \{0,1\}^3$,*

$$R'_{h-\max} = \{(1-x_1, x_2, x_3), (x_1, 1-x_2, x_3), (x_1, x_2, 1-x_3)\}.$$

Proof. We will show that some pinning $R'\colon \{0,1\}^V \to \mathbb{Q}_{\geq 0}$ of R with $|V| = 3$ has the following property. The conclusion of the lemma then follows after choosing a bijection from V to $\{1,2,3\}$.

$$\text{There exists } \mathbf{y} \in \{0,1\}^V \text{ and } d \in \{1,2\} \text{ such that: } \mathbf{y} \oplus U \notin R' \text{ for each subset } U \subseteq V \text{ with } |U| < d, \text{ and } \mathbf{y} \oplus U \in R' \text{ for each subset } U \subseteq V \text{ with } |U| = d. \tag{SP}$$

Given (SP), let $h(i) = 2y_i - 1$ for all $i \in V$. Observe that $R'_{h-\max}$ consists precisely of the three configurations $\mathbf{x} \oplus U$ with $|U| = d$. If $d = 1$, we may take $\mathbf{x} = \mathbf{y}$, and if $d = 2$ we may take $\mathbf{x} = \overline{\mathbf{y}}$.

It remains to establish (SP). We may assume that R is indecomposable. (If $R = S \times T$ for some S and T, then S and T cannot both be basically binary, but both S and T are pinnings of R.) There exists a configuration not in R (otherwise R would be basically binary). In other words, there is an arity zero pinning of R that is the empty relation. Let $R(\mathbf{p}, \cdot)$ be an inclusion-maximal pinning (so $\operatorname{dom} \mathbf{p}$ is minimal) subject to $R(\mathbf{p}, \cdot) = \emptyset$. For each $v \in \operatorname{dom} \mathbf{p}$ the pinning $R(\mathbf{p}|_{\operatorname{dom} \mathbf{p} \setminus \{v\}}, \cdot)$ is non-empty by maximality of $R(\mathbf{p}, \cdot)$. Hence the relations $R(\mathbf{p} \oplus \{v\}, \cdot)$ are non-empty. The weight-function of R is terraced by Lemma 5.27, so $R(\mathbf{p} \oplus \{v\}, \cdot) = R(\mathbf{p} \oplus \{v'\}, \cdot)$ for any $v, v' \in \operatorname{dom} \mathbf{p}$.

If $\operatorname{dom} \mathbf{p} = \{v\}$ for some v then R is the product of $\{1 - p_v\}$ with $R(\mathbf{p} \oplus \{v\}, \cdot)$, which contradicts the indecomposability of R.

If $|\operatorname{dom} \mathbf{p}| \geq 3$, pick $\mathbf{z} \in R(\mathbf{p} \oplus \{v\}, \cdot)$ (for any v) and pick a subset D of order 3 of $\operatorname{dom} \mathbf{p}$. Let $R'\colon \{0,1\}^D \to \mathbb{Q}_{\geq 0}$ be the pinning of R by $(\mathbf{p}|_{\operatorname{dom} \mathbf{p} \setminus D}, \mathbf{z})$. Note that $\mathbf{p}|_D \notin R'$ but $\mathbf{p}|_D \oplus \{v\} \in R'$ for all variables $v \in D$. Hence (SP) holds with $d = 1$.

The remaining case is that $\operatorname{dom} \mathbf{p} = \{i, j\}$ for some distinct variables i, j. Since R is indecomposable, R is not the product of $R(\mathbf{p} \oplus \{i\}, \cdot)$ with $\{\mathbf{p} \oplus \{i\}, \mathbf{p} \oplus \{j\}\}$ or $\{\mathbf{p} \oplus \{i\}, \mathbf{p} \oplus \{j\}, \mathbf{p} \oplus \{i,j\}\}$. Hence $R(\mathbf{p} \oplus \{i\}, \cdot)$ and $R(\mathbf{p} \oplus \{i,j\}, \cdot)$ are not parallel. Let R' be a pinning of R such that $(R'(\mathbf{p} \oplus \{i\}, \cdot), R'(\mathbf{p} \oplus \{i,j\}, \cdot))$ is pinning-minimal subject to not being parallel. By Lemma 5.28 we have $R'(\mathbf{p} \oplus \{i\}, \cdot) \cup R'(\mathbf{p} \oplus \{i,j\}, \cdot) = \{\mathbf{z}, \overline{\mathbf{z}}\}$ where $\mathbf{z} \in R'(\mathbf{p} \oplus \{i\}, \cdot)$ and $\overline{\mathbf{z}} \in R'(\mathbf{p} \oplus \{i,j\}, \cdot)$. Also, to recap: $R'(\mathbf{p}, \cdot) = \emptyset$ and $R'(\mathbf{p} \oplus \{i\}, \cdot) = R'(\mathbf{p} \oplus \{j\}, \cdot) \neq R'(\mathbf{p} \oplus \{i,j\}, \cdot)$.

If $\overline{\mathbf{z}} \notin R'(\mathbf{p} \oplus \{i\}, \cdot)$ then by the delta-matroid property applied to $(\mathbf{p} \oplus \{i,j\}, \overline{\mathbf{z}})$, $(\mathbf{p} \oplus \{i\}, \mathbf{z})$ and j there exists $k \in \{j\} \cup \operatorname{dom} \mathbf{z}$ such that $(\mathbf{p} \oplus \{i,j\}, \overline{\mathbf{z}}) \oplus \{j,k\} \in R'$, but then k must lie in $\operatorname{dom} \mathbf{z}$ and $\overline{\mathbf{z}} \oplus \{k\} \in R'(\mathbf{p} \oplus \{j\}, \cdot)$. Hence $|\operatorname{dom} \mathbf{z}| = 1$, and R' has arity 3, and (SP) holds with $\mathbf{x} = (\mathbf{p}, \overline{\mathbf{z}})$ and $d = 2$. Otherwise $\mathbf{z} \notin R'(\mathbf{p} \oplus \{i,j\}, \cdot)$.

Then, by the delta matroid property applied to $(\mathbf{p} \oplus \{i\}, \mathbf{z})$, $(\mathbf{p} \oplus \{i, j\}, \overline{\mathbf{z}})$ and j, there exists $k \in \{j\} \cup \operatorname{dom} \mathbf{z}$ such that $(\mathbf{p} \oplus \{i\}, \mathbf{z}) \oplus \{j, k\} \in R'$, but then k must lie in $\operatorname{dom} \mathbf{z}$ and $\mathbf{z} \oplus \{k\} \in R'(\mathbf{p} \oplus \{i, j\}, \cdot)$. Hence $|\operatorname{dom} \mathbf{z}| = 1$, and R' has arity 3, and (SP) holds with $\mathbf{z} = (\mathbf{p} \oplus \{i, j\}, \mathbf{z})$ and $d = 1$. $\square$

The following observation is important for getting a complete classification in Theorem 5.3.

Lemma 5.30. *Let R be a delta-matroid in* IMconj. *Then R is basically binary.*

Proof. We may assume that R is indecomposable, because a Cartesian product of basically binary relations is basically binary. Assume for contradiction that R has arity at least three.

Let V be the variable set of R. Note that no variables are pinned: if there exists $i \in V$ and $c \in \{0, 1\}$ such that $x_i = c$ for all $\mathbf{x} \in R$, then R is the product of $\{c\}$ with the pinning of R by $\{i \mapsto c\}$ (the unique function $\{i\} \to \{c\}$), but this contradicts the assumption that R is indecomposable. Since R is in IMconj and no variables are pinned, R is a conjunction of implications of variables. Therefore there is a subset P of $V \times V$ such that

$$R = \{\mathbf{x} \mid x_i \leq x_j \text{ for all } (i, j) \in P\}.$$

Consider the undirected graph G on V where i and j are adjacent if and only if (i, j) or (j, i) is in P. Then G has at least three vertices, and since R is indecomposable, G is connected. Hence there is a vertex i of degree at least two. There exist distinct variables $j, k \in V$ such that $(i, j), (i, k) \in P$, or $(j, i), (k, i) \in P$, or $(j, i), (i, k) \in P$. In the first case, there is no $\ell \in V$ such that $\underline{0} \oplus \{i, \ell\} \in R$. In the second case, there is no $\ell \in V$ such that $\underline{1} \oplus \{i, \ell\} \in R$. In the third case, there is no $\ell \in V$ such that $\underline{0} \oplus \{j, \ell\} \in R$. But the all-zero configuration $\underline{0}$ and the all-one configuration $\underline{1}$ are both in R. Hence the delta-matroid property fails for R. $\square$

We now give a reduction from an unbounded-degree #CSP, which will be used to give reductions from $\#\mathsf{BIS}$ and $\#\mathsf{SAT}$.

Lemma 5.31. *Let Γ be a finite unweighted constraint language, not consisting entirely of delta matroids. Then*

$$\#\mathsf{CSP}^{\geq 0}(\Gamma) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma)$$

Proof. Let $R \in \Gamma$ be a relation that is not a delta-matroid. We will first argue that there is a configuration $\mathbf{z} \in \{0, 1\}^3$ and a (≤ 2)-formula φ over $\{R, \mathrm{PIN}_0, \mathrm{PIN}_1\}$ such that $[\![\varphi]\!]$ is an arity 3 weight-function F satisfying:

$$\begin{aligned} &F(\mathbf{z} \oplus (0, 0, 0)) = F(\mathbf{z} \oplus (1, 1, 1)) = 1, \text{ and} \\ &F(\mathbf{z} \oplus (1, 0, 0)) = F(\mathbf{z} \oplus (1, 1, 0)) = F(\mathbf{z} \oplus (1, 0, 1)) = 0. \end{aligned} \tag{5.1}$$

By Lemma 5.27, R is not terraced, which means there exists a partial configuration $\mathbf{p}$ and $i, j \in \operatorname{dom} \mathbf{p}$ such that $R(\mathbf{p}, \cdot)$ is identically zero but $R(\mathbf{p} \oplus \{i\}, \cdot)$ and

$R(\mathbf{p} \oplus \{j\}, \cdot)$ are not parallel. Fixing $\{i, j\}$, take $\mathbf{p}$ to be maximal such that $R(\mathbf{p} \oplus \{i\}, \cdot)$ and $R(\mathbf{p} \oplus \{j\}, \cdot)$ are not parallel. By Lemma 5.28, $R(\mathbf{p} \oplus \{i\}, \cdot)$ and $R(\mathbf{p} \oplus \{j\}, \cdot)$ are distinct non-empty subsets of $\{\mathbf{y}, \overline{\mathbf{y}}\}$ for some $\mathbf{y}$. Without loss of generality (swapping i and j if necessary, and swapping $\mathbf{y}$ and $\overline{\mathbf{y}}$ if necessary), $R(\mathbf{p} \oplus \{j\}, \cdot) = \{\mathbf{y}\}$.

Pick some index k not in $\operatorname{dom} \mathbf{p}$. Let φ be the (≤ 2)-formula on variable set $v_1, \dots, v_n$ where n is the arity of R, with external variables (v_i, v_j, v_k), with a constraint $\langle (v_1, \dots, v_n), R\rangle$ and constraints $\langle (v_t), \mathrm{PIN}_{p_t}\rangle$ for each $t \in \operatorname{dom} \mathbf{p} \setminus \{i, j\}$. Let $F = [\![\varphi]\!]$, so for all $\mathbf{x} \in \{0,1\}^{\{i,j,k\}}$, the quantity $F(x_i, x_j, x_k)$ is the number of extensions of $\mathbf{x}$ to a configuration of R agreeing with $\mathbf{p}$ on $\operatorname{dom} \mathbf{p} \setminus \{i, j\}$. Set $\mathbf{z} = (1-p_i, p_j, 1-y_k)$. Since $R(\mathbf{p}, \cdot)$ is identically zero, we have $F(\mathbf{z} \oplus (1,0,0)) = F(\mathbf{z} \oplus (1,0,1)) = 0$. Since $R(\mathbf{p} \oplus \{j\}, \cdot) = \{\mathbf{y}\}$ we have $F(\mathbf{z} \oplus (1,1,1)) = 1$ and $F(\mathbf{z} \oplus (1,1,0)) = 0$. Since $\{\overline{\mathbf{y}}\} \subseteq R(\mathbf{p} \oplus \{i\}, \cdot) \subseteq \{\mathbf{y}, \overline{\mathbf{y}}\}$ we have $F(\mathbf{z} \oplus (0,0,0)) = 1$.

So F satisfies (5.1). By pinning (Lemma 5.25) and substitution (Lemma 5.21) we have $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma \cup \{F\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma)$. Let $h(1) = 2 - 4z_1$ and $h(2) = 2z_2 - 1$ and $h(3) = 2z_3 - 1$. Then $S = F_{h-\max}$ is the relation $\{\mathbf{z}, \overline{\mathbf{z}}\}$. By h-maximisation (Lemma 5.23) we have $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma \cup \{S\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma \cup \{F\})$. Taking a copy of S if necessary, we have $S = \mathrm{EQ}_3 = \{(0,0,0),(1,1,1)\}$ or $S = \{(0,1,1),(1,0,0)\}$. In the second case note that $\mathrm{EQ}_3(x,y,z) = \sum_{Y,Z=0}^{1} S(x,Y,Z) S(Y,y,z)$ for all $x, y, z \in \{0,1\}$, so by substitution (Lemma 5.21), $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma \cup \{\mathrm{EQ}_3\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma \cup \{S\})$. By Lemma 5.24(iv.), $\#\mathsf{CSP}^{\geq 0}(\Gamma) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\Gamma \cup \{\mathrm{EQ}_3\})$. □

Lemma 5.32. *Let R be a delta-matroid that is not basically binary. Then $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(R)$.*

Proof. Define the temporary notation $\mathrm{PM}^{\mathbf{x}} = \{(1-x_1, x_2, x_3), (x_1, 1-x_2, x_3), (x_1, x_2, 1-x_3)\}$ for all $\mathbf{x} \in \{0,1\}^3$. We will first argue that $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathrm{PM}^{\mathbf{x}})$ for all $\mathbf{x} \in \{0,1\}^3$.

For $x_1 + x_2 + x_3 \geq 2$, note that reversing the roles of zero and one gives a AP-reduction, $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathrm{PM}^{\overline{\mathbf{x}}}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathrm{PM}^{\mathbf{x}})$. So we only need to consider the case $x_1 + x_2 + x_3 \leq 1$. Lemma 4.15 implies $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathrm{PM}^{(0,0,0)})$. For $\mathbf{x} = (1,0,0)$ note

$$\mathrm{PM}^{(0,0,0)}(x,y,z) = \sum_{t,x'=0}^{1} \mathrm{PIN}_1(t)\mathrm{PM}^{(1,0,0)}(t,x,x')\mathrm{PM}^{(1,0,0)}(x',y,z)$$

By Lemma 4.15 and substitution (Lemma 5.21),

$$\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathrm{PM}^{(0,0,0)}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathrm{PM}^{(1,0,0)}, \mathrm{PIN}_1). \tag{5.2}$$

By pinning (Lemma 5.25) we get $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathrm{PM}^{(1,0,0)})$. By permuting variables, the same reduction applies whenever $x_1 + x_2 + x_3 = 1$. Thus $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathrm{PM}^{\mathbf{x}})$ for all $\mathbf{x} \in \{0,1\}^3$.

By Lemma 5.29 there is a copy R' of a pinning of R and some $\mathbf{x} \in \{0,1\}^3$ such that $\mathbf{x} \oplus U \notin R$ for subsets U of $\{1,2,3\}$ with $|U| < d$ and $\mathbf{x} \oplus U \in R$ for $|U| = d$. Let $h(1) = 2x_1 - 1$ and $h(2) = 2x_2 - 1$ and $h(3) = 2x_3 - 1$. Then $R'_{h-\max}$ is $\mathrm{PM}^{\mathbf{x}}$ (if $d = 1$) or $\mathrm{PM}^{\overline{\mathbf{x}}}$ (if

$d = 2$). We have shown that $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}_{\leq 2}^{\geq 0}(R'_{h-\max})$, and by pinning (Lemma 5.25) and h-maximisation (Lemma 5.23) we have $\#\mathsf{CSP}_{\leq 2}^{\geq 0}(R'_{h-\max}) \leq_{\mathrm{AP}} \#\mathsf{CSP}_{\leq 2}^{\geq 0}(R)$. $\square$

Theorem 5.3. Let Γ be a finite unweighted constraint language. If every relation in Γ is basically binary, or if $\Gamma \subseteq \mathsf{NEQconj}$, then $\#\mathsf{CSP}_{\leq 2}^{\geq 0}(\Gamma)$ is in FP. Otherwise:

- If every relation in Γ is a delta-matroid then $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}_{\leq 2}^{\geq 0}(\Gamma)$.
- If some relation in Γ is not a delta-matroid and $\Gamma \subseteq \mathsf{IMconj}$, then $\#\mathsf{BIS} =_{\mathrm{AP}} \#\mathsf{CSP}_{\leq 2}^{\geq 0}(\Gamma)$.
- If some relation in Γ is not a delta-matroid and $\Gamma \not\subseteq \mathsf{IMconj}$ then $\#\mathsf{SAT} =_{\mathrm{AP}} \#\mathsf{CSP}_{\leq 2}^{\geq 0}(\Gamma)$.

Proof. The inclusion in FP is given by Lemma 5.24(i.). We will therefore assume that Γ contains a relation that is not in NEQ-conj and a relation that is not basically binary. Consider the four cases depending on whether $\Gamma \subseteq \mathsf{IMconj}$ and whether Γ consists entirely of delta-matroids:

IMconj	delta matroids	
yes	yes	impossible by Lemma 5.30
no	yes	$\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}_{\leq 2}^{\geq 0}(\Gamma)$ by Lemma 5.32
yes	no	$\#\mathsf{BIS} =_{\mathrm{AP}} \#\mathsf{CSP}_{\leq 2}^{\geq 0}(\Gamma)$ by Lemma 5.31, Proposition 5.8 and Lemma 5.24(ii.)
no	no	$\#\mathsf{SAT} =_{\mathrm{AP}} \#\mathsf{CSP}_{\leq 2}^{\geq 0}(\Gamma)$ by Lemma 5.31, Proposition 5.8 and Lemma 5.24(iii.)

$\square$

5.4 Extensions of the main theorem

In this section we will establish the extensions of Theorem 5.3 mentioned in the introduction.

5.4.1 Simulating an unbounded degree problem

To extend Lemma 5.31 from non-delta-matroids to non-terraced weight-functions, rather than reducing $\#\mathsf{CSP}(\mathcal{F})$ to $\#\mathsf{CSP}_{=2}(\mathcal{F})$, we will reduce from a #CSP using functions of the form $T^{\otimes k}F$ and $T^{\otimes k}(FG)$.

Definition 5.33. For any weighted constraint language $\mathcal{F}$ define $T\mathcal{F}^1$ and $T\mathcal{F}^2$ by

$$T\mathcal{F}^1 = \{T^{\otimes k}F \mid F \text{ is an arity } k \text{ function in } \mathcal{F} \text{ for some } k \geq 0\}$$
$$T\mathcal{F}^2 = \{T^{\otimes k}(FG) \mid F \text{ and } G \text{ are arity } k \text{ functions in } \mathcal{F} \text{ for some } k \geq 0\}.$$

Lemma 5.34. *Let $\mathcal{F}$ be a finite weighted constraint language. Let $T \in \mathbb{Q}^{2\times 2}$.*

1. *If $\mathcal{F}$ contains some $G\colon \{0,1\}^3 \to \mathbb{Q}_{\geq 0}$ such that for all $x, y \in \{0,1\}$ we have $G(1,0,y) = 0$ and $G(x,x,y) = T_{x,y}$, then $\#\mathsf{CSP}(T\mathcal{F}^1) \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(\mathcal{F})$.*

2. If $\mathcal{F}$ contains some $G\colon \{0,1\}^4 \to \mathbb{Q}_{\geq 0}$ such that for all $x, y, y' \in \{0,1\}$ we have $G(1,0,y,y') = 0$ and $G(x,x,y,y') = \mathrm{EQ}_2(y,y')T_{x,y}$, then $\#\mathsf{CSP}(T\mathcal{F}^2) \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(\mathcal{F})$.

Proof. (1.) The reduction is given an instance (V,C) of $\#\mathsf{CSP}(T\mathcal{F}^1)$. We may assume that every variable has non-zero degree. It will be convenient to label the functions and variables used by constraints as $c = \langle (v_{c,1},\ldots,v_{c,k_c}), T^{\otimes k_c}F_c\rangle$ for $c \in C$. We wish to approximate

$$Z_{V,C} = \sum_{\mathbf{z}\in\{0,1\}^V} \prod_{c\in C} (T^{\otimes k_c}F_c)(\mathbf{z}(v_{c,1}),\ldots,\mathbf{z}(v_{c,k_c})).$$

We will enumerate each use of each variable in the following way. Define $L = \{(v,d) \mid v \in V, 1 \leq d \leq \deg_C(v)\}$ and $R = \{(c,j) \mid c \in C, 1 \leq j \leq k_c\}$. There is a bijection $g : L \to R$ where $g(v,i) = (c,j)$ means that that i'th occurrence of v, according to a fixed enumeration of C, is as the j'th variable in the constraint c. In other words, for all $c \in C$, for all $1 \leq i \leq k_c$ there exists $1 \leq j \leq \deg_C(v_{c,i})$ such that $g(v_{c,i},j) = (c,i)$.

Define (V',C') to be the instance of $\#\mathsf{CSP}_{=2}(\mathcal{F})$ where:

- the variable set V' is the disjoint union of L and R,
- there is one constraint $\langle((v,d-1),(v,d),g(v,d)),G\rangle$ for each pair $(v,d) \in L$, where $(v,0)$ means $(v,\deg_C(v))$, and also
- there is one constraint $\langle((c,1),\ldots,(c,k_c)),F_c\rangle$ for each $c \in C$.

Thus

$$Z_{V',C'} = \sum_{\mathbf{x},\mathbf{y}} \left(\prod_{v,d} G(\mathbf{x}(v,d-1),\mathbf{x}(v,d),\mathbf{y}(g(v,d)))\right)\left(\prod_{c} F_c(\mathbf{y}(c,1),\ldots,\mathbf{y}(c,k_c))\right).$$

Here, and for the rest of the proof of case (1.), indices c,j,v,d range over $c \in C$ and $1 \leq j \leq k_c$ and $v \in V$ and $1 \leq d \leq \deg_C(v)$. The variables $\mathbf{x},\mathbf{y},\mathbf{z}$ range over $\mathbf{x}\colon L \to \{0,1\}$ and $\mathbf{y}\colon R \to \{0,1\}$ and $\mathbf{z}\colon V \to \{0,1\}$. And $\mathbf{x}(v,0)$ means $\mathbf{x}(v,\deg_C(v))$.

The reduction queries the $\#\mathsf{CSP}_{=2}(\mathcal{F})$ oracle on (V',C'), passing through the error parameter, and returns the result. To show that the reduction is correct we will show that $Z_{V,C} = Z_{V',C'}$. Define

$$\begin{aligned}
\mathrm{ZTerms}(\mathbf{z}) &= \prod_c (T^{\otimes k_c} F_c)(\mathbf{z}(v_{c,1}), \ldots, \mathbf{z}(v_{c,k_c})) \\
\mathrm{YZTrans}(\mathbf{y}, \mathbf{z}) &= \prod_{c,j} T_{\mathbf{z}(v_{c,j}),\mathbf{y}(c,j)} \\
\mathrm{YTerms}(\mathbf{y}) &= \prod_c F_c(\mathbf{y}(c,1), \ldots, \mathbf{y}(c,k_c)) \\
\mathrm{XEq}(\mathbf{x}) &= \prod_v \mathrm{EQ}_{\deg_C(v)}(\mathbf{x}(v,1), \cdots, \mathbf{x}(v, \deg_C(v))) \\
\mathrm{XYTrans}(\mathbf{x}, \mathbf{y}) &= \prod_{v,d} T_{\mathbf{x}(v,d),\mathbf{y}(g(v,d))} \\
\mathrm{XYGTrans}(\mathbf{x}, \mathbf{y}) &= \prod_{v,d} G(\mathbf{x}(v, d-1), \mathbf{x}(v,d), \mathbf{y}(g(v,d)))
\end{aligned}$$

Note:

- For fixed $\mathbf{z}$ we have $\mathrm{ZTerms}(\mathbf{z}) = \sum_{\mathbf{y}} \mathrm{YZTrans}(\mathbf{y}, \mathbf{z})\mathrm{YTerms}(\mathbf{y})$ by expanding the definition of $T^{\otimes k_c} F_c$.
- Summing over $\mathbf{x}$ with the factor $\mathrm{XEq}(\mathbf{x})$ is the same as summing over $\mathbf{z}$ and defining $\mathbf{x}$ by $\mathbf{x}(v,d) = \mathbf{z}(v)$. Hence summing over $\mathbf{x}$ with the factor $\mathrm{XEq}(\mathbf{x})\mathrm{XYTrans}(\mathbf{x}, \mathbf{y})$ is the same as summing over $\mathbf{z}$ with the factor $\mathrm{YZTrans}(\mathbf{y}, \mathbf{z})$.
- Fix $\mathbf{x}$ and $\mathbf{y}$. If $\mathrm{XEq}(\mathbf{x}) = 1$ then $\mathrm{XYTrans}(\mathbf{x}, \mathbf{y}) = \mathrm{XYGTrans}(\mathbf{x}, \mathbf{y})$ by definition of G. And if $\mathrm{XEq}(\mathbf{x})$ is zero then so is $\mathrm{XYGTrans}(\mathbf{x}, \mathbf{y})$, which implies $\mathrm{XEq}(\mathbf{x})\mathrm{XYTrans}(\mathbf{x}, \mathbf{y}) = \mathrm{XYGTrans}(\mathbf{x}, \mathbf{y}) = 0$.

Hence

$$\begin{aligned}
Z_{V,C} &= \sum_{\mathbf{z}} \mathrm{ZTerms}(\mathbf{z}) \\
&= \sum_{\mathbf{y},\mathbf{z}} \mathrm{YZTrans}(\mathbf{y}, \mathbf{z})\mathrm{YTerms}(\mathbf{y}) \\
&= \sum_{\mathbf{x},\mathbf{y}} \mathrm{XEq}(\mathbf{x})\mathrm{XYTrans}(\mathbf{x}, \mathbf{y})\mathrm{YTerms}(\mathbf{y}) \\
&= \sum_{\mathbf{x},\mathbf{y}} \mathrm{XYGTrans}(\mathbf{x}, \mathbf{y})\mathrm{YTerms}(\mathbf{y}) \\
&= Z_{V',C'}
\end{aligned}$$

(2.) The reduction is given an instance (V, C) of $\#\mathsf{CSP}(T\mathcal{F}^2)$. We may assume that every variable has non-zero degree. It will be convenient to label the functions and variables used by constraints as $c = \langle (v_{c,1}, \ldots, v_{c,k_c}), T^{\otimes k_c}(F_{c,1}F_{c,2})\rangle$ for $c \in C$. We wish to approximate

$$Z_{V,C} = \sum_{\mathbf{z} \in \{0,1\}^V} \prod_{c \in C} (T^{\otimes k_c}(F_{c,1}F_{c,2}))(\mathbf{z}(v_{c,1}), \dots, \mathbf{z}(v_{c,k_c})).$$

Let L, R, g be defined as before. Define (V', C') to be the instance of $\#\mathsf{CSP}_{=2}(\mathcal{F})$ where:

- the variable set is the disjoint union of L and $R \times \{1,2\}$,
- there is one constraint $\langle ((v, d-1), (v, d), (g(v,d), 1), (g(v,d), 2)), G \rangle$ for each pair $(v, d) \in L$, where $(v, 0)$ means $(v, \deg_C(v))$, and also
- there is one constraint $\langle (((c,1), b), \dots, ((c, k_c), b)), F_{c,b} \rangle$ for each $c \in C$ and each $b \in \{1, 2\}$.

Thus

$$Z_{V',C'} = \sum_{\mathbf{x},\mathbf{y}} \left(\prod_{v,d} G(\mathbf{x}(v, d-1), \mathbf{x}(v,d), \mathbf{y}(g(v,d),1), \mathbf{y}(g(v,d),2)) \right) \cdot \left(\prod_{c,b} F_{c,b}(\mathbf{y}((c,1),b), \dots, \mathbf{y}((c,k_c),b)) \right).$$

Here and for the rest of the proof, indices c, j, b, v, d range over $c \in C$ and $1 \le j \le k_c$ and $b \in \{1,2\}$ and $v \in V$ and $1 \le d \le \deg_C(v)$. The variables $\mathbf{x}, \mathbf{y}, \mathbf{z}$ range over $\mathbf{x} \colon L \to \{0,1\}$ and $\mathbf{y} \colon R \times \{1,2\} \to \{0,1\}$ and $\mathbf{z} \colon V \to \{0,1\}$. As before, $\mathbf{x}(v, 0)$ means $\mathbf{x}(v, \deg_C(v))$.

The reduction queries the $\#\mathsf{CSP}_{=2}(\mathcal{F})$ oracle on (V', C'), passing through the error parameter, and returns the result. To show that the reduction is correct we will show that $Z_{V,C} = Z_{V',C'}$. Define

$$\begin{aligned}
\mathrm{ZTerms}(\mathbf{z}) &= \prod_{c} (T^{\otimes k_c}(F_{c,1}F_{c,2}))(\mathbf{z}(v_{c,1}), \dots, \mathbf{z}(v_{c,k_c})) \\
\mathrm{YZTrans}(\mathbf{y}, \mathbf{z}) &= \prod_{c,j} \mathrm{EQ}_2(\mathbf{y}((c,j),1), \mathbf{y}((c,j),2)) T_{\mathbf{z}(v_{c,j}), \mathbf{y}((c,j),1)} \\
\mathrm{YTerms}(\mathbf{y}) &= \prod_{c,b} F_{c,b}(\mathbf{y}((c,1),b), \dots, \mathbf{y}((c,k_c),b)) \\
\mathrm{XEq}(\mathbf{x}) &= \prod_{v} \mathrm{EQ}_{\deg_C(v)}(\mathbf{x}(v,1), \dots, \mathbf{x}(v, \deg_C(v))) \qquad \text{(as in case (1.))} \\
\mathrm{XYTrans}(\mathbf{x}, \mathbf{y}) &= \prod_{v,d} \mathrm{EQ}_2(\mathbf{y}(g(v,d),1), \mathbf{y}(g(v,d),2)) T_{\mathbf{x}(v,d), \mathbf{y}(g(v,d),1)} \\
\mathrm{XYGTrans}(\mathbf{x}, \mathbf{y}) &= \prod_{v,d} G(\mathbf{x}(v, d-1), \mathbf{x}(v,d), \mathbf{y}(g(v,d),1), \mathbf{y}(g(v,d),2))
\end{aligned}$$

As in case (1.) we have $\mathrm{ZTerms}(\mathbf{z}) = \sum_{\mathbf{y}} \mathrm{YZTrans}(\mathbf{y},\mathbf{z})\mathrm{YTerms}(\mathbf{y})$ by expanding the definition of $T^{\otimes k_c}(F_{c,1}F_{c,2})$. The rest of the argument is identical to case (1.) and we get $Z_{V,C} = Z_{V',C'}$. □

5.4.2 Pinnings

As in the main theorem, we will use pinnings. The following lemma gives the necessary analogue of Lemma 5.25.

Lemma 5.35. *Let $\mathcal{F}$ be a finite weighted constraint language. Let F' be a copy of a pinning of a weight-function $F \in \mathcal{F}$. Let $(1,0),(0,1) \in W \subseteq \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$. Then $\#\mathsf{CSP}^W_{=2}(\mathcal{F} \cup \{F', \mathrm{PIN}_0, \mathrm{PIN}_1\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^W_{=2}(\mathcal{F})$.*

Proof. As in Lemma 5.25, it suffices to show that $\#\mathsf{CSP}^W_{=2}(\mathcal{F} \cup \{\mathrm{PIN}_0, \mathrm{PIN}_1\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^W_{=2}(\mathcal{F})$. The reduction is given an instance (V, C, w) of $\#\mathsf{CSP}^W_{=2}(\mathcal{F} \cup \{\mathrm{PIN}_0, \mathrm{PIN}_1\})$ and an error parameter ε. If $\langle (v), \mathrm{PIN}_0\rangle, \langle (v), \mathrm{PIN}_1\rangle \in C$ for some $v \in V$ then the reduction can correctly output $Z_{V,C,w} = 0$, so we will assume this does not occur. We will define an instance (V', C', w') of $\#\mathsf{CSP}^W_{=2}(\mathcal{F})$ such that $Z_{V,C,w}$ can be computed easily from $Z_{V',C',w'}$.

Let $C_{\mathcal{F}}$ be the list of constraints in C that do not use PIN_0 or PIN_1. For each $d \in \{0,1,2\}$ let $V_d = \{v \mid \deg_{C_{\mathcal{F}}}(v) = d\}$. Let V' be the disjoint union of $V_2 \times \{0,1\}$ and V_1. For each $i \in \{0,1\}$ define $s_i \colon V_1 \cup V_2 \to V'$ by

$$s_i(v) = \begin{cases} v & \text{if } v \in V_1, \text{ and} \\ (v,i) & \text{otherwise.} \end{cases}$$

Let C' be the list with the two constraints $\langle (s_0(v_1), \dots, s_0(v_k)), F\rangle$ and $\langle (s_1(v_1), \dots, s_1(v_k)), F\rangle$ for each constraint $\langle (v_1, \dots, v_k), F\rangle \in C$. Define $w' \colon V' \to W$ by

$$\begin{aligned} w'(v) &= (1,0) && \text{for } v \in V_1 \text{ where } \langle (v), \mathrm{PIN}_0\rangle \in C \\ w'(v) &= (0,1) && \text{for } v \in V_1 \text{ where } \langle (v), \mathrm{PIN}_1\rangle \in C \\ w'(v,i) &= w(v) && \text{for all } v \in V_2 \text{ and } i \in \{0,1\}. \end{aligned}$$

For each $v \in V_0 \cup V_1$ there is a unique $p(v) \in \{0,1\}$ such that $\langle (v), \mathrm{PIN}_{p(v)}\rangle \in C$. Define $K_0 = \prod_{v \in V_0} w(v)_{p(v)}$ and $K_1 = \prod_{v \in V_1} w(v)_{p(v)}$. Then

$$Z^2_{V,C,w} = K_0^2 \sum_{\mathbf{x} \in \{0,1\}^{V'}} \mathrm{wt}_{V_1 \cup V_2, C_{\mathcal{F}}, w|_{V_1 \cup V_2}}(\mathbf{x} \circ s_0)\, \mathrm{wt}_{V_1 \cup V_2, C_{\mathcal{F}}, w|_{V_1 \cup V_2}}(\mathbf{x} \circ s_1) = K_0^2 K_1^2 Z_{V',C',w'}.$$

The reduction should call the oracle on (V', C', w') with error parameter ε to obtain a value Z' satisfying $e^{-\varepsilon/2}\sqrt{Z_{V',C',w'}} \leq \sqrt{Z'} \leq e^{\varepsilon/2}\sqrt{Z_{V',C',w'}}$ (with probability at least 3/4) then compute a rational Z'' such that $e^{-\varepsilon/2}\sqrt{Z'} \leq Z'' \leq e^{\varepsilon/2}\sqrt{Z'}$, then return $K_0 K_1 Z''$. □

Since the constraint language is no longer restricted to relations, to show #BIS- and #SAT-hardness we will require some observations about weight-functions that are pinning-minimal subject to various conditions.

Lemma 5.36. *Let F be a pinning-minimal weight-function subject to not being log-supermodular. Then* $\operatorname{supp}(F) \subseteq \{\underline{0}, \mathbf{x}, \overline{\mathbf{x}}, \underline{1}\}$ *for some* $\mathbf{x}$*.*

Proof. For all $\mathbf{x}, \mathbf{y}$ such that $F(\mathbf{x} \wedge \mathbf{y})F(\mathbf{x} \vee \mathbf{y}) < F(\mathbf{x})F(\mathbf{y})$, the pinning of F by $\{i \mapsto x_i \mid x_i = y_i\}$ is not log-supermodular so $\mathbf{y} = \overline{\mathbf{x}}$. Taking the contrapositive, for all $\mathbf{y}, \mathbf{z}$ such that $\mathbf{z} \neq \overline{\mathbf{y}}$ we have $F(\mathbf{z} \wedge \mathbf{y})F(\mathbf{z} \vee \mathbf{y}) \geq F(\mathbf{z})F(\mathbf{y})$. And there exists $\mathbf{x}$ with $F(\underline{0})F(\underline{1}) < F(\mathbf{x})F(\overline{\mathbf{x}})$. Let $\mathbf{z} \notin \{\underline{0}, \underline{1}, \mathbf{x}, \overline{\mathbf{x}}\}$. Then

$$\begin{aligned}
F(\mathbf{x} \wedge \mathbf{z})F(\mathbf{x} \vee \mathbf{z}) &\geq F(\mathbf{x})F(\mathbf{z}) \\
F(\overline{\mathbf{x}} \wedge \mathbf{z})F(\overline{\mathbf{x}} \vee \mathbf{z}) &\geq F(\overline{\mathbf{x}})F(\mathbf{z}) \\
F(\underline{0})F(\mathbf{z}) &\geq F(\mathbf{x} \wedge \mathbf{z})F(\overline{\mathbf{x}} \wedge \mathbf{z}) \\
F(\mathbf{z})F(\underline{1}) &\geq F(\mathbf{x} \vee \mathbf{z})F(\overline{\mathbf{x}} \vee \mathbf{z})
\end{aligned}$$

In each case we have used the fact that the configurations on the right-hand-side are not complements, or, equivalently, the configurations on the left-hand-side are not $\underline{0}$ and $\underline{1}$.

Multiplying these four inequalities we get $F(\underline{0})F(\underline{1})C \geq F(\mathbf{x})F(\overline{\mathbf{x}})C$ where

$$C = F(\mathbf{z})^2 F(\mathbf{x} \wedge \mathbf{z})F(\mathbf{x} \vee \mathbf{z})F(\overline{\mathbf{x}} \wedge \mathbf{z})F(\overline{\mathbf{x}} \vee \mathbf{z}) \geq F(\mathbf{x})F(\overline{\mathbf{x}})F(\mathbf{z})^4.$$

But $F(\underline{0})F(\underline{1}) < F(\mathbf{x})F(\overline{\mathbf{x}})$ so $C = 0$ and hence $F(\mathbf{z}) = 0$. □

Lemma 5.37. *Let R be a pinning-minimal relation subject to not being closed under joins (so there exists $\mathbf{x}, \mathbf{y} \in R$ such that $\mathbf{x} \vee \mathbf{y} \notin R$). Then $R = \{\underline{0}, \mathbf{x}, \overline{\mathbf{x}}\}$ or $R = \{\mathbf{x}, \overline{\mathbf{x}}\}$.*

Proof. For all $\mathbf{x}, \mathbf{y} \in R$ with $\mathbf{x} \vee \mathbf{y} \notin R$, the pinning of R by $\{i \mapsto x_i \mid x_i = y_i\}$ is not closed under joins so $\mathbf{y} = \overline{\mathbf{x}}$. Hence there exists $\mathbf{x}$ with $\mathbf{x}, \overline{\mathbf{x}} \in R$, and $\underline{1} \notin R$. Also, taking contrapositives, if $\mathbf{y}, \mathbf{z} \in R$ and $\mathbf{y} \neq \overline{\mathbf{z}}$ then $\mathbf{y} \vee \mathbf{z} \in R$.

Consider $\mathbf{y} \in R \setminus \{\mathbf{x}, \overline{\mathbf{x}}\}$. By the previous paragraph, $\mathbf{x} \vee \mathbf{y} \in R$ and $\overline{\mathbf{x}} \vee \mathbf{y} \in R$. But $(\mathbf{x} \vee \mathbf{y}) \vee (\overline{\mathbf{x}} \vee \mathbf{y}) = \underline{1} \notin R$, so $\mathbf{x} \vee \mathbf{y}$ is the complement of $\overline{\mathbf{x}} \vee \mathbf{y}$. Hence $\max(x_i, y_i) = 1 - \max(1 - x_i, y_i) = \min(x_i, 1 - y_i)$ for all variables i, which implies $\mathbf{y} = \underline{0}$. □

Recall that F is *IM-terraced* if for all partial configurations $\mathbf{p}$ such that the pinning $F(\mathbf{p}, \cdot)$ is identically zero, for all $i, j \in \operatorname{dom} \mathbf{p}$ such that $p_i \neq p_j$, the pinnings $F(\mathbf{p} \oplus \{i\}, \cdot)$ and $F(\mathbf{p} \oplus \{j\}, \cdot)$ are parallel.

Lemma 5.38. *For every weight-function G that is pinning-minimal subject to not being IM-terraced, there is a copy $F \colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ of G such that:*

- *for all $y_3, \ldots, y_k \in \{0,1\}$ we have $F(1, 0, y_3, \ldots, y_k) = 0$, and*

- *there exists a configuration* $\mathbf{z}$ *of* $\{3, 4, \cdots, k\}$ *and a non-singular matrix* $T \in \mathbb{Q}^{2\times 2}$ *such that for all* $x, y_3, \cdots, y_k \in \{0, 1\}$ *we have*

$$F(x, x, y_3, \cdots, y_k) = \begin{cases} T_{x,y_3} & \text{if } \mathbf{y} = \mathbf{z} \text{ or } \mathbf{y} = \overline{\mathbf{z}} \\ 0 & \text{otherwise.} \end{cases}$$

Proof. Consider an arbitrary weight-function $G\colon \{0,1\}^V \to \mathbb{Q}_{\geq 0}$ that is pinning-minimal subject to not being IM-terraced. Since G is not IM-terraced there exist $\mathbf{p}, i, j$ such that $p_i = 1$ and $p_j = 0$ and $G(\mathbf{p}, \cdot)$ is identically zero but $G(\mathbf{p} \oplus \{i\}, \cdot)$ and $G(\mathbf{p} \oplus \{j\}, \cdot)$ are not parallel. Let $\mathbf{p}'$ be the restriction of $\mathbf{p}$ to $\operatorname{dom} \mathbf{p} \setminus \{1, 2\}$. Then $F' = F(\mathbf{p}', \cdot)$ is also not IM-terraced: $F'((\mathbf{p} \oplus \{i\})|_{\{i,j\}}, \cdot) = F(\mathbf{p} \oplus \{i\}, \cdot)$ and $F'((\mathbf{p} \oplus \{j\})|_{\{i,j\}}, \cdot) = F(\mathbf{p} \oplus \{j\}, \cdot)$ are not parallel. Hence $\operatorname{dom} \mathbf{p} = \{i, j\}$ by minimality of F.

Pick a bijection π from V to $\{1, \ldots, k\}$ sending i to 1 and j to 2. Then the copy F of G defined by $F(\mathbf{x}) = G(\mathbf{x} \circ \pi)$ has the same properties with $i = 1$ and $j = 2$; letting 00, 10 and 11 denote $(0,0)$ and $(1,0)$ and $(1,1)$ considered as partial configurations living in $\{0,1\}^{\{1,2\}}$, the pinning $F(10, \cdot)$ is identically zero but $F(00, \cdot)$ and $F(11, \cdot)$ are not parallel.

We will argue that $(F(00, \cdot), F(11, \cdot))$ is pinning-minimal subject to not being parallel. We need to check that for any non-empty partial configuration $\mathbf{y}$ of $\{3, \cdots, k\}$ the pinnings $F((00, \mathbf{y}), \cdot)$ and $F((11, \mathbf{y}), \cdot)$ are parallel. $F((10, \mathbf{y}), \cdot)$ is identically zero, and $10 \oplus \{1\} = 00$ and $10 \oplus \{2\} = 11$, and $F(\mathbf{y}, \cdot)$ is IM-terraced by minimality of F, so $F((00, \mathbf{y}), \cdot)$ and $F((11, \mathbf{y}), \cdot)$ are parallel.

By Lemma 5.28 there exists $\mathbf{z} \in \{0,1\}^{\{3,4,\cdots,k\}}$ such that $\operatorname{supp}(F(00, \cdot)) \cup \operatorname{supp}(F(11, \cdot)) = \{\mathbf{z}, \overline{\mathbf{z}}\}$. Without loss of generality we may take $z_3 = 0$. Set $T_{00} = F(00, \mathbf{z})$ and $T_{01} = F(00, \overline{\mathbf{z}})$ and $T_{10} = F(11, \mathbf{z})$ and $T_{11} = F(11, \overline{\mathbf{z}})$. This T satisfies the required expression for F. Furthermore the weight-functions F_{00} and F_{11} are not parallel, hence neither are the vectors (T_{00}, T_{01}) and (T_{10}, T_{11}), and hence T is non-singular. $\square$

5.4.3 #BIS- and #SAT-hardness

We will need a few constructions to help reduce from a suitable unbounded-degree #CSP. We work in the setting of finite sets W as much as possible, to help the proof of Theorem 5.13.

To use Proposition 5.8 we need to insert arity 1 weight-functions into the constraint language of a #CSP. Lemma 5.39 provides these, and Lemma 5.40 says that we can reduce one of the #CSPs considered in Proposition 5.8 to a read-twice #CSP with a non-IM-terraced weight-function. Lemma 5.42 applies this to reducing #BIS and #SAT to certain #$\mathsf{CSP}^W_{=2}$ problems.

Lemma 5.39. *Let* $B = 1$ *or* $B = 2$ *and let* $T \in \mathbb{Q}^{2\times 2}_{\geq 0}$ *be non-singular. Assume that either:* $T_{00} > T_{10}$ *and* $T_{01} < T_{11}$*, or* $T_{00} < T_{10}$ *and* $T_{01} > T_{11}$*. Let* $U(0), U(1)$ *be positive rationals and let* F *be any weight-function with* $|\operatorname{supp}(F)| > 1$*. Then* $U = [\![\varphi]\!]$ *for some* $\mathbb{N}$*-formula* φ *over* $T\mathcal{F}^B$ *where* $\mathcal{F}$ *is a finite set of simple weightings of* F*.*

Proof. First we will write U in the form $U(x) = U_1(x)\dots U_k(x)$ such that for each $1 \le i \le k$ the ratio $U_i(0)/U_i(1)$ lies in the closed interval from T_{00}/T_{10} to T_{01}/T_{11} where one of these endpoints may be $+\infty$ (so the interval is $[T_{00}/T_{10}, +\infty)$ when $T_{11} = 0$, and $[T_{01}/T_{11}, +\infty)$ when $T_{10} = 0$).

If $U(0) \le U(1)$, let $i \in \{0, 1\}$ satisfy $T_{0i} < T_{1i}$. Let $k \ge 1$ be the unique integer with $(T_{0i}/T_{1i})^k < U(0)/U(1) \le (T_{0i}/T_{1i})^{k-1}$. Then let $U_1(x) = \dots = U_{k-1}(x) = T_{xi}$ and $U_k(x) = U(x)/(T_{xi})^{k-1}$ for each $x \in \{0, 1\}$. Note that $T_{0i}/T_{1i} < U_k(0)/U_k(1) \le 1$.

If $U(0) > U(1)$, let $i \in \{0, 1\}$ satisfy $T_{0i} > T_{1i}$. Let $k \ge 1$ be the unique integer with $(T_{0i}/T_{1i})^{k-1} \le U(0)/U(1) < (T_{0i}/T_{1i})^k$. Then let $U_1(x) = \dots = U_{k-1}(x) = T_{xi}$ and $U_k(x) = U(x)/(T_{xi})^{k-1}$ for each $x \in \{0, 1\}$. Note that $1 \le U_k(0)/U_k(1) < T_{0i}/T_{1i}$.

We have constructed $U_1, \dots, U_k$. Now let $\mathbf{x}, \mathbf{x}'$ be distinct elements of $\mathrm{supp}(F)$. By permuting variables if necessary we can assume that $x_n \ne x'_n$ where n is the arity of F. Let

$$F'(y_n) = \sum_{x_1,\dots,x_{n-1},y_1,\dots,y_{n-1}=0}^{1} T_{x_1,y_1}\dots T_{x_{n-1},y_{n-1}} F(\mathbf{y})^B \qquad \text{for } y_n \in \{0, 1\}.$$

Since $x_n \ne x'_n$ we have $F'(0), F'(1) > 0$. Define $W_i(0), W_i(1)$ by the following equation.

$$\begin{pmatrix} F'(0)W_i(0) \\ F'(1)W_i(1) \end{pmatrix} = T^{-1} \begin{pmatrix} U_i(0) \\ U_i(1) \end{pmatrix} = \frac{1}{\det T} \begin{pmatrix} T_{11} & -T_{01} \\ -T_{10} & T_{00} \end{pmatrix} \begin{pmatrix} U_i(0) \\ U_i(1) \end{pmatrix}.$$

If $T_{00} > T_{10}$ then $\det T > 0$ and so $W_i(0), W_i(1) \ge 0$ because $T_{01}/T_{11} \le U_i(0)/U_i(1) \le T_{00}/T_{10}$ (where the final inequality should be ignored when $T_{10} = 0$). If $T_{00} > T_{10}$ then $\det T < 0$ and so $W_i(0), W_i(1) \ge 0$ because $T_{00}/T_{10} \le U_i(0)/U_i(1) \le T_{01}/T_{11}$ (where the final inequality should be ignored when $T_{11} = 0$).

For each $1 \le i \le k$ define F_i to be the simple weighting $F_i(x_1, \dots, x_n) = F(x_1, \dots, x_n)W_i(x_n)$. Then

$$U(x_n) = \prod_{i=1}^{k} U_i(x_n) = \begin{cases} \prod_{i=1}^{k} \sum_{x_1,\dots,x_{n-1}=0}^{1} (T^{\otimes n} F_i)(\mathbf{x}) & \text{if } B = 1 \\ \prod_{i=1}^{k} \sum_{x_1,\dots,x_{n-1}=0}^{1} (T^{\otimes n} (F F_i))(\mathbf{x}) & \text{if } B = 2. \end{cases}$$

This implicitly defines a $\mathbb{N}$-formula φ over $T\mathcal{F}^B$ with $U = [\![\varphi]\!]$, where $\mathcal{F} = \{F, F_1, \dots, F_k\}$. ∎

Lemma 5.40. *Let $\mathcal{F}$ be a finite weighted constraint language containing a non-IM-terraced weight-function. There exists $B \in \{1, 2\}$ and a non-singular matrix $T \in \mathbb{Q}_{\ge 0}^{2\times 2}$ such that for all finite sets of arity 1 weight-functions S there is a finite set $W \subseteq \mathbb{Q}_{\ge 0} \times \mathbb{Q}_{\ge 0}$ such that*

$$\#\mathsf{CSP}(T\mathcal{F}^B \cup S) \le_{\mathrm{AP}} \#\mathsf{CSP}^W_{=2}(\mathcal{F}).$$

Proof. We can always insert $(0,1)$ and $(1,0)$ into W, so by Lemma 5.35 we can assume that $\mathcal{F}$ is closed under taking pinnings.[2]

Choose a weight-function $F \in \mathcal{F}$ that is pinning-minimal subject to not being IM-terraced. Permuting variables if necessary, F has the form given by Lemma 5.38. So $F(1,0,y_3,\dots,y_k) = 0$ for all $y_3,\dots,y_k \in \{0,1\}$, and there exists $T \in \mathbb{Q}_{\geq 0}^{2\times 2}$ and $z_3,\dots,z_k \in \{0,1\}$ and $B \geq 1$ such that for all $x, y_3,\dots,y_k \in \{0,1\}$ we have

$$F(x,x,y_3,\dots,y_k) = \begin{cases} T_{x,y_3} & \text{if } \mathbf{y} \in \{\mathbf{z}, \overline{\mathbf{z}}\} \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

If $k \geq 5$ there are $3 \leq i < j \leq k$ with $y_i = y_j$. Define $F' \colon \{0,1\}^{k-2} \to \mathbb{Q}_{\geq 0}$ by

$$F'(x_1,x_2,y_3,\cdots,y_{i-1},y_{i+1},\cdots,y_{j-1},y_{j+1},\cdots,y_k) = \sum_{t=0}^{1} F(x_1,x_2,y_3,\cdots,y_{i-1},t,y_{i+1},\cdots,y_{j-1},t,y_{j+1},\cdots,y_k)$$

Let $\mathbf{y}'$ denote $\mathbf{y}$ with the i'th and j'th components deleted, and let $\mathbf{z}'$ denote $\mathbf{z}$ with the i'th and j'th components deleted. Equation (5.3) holds with $F, \mathbf{y}, \mathbf{z}$ replaced by $F', \mathbf{y}', \mathbf{z}'$, which implies that F' is not IM-terraced. F' is defined by a $(=2)$-formula over F, so by substitution (Lemma 5.21), $\#\mathsf{CSP}^{W}_{=2}(\mathcal{F} \cup \{F'\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{=2}(\mathcal{F})$ whenever $(1,1) \in W \subseteq \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$. Repeating this we can assume $k \leq 4$. Set $B = k-2$.

If $B = 2$ and $z_3 \neq z_4$, define F' by

$$F'(x_1,x_2,y_3,y_4) = \sum_{t,y_4'=0}^{1} F(x_1,x_2,y_3,y_4')F(t,t,y_4',y_4)$$

Then $F'(1,0,y_3,y_4) = 0$ for all $y_3,y_4 \in \{0,1\}$. Also, for all $x,y_3,y_4 \in \{0,1\}$,

$$F'(x,x,y_3,y_4) = \begin{cases} \sum_{t=0}^{1} F(t,t,1-y_3,y_3)T_{x,y_3} & \text{if } y_3 = y_4 \\ 0 & \text{otherwise.} \end{cases}$$

By substitution (Lemma 5.21) we can use F' instead of F. Therefore we can assume that $\mathbf{z}$ is either (0) or $(0,0)$.

Furthermore by taking a simple weighting of F and invoking Lemma 5.26, we can assume that there exist $i,j \in \{0,1\}$ such that $T_{i0} > T_{i1}$ and $T_{j0} < T_{j1}$. Indeed let $U(0) = T_{01} + T_{11}$ and $U(1) = T_{00} + T_{10}$. Replacing F by the simple weighting F' defined by

$$F'(x_1,x_2,y_3,y_4) = U(y_3)F(x_1,x_2,y_3,y_4)$$

[2]This is not literally true, since our convention is that constraint languages consist of standard weight-functions. The more precise statement is that we can assume that for every pinning F of a function in $\mathcal{F}$, some copy of F is in $\mathcal{F}$.

has the effect of replacing T_{xy} by $U(y)T_{xy}$. If $T_{00}T_{11} > T_{01}T_{10}$ then $U(0)T_{00} > U(1)T_{01}$ and $U(0)T_{10} < U(1)T_{11}$. Otherwise $T_{00}T_{11} < T_{10}T_{01}$ so $U(0)T_{00} < U(1)T_{01}$ and $U(0)T_{10} > U(1)T_{11}$.

Let S' be the set $\{U \in S \mid U(0), U(1) > 0\} \cup \{U_0, U_1\}$ where $U_0(0) = 2, U_0(1) = 1$ and $U_1(0) = 1, U_1(1) = 2$. By Lemma 5.39 there is a finite set $\mathcal{G}$ of simple weightings of F (note $|\operatorname{supp}(F)| > 1$) such that each $U \in S'$ can be expressed by a $\mathbb{N}$-formula over $\{T^{\otimes}\mathcal{G}^B\}$. By substitution (Lemma 5.21), Lemma 5.34, and simple weighting (Lemma 5.26), we have

$$\#\mathsf{CSP}(T\mathcal{F}^B \cup S') \leq_{\mathrm{AP}} \#\mathsf{CSP}(T\mathcal{F}^B \cup T\mathcal{G}^B) \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(\mathcal{F} \cup \mathcal{G}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{=2}(\mathcal{F})$$

for some finite set W. Using U_0 and U_1 as variable weights we have:

$$\#\mathsf{CSP}^{\{(2^a,2^b)|a,b\in\mathbb{Z}\}}(T\mathcal{F}^B \cup S') \leq_{\mathrm{AP}} \#\mathsf{CSP}(T\mathcal{F}^B \cup S')$$

But $\mathrm{PIN}_0 = (U_1)_{h-\max}$ with $h(1) = -1$, and similarly $\mathrm{PIN}_1 = (U_0)_{h-\max}$ with $h(1) = 1$, so by h-maximisation (Lemma 5.23) we have

$$\#\mathsf{CSP}(T\mathcal{F}^B \cup S' \cup \{\mathrm{PIN}_0, \mathrm{PIN}_1\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\{(2^a,2^b)|a,b\in\mathbb{Z}\}}(T\mathcal{F}^B \cup S')$$

The weight-functions in $S \setminus S'$ are just constant multiples of PIN_0 and PIN_1 so we have established that $\#\mathsf{CSP}(T\mathcal{F}^B \cup S) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{=2}(\mathcal{F})$. □

Lemma 5.41. *Let $F\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ be indecomposable. $F \in \mathsf{WNEQ}$ if and only if $|\operatorname{supp}(F)| \leq 2$.*

(In [27], this property is used to define the complex-valued analogue $\mathcal{E}$ of WNEQ. But we have defined WNEQ in a different way.)

Proof. If $F \in \mathsf{WNEQ}$ then F is a simple weighting of a $\mathsf{NEQconj}$ relation: there are numbers $\lambda, U_1(0), U_1(1), \ldots, U_k(0), U_k(1) \in \mathbb{Q}_{\geq 0}$ and sets $A, B \subseteq \{1, \ldots, k\} \times \{1, \ldots, k\}$ such that

$$\begin{aligned} F(x_1, \ldots, x_k) &= \lambda \left(\prod_{i=1}^{k} U_i(x_i) \right) \left(\prod_{(i,j)\in A} \mathrm{EQ}_2(x_i, x_j) \right) \left(\prod_{(i,j)\in B} \mathrm{NEQ}(x_i, x_j) \right) \\ &= \lambda \prod_{P} \left[\left(\prod_{i\in P} U_i(x_i) \right) \left(\prod_{(i,j)\in A\cap P^2} \mathrm{EQ}_2(x_i, x_j) \right) \left(\prod_{(i,j)\in B\cap P^2} \mathrm{NEQ}(x_i, x_j) \right) \right] \end{aligned}$$

where P runs over equivalence classes of the equivalence relation generated by $i \sim j$ if $(i, j) \in A \cup B$. This expresses F as a tensor product. But F is indecomposable, so there must only be one equivalence class. Consider two tuples $\mathbf{x}, \mathbf{y} \in \operatorname{supp}(F)$. If $i \sim j$ then either $x_i = x_j$ and $y_i = y_j$, or $x_i \neq x_j$ and $y_i \neq y_j$; in either case $x_i = y_i$ if and only if $x_j = y_j$. Thus $\mathbf{x} \triangle \mathbf{y}$ is a union of equivalence classes. Since there is only one equivalence class, $\mathbf{x} = \mathbf{y}$ or $\mathbf{x} = \overline{\mathbf{y}}$. We have shown that $|\operatorname{supp}(F)| \leq 2$.

Conversely, if $|\operatorname{supp}(F)| \leq 2$ then either the arity of F is zero, in which case F is certainly in WNEQ, or the support R of F is a subset of $\{\mathbf{y}, \mathbf{z}\}$ for distinct vectors $\mathbf{y}, \mathbf{z} \in \{0,1\}^k$. Pick t with $y_t \neq z_t$. Then for all $\mathbf{x} \in \{0,1\}^k$ we have

$$F(\mathbf{x}) = U(x_t) \left(\prod_{i: y_i = z_i = 0} \mathrm{PIN}_0(x_i) \right) \left(\prod_{i: y_i = z_i = 1} \mathrm{PIN}_1(x_i) \right) \cdot \left(\prod_{i: y_i \neq y_t = z_i} \mathrm{NEQ}(x_t, x_i) \right) \left(\prod_{i: y_i = y_t \neq z_i} \mathrm{EQ}_2(x_t, x_i) \right)$$

so $F \in \mathsf{WNEQ}$. □

Lemma 5.42. *Let $\mathcal{F}$ be a finite weighted constraint language. Assume that $\mathcal{F}$ contains a weight-function that is not in WNEQ and a weight-function that is not IM-terraced. There is a finite set $W \subseteq \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$ such that $\#\mathsf{BIS} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{=2}(\mathcal{F})$ and if $\mathcal{F} \not\subseteq \mathsf{LSM}$ then $\#\mathsf{SAT} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{=2}(\mathcal{F})$.*

Proof. As in Lemma 5.40, we can assume $\mathcal{F}$ is closed under pinnings. Let B, T be as given by Lemma 5.40 applied to $\mathcal{F}$. We first show that $T\mathcal{F}^B$ has the same properties as $\mathcal{F}$ for the purposes of Proposition 5.8: firstly that $T\mathcal{F}^B \not\subseteq \mathsf{WNEQ}$, and secondly that $\mathcal{F} \not\subseteq \mathsf{LSM}$ implies $T\mathcal{F}^B \not\subseteq \mathsf{LSM}$.

For any function $F\colon \{0,1\}^V \to \mathbb{Q}_{\geq 0}$ and any $p \in \{\frac{1}{2}, 1, 2\}$, define $F^p\colon \{0,1\}^V \to \mathbb{R}_{\geq 0}$ by $F^p(\mathbf{x}) = F(\mathbf{x})^p$.

To show $T\mathcal{F}^B \not\subseteq \mathsf{WNEQ}$, let $G\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ be a pinning-minimal weight-function subject to $G \in \mathcal{F} \setminus \mathsf{WNEQ}$. Since WNEQ is closed under taking tensor products, this implies that G is indecomposable. By Lemma 5.41, the support of G has order at most two. Either $T_{00}T_{11} > 0$ and $\operatorname{supp}(T^{\otimes k}G^B) \supseteq \operatorname{supp}(G)$, or $T_{01}T_{10} > 0$ and $\operatorname{supp}(T^{\otimes k}G^B) \supseteq \{\overline{\mathbf{x}} \mid \mathbf{x} \in \operatorname{supp}(G)\}$. In either case $|\operatorname{supp}(T^{\otimes k}G^B)| \geq |\operatorname{supp}(G)| > 2$. If $T^{\otimes k}G^B = G_1 \otimes G_2$ then $G = (S^{\otimes V_1}G_1)^{1/B} \otimes (S^{\otimes V_2}G_2)^{1/B}$ where S is the matrix inverse of T, and V_1 and V_2 are the variable sets of G_1 and G_2 respectively. But G is indecomposable, so G_1 or G_2 has arity zero.[3] We have shown that $T^{\otimes k}G^B$ is an indecomposable weight-function whose support has order greater then two, which by Lemma 5.41 implies that $T^{\otimes k}G^B$ is not in WNEQ.

Assume $\mathcal{F} \not\subseteq \mathsf{LSM}$; we will now argue that $T\mathcal{F}^B \not\subseteq \mathsf{LSM}$. Let H be a weight-function in $\mathcal{F}$ that is pinning-minimal subject to not being log-supermodular. In particular by Lemma 5.36, $\operatorname{supp}(H) \subseteq \{\underline{0}, \mathbf{x}, \overline{\mathbf{x}}, \underline{1}\}$ for some vector $\mathbf{x}$ with a zeros and b ones for some

[3] We defined a function F to be indecomposable if $F = G \otimes H$ implies that G or H has arity zero, for all rational-valued G and H. But we are using the property that an indecomposable function cannot be a tensor product of *real*-valued functions of positive arity. To justify this, note that if F is rational-valued and $F = G \otimes H$ for some real-valued G and H then, as long as F is not identically zero, we have $F = G' \otimes H'$ where $G'(\mathbf{y}) = \sum_{\mathbf{z}} F(\mathbf{y}, \mathbf{z})$ and $H'(\mathbf{z}) = \sum_{\mathbf{y}} F(\mathbf{y}, \mathbf{z}) / \sum_{\mathbf{y},\mathbf{z}} F(\mathbf{y}, \mathbf{z})$. Here $\mathbf{y}$ ranges over configurations of the variable set of G, and $\mathbf{z}$ ranges over configurations of the variable set of H.

$a, b \geq 1$. Hence

$$\begin{pmatrix} (T^{\otimes(a+b)}H^B)(\underline{0}) & (T^{\otimes(a+b)}H^B)(\mathbf{x}) \\ (T^{\otimes(a+b)}H^B)(\overline{\mathbf{x}}) & (T^{\otimes(a+b)}H^B)(\underline{1}) \end{pmatrix}$$
$$= \begin{pmatrix} T^a_{00} & T^a_{01} \\ T^a_{10} & T^a_{11} \end{pmatrix} \begin{pmatrix} H(\underline{0})^B & H(\mathbf{x})^B \\ H(\overline{\mathbf{x}})^B & H(\underline{1})^B \end{pmatrix} \begin{pmatrix} T^b_{00} & T^b_{10} \\ T^b_{01} & T^b_{11} \end{pmatrix}$$

Denote the latter expression by $M_1 M_2 M_3$. Since $H(\underline{0})H(\underline{1}) < H(\mathbf{x})H(\overline{\mathbf{x}})$, the middle matrix M_2 has a negative determinant. The determinants of the neighbouring matrices M_1 and M_3 have the same sign: if $T_{00}T_{11} > T_{01}T_{10}$ they both have a positive determinant, otherwise they both have a negative determinant. Therefore the matrix on the left-hand-side has a negative determinant, and hence $T^{\otimes(a+b)}H^B \in T\mathcal{F}^B$ is not log-supermodular.

Let B, T be as given by Lemma 5.40 applied to $\mathcal{F}$. By Proposition 5.8 there is a finite set of arity 1 weight-functions S such that $\#\mathsf{BIS} \leq_{\mathrm{AP}} \#\mathsf{CSP}(T\mathcal{F}^B \cup S)$, and if $T\mathcal{F}^B \not\subseteq \mathsf{LSM}$ then $\#\mathsf{SAT} \leq_{\mathrm{AP}} \#\mathsf{CSP}(T\mathcal{F}^B \cup S)$. By the choice of B and T we have $\#\mathsf{CSP}(T\mathcal{F}^B \cup S) \leq_{\mathrm{AP}} \#\mathsf{CSP}^W_{=2}(\mathcal{F})$ for some finite set W. □

5.4.4 Proof of Theorem 5.10

Lemma 5.43 extends the #PM-hardness result of Lemma 5.32 to certain weight-functions.

Lemma 5.43. *Let F be a terraced weight-function whose support is not basically binary. Then $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(F)$.*

Proof. By Lemma 5.27, $\mathrm{supp}(F)$ is a delta-matroid. As in the proof of Lemma 5.32, there is an arity 3 copy G of a pinning of F, integers $h(1), h(2), h(3)$, and a vector $\mathbf{x} \in \{0,1\}^3$ such that $\mathrm{supp}(G_{h-\max})$ is the relation $\mathrm{PM}^{\mathbf{x}} = \{\mathbf{x} \oplus (1,0,0), \mathbf{x} \oplus (0,1,0), \mathbf{x} \oplus (0,0,1)\}$. Let $U_1(x_1) = 1/G(1,0,0)$, $U_2(x_2) = 1/G(0,1,0)$, and $U_3(x_3) = 1/G(0,0,1)$. Let $U_1(1-x_1) = U_2(1-x_2) = U_3(1-x_3) = 1$. Then $G(y_1,y_2,y_3)U_1(y_1)U_2(y_2)U_3(y_3) = \mathrm{PM}^{\mathbf{x}}(y_1,y_2,y_3)$ for all $y_1,y_2,y_3 \in \{0,1\}$. Thus $\mathrm{PM}^{\mathbf{x}}$ is a simple weighting of $G_{h-\max}$. In the proof of Lemma 5.32 we showed $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathrm{PM}^{\mathbf{x}}, \mathrm{PIN}_0, \mathrm{PIN}_1)$ (equation (5.2)). Using this, pinning (Lemma 5.35), simple weighting (Lemma 5.26), h-maximisation (Lemma 5.23), and pinning again,

$$\begin{aligned} \#\mathsf{PM} &\leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathrm{PM}^{\mathbf{x}}, \mathrm{PIN}_0, \mathrm{PIN}_1) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathrm{PM}^{\mathbf{x}}) \\ &\leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(G_{h-\max}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(G) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(F). \end{aligned}$$ □

Lemma 5.44 uses h-maximisation to simulate non-IM-terraced weight-functions from non-terraced weight-functions.

Lemma 5.44. *Let $\mathcal{F}$ be a finite weighted constraint language, containing a weight-function whose support is not in* IMconj*, and a weight-function that is not terraced. Then $\#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F} \cup \{F\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F})$ for some weight-function F that is not IM-terraced.*

Proof. Let $G \in \mathcal{F}$ satisfy $\mathrm{supp}(G) \notin \mathsf{IMconj}$. By Lemma 5.36 applied to $\mathrm{supp}(G)$, there exists $\mathbf{p}$ such that, if we let $H = G(\mathbf{p}, \cdot)$, then $\{\mathbf{z}, \overline{\mathbf{z}}\} \subseteq \mathrm{supp}(H) \subsetneq \{\underline{0}, \mathbf{z}, \overline{\mathbf{z}}, \underline{1}\}$ for some $\mathbf{z}$ not equal to $\underline{0}$ or $\underline{1}$. Permuting variables if necessary, there exist $a, b \geq 1$ such that $\mathbf{z} \in \{0,1\}^{a+b}$, with $z_i = 0$ for $1 \leq i \leq a$ and $z_i = 1$ for $a+1 \leq i \leq a+b$.

Define $R = \{\mathbf{z}, \overline{\mathbf{z}}\}$. Then R is a simple weighting of an h-maximisation of H. Specifically, define $h \in \mathbb{Z}^{a+b}$ as follows. If $H(\underline{1}) \neq 0$, let $h(1) = h(a+1) = 1$, and $h(i) = 0$ elsewhere. If $H(\underline{0}) \neq 0$, let $h(1) = h(a+1) = -1$, and $h(i) = 0$ otherwise. Define $W(0) = 1/H(\mathbf{z})$ and $W(1) = 1/H(\overline{\mathbf{z}})$. Then $R(\mathbf{x}) = H_{h-\max}(\mathbf{x})W(x_1)$ for all $\mathbf{x}$. By h-maximisation (Lemma 5.23), pinning (Lemma 5.35) and simple weighting (Lemma 5.26),

$$\#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F} \cup \{R\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F}).$$

If the arity of R is two, then $R = \mathrm{NEQ}$. Let $F' \in \mathcal{F}$ be a weight-function that is not terraced, so there exist $\mathbf{p}, i, j$ such that $F'(\mathbf{p}, \cdot)$ is identically zero, but $F'(\mathbf{p} \oplus \{i\}, \cdot)$ and $F'(\mathbf{p} \oplus \{j\}, \cdot)$ are not parallel. If $p_i \neq p_j$ then F' is not IM-terraced, so the conclusion follows by setting $F = F'$. Let n be the arity of F' and define $F\colon \{0,1\}^n \to \mathbb{Q}_{\geq 0}$ by

$$F(\mathbf{x}) = \sum_y \mathrm{NEQ}(x_i, y) F'(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n) \qquad (\mathbf{x} \in \{0,1\}^n).$$

Note that this implicitly defines F as $[\![\varphi]\!]$ for a $(=2)$-formula φ over $\{F', \mathrm{NEQ}\}$. Furthermore F is not IM-terraced, because $F(\mathbf{p} \oplus S, \cdot) = F'((\mathbf{p} \oplus \{i\}) \oplus S, \cdot)$ for all $S \subseteq \{i, j\}$. By substitution (Lemma 5.21), $\#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F} \cup \{F\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F} \cup \{R\})$, so we are done.

If the arity of R is greater than two then note that

$$\mathrm{EQ}_{2a}(\mathbf{x}, \mathbf{x}') = \sum_{\mathbf{y} \in \{0,1\}^b} F(\mathbf{x}, \mathbf{y}) F(\mathbf{x}', \mathbf{y}) \qquad \text{for all } \mathbf{x}, \mathbf{x}' \in \{0,1\}^a, \text{ and}$$
$$\mathrm{EQ}_{2b}(\mathbf{y}, \mathbf{y}') = \sum_{\mathbf{x} \in \{0,1\}^a} F(\mathbf{x}, \mathbf{y}) F(\mathbf{x}, \mathbf{y}') \qquad \text{for all } \mathbf{y}, \mathbf{y}' \in \{0,1\}^b.$$

Here commas just denote concatenation: $(\mathbf{x}, \mathbf{x}') = (x_1, \ldots, x_a, x'_1, \ldots, x'_a)$ and so on. By substitution (Lemma 5.21) we have $\#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F} \cup \{\mathrm{EQ}_{2a}, \mathrm{EQ}_{2b}\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F} \cup \{R\})$, but EQ_{2a} is not IM-terraced unless $a = 1$, in which case $b > 1$ and EQ_{2b} is not IM-terraced. □

Theorem 5.10. Let $\mathcal{F}$ be a finite weighted constraint language.

(i.) If $\mathcal{F} \subseteq \mathsf{WNEQ}$ or every weight-function in $\mathcal{F}$ is basically binary, then $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(\mathcal{F})$ is in FP.

(ii.) Otherwise, if there is a non-terraced weight-function in $\mathcal{F}$, then we have a similar classification to Proposition 5.8: $\#\mathsf{BIS} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F})$ and if $\mathcal{F} \not\subseteq \mathsf{LSM}$ then $\#\mathsf{SAT} =_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F})$.

(iii.) Otherwise (when neither of the two conditions above hold), if there is a weight-function in $\mathcal{F}$ whose support is not basically binary, then $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F})$.

Proof. (i.) Lemma 5.24(i.).

(ii.) There is a weight-function $F\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ in $\mathcal{F}$ that is not terraced. By definition there is a pinning $F(\mathbf{p},\cdot)$ and there are variables $i, j \in \operatorname{dom}\mathbf{p}$ such that $F(\mathbf{p},\cdot)$ is identically zero but $F(\mathbf{p}\oplus\{i\},\cdot)$ and $F(\mathbf{p}\oplus\{j\},\cdot)$ are not parallel.

If $p_i \neq p_j$, so F is not IM-terraced, set $F' = F$. Otherwise $p_i = p_j$. There are $\mathbf{x}, \mathbf{y} \in \{0,1\}^{\{1,\dots,k\}\setminus\operatorname{dom}\mathbf{p}}$ such that $F(\mathbf{p}\oplus\{i\},\mathbf{x}), F(\mathbf{p}\oplus\{j\},\mathbf{y}) > 0$. But if $p_i = p_j = 0$ then $F((\mathbf{p}\oplus\{i\},\mathbf{x}) \wedge (\mathbf{p}\oplus\{j\},\mathbf{y})) = F(\mathbf{p},\mathbf{x}\wedge\mathbf{y}) = 0$, and if $p_i = p_j = 1$ then $F((\mathbf{p}\oplus\{i\},\mathbf{x}) \vee (\mathbf{p}\oplus\{j\},\mathbf{y})) = F(\mathbf{p},\mathbf{x}\vee\mathbf{y}) = 0$. Hence $\operatorname{supp}(F)$ is not in IMconj. By Lemma 5.44 there is a non-IM-terraced F' such that $\#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F}\cup\{F'\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F})$.

By Lemma 5.42 $\#\mathsf{BIS} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F}\cup\{F'\})$, and $\#\mathsf{SAT} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F}\cup\{F'\})$ if $\mathcal{F}$ contains a weight-function that is not log-supermodular. Also, $\#\mathsf{CSP}^{\geq 0}_{=2}(\mathcal{F}) \leq_{\mathrm{AP}} \#\mathsf{SAT}$ by Lemma 5.24(iii.).

(iii.) There is a weight-function F in $\mathcal{F}$ that does not have basically binary support. By Lemma 5.43 we have $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{=2}(F)$. □

5.4.5 Proof of Theorem 5.13

Lemma 5.45. *For every weight-function G that is pinning-minimal subject to not being terraced, there is a copy $F\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ of G and values $p_1, p_2 \in \{0,1\}$ such that*

- *for all $y_3,\dots,y_k \in \{0,1\}$ we have $F(p_1,p_2,y_3,\dots,y_k) = 0$, and*
- *there exists a configuration $\mathbf{z} \in \{0,1\}^{\{3,\cdots,k\}}$ and a non-singular matrix $T \in \mathbb{Q}^{2\times 2}$ such that for all $x_1, x_2, y_3, \cdots, y_k \in \{0,1\}$ satisfying $(x_1,x_2) \in \{(p_1, 1-p_2), (1-p_1, p_2)\}$ we have*

$$F(x_1,x_2,y_3,\cdots,y_k) = \begin{cases} T_{x_1,y_3} & \text{if } \mathbf{y} = \mathbf{z} \text{ or } \mathbf{y} = \overline{\mathbf{z}} \\ 0 & \text{otherwise.} \end{cases}$$

Proof. (This can be argued as in the proof of Lemma 5.38, but we will derive it from the statement of Lemma 5.38.)

Given a weight-function $G\colon \{0,1\}^V \to \mathbb{Q}_{\geq 0}$ that is pinning-minimal subject to not being terraced, there exists $\mathbf{q}, i, j$ such that $G(\mathbf{q},\cdot)$ is identically zero, but $G(\mathbf{q}\oplus\{i\},\cdot)$ and $G(\mathbf{q}\oplus\{j\},\cdot)$ are not parallel. Let S be the set containing: i if $q_i = 0$, and j if $q_j = 1$. Define $G^S\colon \{0,1\}^V \to \mathbb{Q}_{\geq 0}$ by $G^S(\mathbf{x}) = G(\mathbf{x}\oplus S)$. Then $G^S(\mathbf{q}\oplus S,\cdot)$ is identically zero but $G^S(\mathbf{q}\oplus S\oplus\{i\},\cdot) = G(\mathbf{q}\oplus\{i\},\cdot)$ and $G^S(\mathbf{q}\oplus S\oplus\{j\},\cdot) = G(\mathbf{q}\oplus\{j\},\cdot)$ are not parallel. Therefore G^S is not IM-terraced. Every pinning of G by a non-empty partial configuration is terraced, so every pinning of G^S by a non-empty partial configuration is terraced, and hence IM-terraced. So G^S is pinning-minimal subject to not being IM-terraced. (We can now forget about i and j.)

Applying Lemma 5.38 to G^S we get a bijection $\pi\colon V \to \{1,\dots,k\}$, values $z_3^S,\dots,z_k^S$ and a matrix T such that the weight-function $F^S\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ defined by $F^S(\mathbf{x}) = G^S(\mathbf{x}\circ\pi)$ satisfies the two bullet points of Lemma 5.38. For each $3 \leq i \leq k$, define $z_i = z_i^S$ if $i \notin \pi(S)$, and $z_i = 1 - z_i^S$ if $i \in \pi(S)$. Define $p_1, p_2 \in \{0,1\}$ by setting $p_1 = 1$ if and only if $1 \notin \pi(S)$, and setting $p_2 = 0$ if and only if $2 \notin \pi(S)$. Then the weight-function $F\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ defined by $F(\mathbf{x}) = G(\mathbf{x}\circ\pi)$ satisfies the two bullet points which were to be shown. $\square$

Theorem 5.13. Let $\mathcal{F}$ be a finite weighted constraint language. Assume $\mathcal{F} \not\subseteq \mathsf{WNEQ}$, and that not every weight-function in $\mathcal{F}$ is basically binary, and that not every weight-function in $\mathcal{F}$ is terraced. (This is the same setting as the $\#\mathsf{BIS}$ and $\#\mathsf{SAT}$ reductions in Theorem 5.10.)

Unless all the following conditions hold, there is a finite set $W \subseteq \mathbb{Q}_{\geq 0}\times\mathbb{Q}_{\geq 0}$ such that $\#\mathsf{BIS} \leq_{\mathrm{AP}} \#\mathsf{CSP}^W_{=2}(\mathcal{F})$, and if furthermore $\mathcal{F} \not\subseteq \mathsf{LSM}$ then $\#\mathsf{SAT} =_{\mathrm{AP}} \#\mathsf{CSP}^W_{=2}(\mathcal{F})$.

(i.) Every weight-function $F \in \mathcal{F}$ is IM-terraced.

(ii.) Either the support of every weight-function F in $\mathcal{F}$ is closed under meets — $F(\mathbf{x}), F(\mathbf{y}) \neq 0 \implies F(\mathbf{x}\wedge\mathbf{y}) \neq 0$ — or the support of every weight-function F in $\mathcal{F}$ is closed under joins — $F(\mathbf{x}), F(\mathbf{y}) \neq 0 \implies F(\mathbf{x}\vee\mathbf{y}) \neq 0$.

(iii.) No pinning of the support of a weight-function $F \in \mathcal{F}$ is a copy of EQ_2, that is, there are no $\mathbf{x}, i, j$ satisfying $x_i = x_j$ and $F(\mathbf{x}), F(\mathbf{x}\oplus\{i,j\}) \neq 0$ and $F(\mathbf{x}\oplus\{i\}) = F(\mathbf{x}\oplus\{j\}) = 0$.

Proof. By Lemma 5.35 we can assume $\mathcal{F}$ is closed under pinnings. We will consider each condition in turn.

(i.) Assume that $\mathcal{F}$ is not IM-terraced. The conclusion follows from Lemma 5.42.

(ii.) Assume that condition (ii.) does not hold but condition (i.) holds.

There is a weight-function $G' \in \mathcal{F}$ such that $\mathrm{supp}(G')$ is not closed under joins; let G be a pinning of G' that is pinning-minimal subject to $\mathrm{supp}(G)$ not being closed under joins. By Lemma 5.37 there exists $\mathbf{x} \notin \{\underline{0},\underline{1}\}$ such that $\mathrm{supp}(G) = \{\underline{0},\mathbf{x},\overline{\mathbf{x}}\}$ or $\mathrm{supp}(G) = \{\mathbf{x},\overline{\mathbf{x}}\}$. Taking a copy of G if necessary we can assume that G is a weight-function of arity $a+b$ and $x_1 = \cdots = x_a = 0$ and $x_{a+1} = \cdots = x_{a+b} = 1$, for some $a, b \geq 1$. Suppose for contradiction that $a \geq 2$. Letting $\mathbf{p} = \{1 \mapsto 0, 2 \mapsto 1\}$ we find that $G(\mathbf{p},\cdot)$ is identically zero but $G(\mathbf{p}\oplus\{1\},\cdot)$ and $G(\mathbf{p}\oplus\{2\},\cdot)$ are not parallel, contradicting the fact that G is IM-terraced. Similarly, suppose for contradiction that $b \geq 2$. Letting $\mathbf{p} = \{a{+}1 \mapsto 0, a{+}2 \mapsto 1\}$ we find that $G(\mathbf{p},\cdot)$ is identically zero but $G(\mathbf{p}\oplus\{a+1\},\cdot)$ and $G(\mathbf{p}\oplus\{a+2\},\cdot)$ are non-parallel, which again contradicts the fact that G is IM-terraced.

Pick a non-terraced weight-function $F' \in \mathcal{F}$. By Lemma 5.45 there is a copy $F\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ of F' such that: there exist $p_1, p_2, z_3, z_4, \dots, z_k \in \{0,1\}$ and a non-singular two-by-two matrix T such that $F(\mathbf{p},\cdot)$ is identically zero and for all $\mathbf{x} \in \{(1-p_1,p_2),(p_1,1-p_2)\}$ and all $y_3,\ldots,y_k \in \{0,1\}$ we have

$$F(x_1,x_2,y_3,\cdots,y_k) = \begin{cases} T_{x_1,y_3} & \text{if } \mathbf{y}=\mathbf{z} \text{ or } \mathbf{y}=\overline{\mathbf{z}} \\ 0 & \text{otherwise} \end{cases}$$

We have assumed that condition (i.) holds, so F' is IM-terraced, so $p_1 = p_2$. Permuting the domain $\{0,1\}$ if necessary we can assume $p_1 = p_2 = 0$ without loss of generality.

So $\mathrm{supp}(G)$ is either NEQ or NAND. Define $H\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ by

$$H(x_1,x_2,y_3,\ldots,y_k) = \sum_{t=0,1} G(x_1,t)F(t,x_2,y_3,\ldots,y_k)$$

Denote $F(\{1\mapsto i, 2\mapsto j\},\cdot)$ by F_{ij}, and similarly define H_{ij}. We have:

$$\begin{aligned} H_{10} &= G(1,0)F_{00} + G(1,1)F_{10} \text{ which is identically zero} \\ H_{00} &= G(0,0)F_{00} + G(0,1)F_{10} = G(0,1)F_{10} \\ H_{11} &= G(1,0)F_{01} + G(1,1)F_{11} = G(1,0)F_{01} \end{aligned}$$

Hence H is not IM-terraced. (A related trick, expressing a function with support IMP using OR and NAND, is used in [51].)

We showed (case (i.)) that there is a finite set W such that $\#\mathsf{BIS} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{=2}(\mathcal{F} \cup \{H\})$, and if $\mathcal{F} \not\subseteq \mathsf{LSM}$ we can replace $\#\mathsf{BIS}$ by $\#\mathsf{SAT}$. By substitution (Lemma 5.21), $\#\mathsf{CSP}^{W}_{=2}(\mathcal{F} \cup \{H\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{=2}(\mathcal{F} \cup \{F,G\})$, and by pinning (Lemma 5.35), $\#\mathsf{CSP}^{W}_{=2}(\mathcal{F} \cup \{F,G\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W'}_{=2}(\mathcal{F})$ where $W' = W \cup \{(0,1),(1,0)\}$.

(iii.) Assume that condition (iii.) does not hold but conditions (i.) and (ii.) do hold. By permuting the domain $\{0,1\}$ if necessary we can assume without loss of generality that the support of every weight-function in $\mathcal{F}$ is closed under meets.

Pick a non-terraced weight-function $H \in \mathcal{F}$. We will argue that $\mathrm{supp}(H)$ is not closed under joins. By definition there exist $\mathbf{p}, i, j$ such that $H(\mathbf{p},\cdot)$ is identically zero but $H(\mathbf{p}\oplus\{i\},\cdot)$ and $H(\mathbf{p}\oplus\{j\},\cdot)$ are not parallel. Since H is IM-terraced, $p_i = p_j$. There exist $\mathbf{y} \in \mathrm{supp}(H(\mathbf{p}\oplus\{i\},\cdot))$ and $\mathbf{y}' \in \mathrm{supp}(H(\mathbf{p}\oplus\{j\},\cdot))$. In other words, $H(\mathbf{p}\oplus\{i\},\mathbf{y}), H(\mathbf{p}\oplus\{j\},\mathbf{y}') > 0$. Since $\mathrm{supp}(H)$ is closed under joins, $H((\mathbf{p}\oplus\{i\})\wedge(\mathbf{p}\oplus\{j\}),\mathbf{y}\wedge\mathbf{y}')$. Hence $p_i = p_j = 1$, and $H((\mathbf{p}\oplus\{i\})\vee(\mathbf{p}\oplus\{j\}),\mathbf{y}\vee\mathbf{y}')$. So $\mathrm{supp}(H)$ is not closed under joins.

By the same argument used in the second paragraph of condition (ii.), the fact that H is IM-terraced and not closed under joins implies that there is a pinning G of H of arity 2 not closed under joins, and taking a copy of necessary we can assume $\mathrm{supp}(G) = \mathrm{NAND}$. Let $h(1) = h(2) = 1$ so $\mathrm{supp}(G_{h-\max}) = \mathrm{NEQ}$. Since

$\mathcal{F} \cup \{G_{h-\max}\}$ fails condition (ii.), there is a finite set W such that $\#\mathsf{BIS} \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{=2}(\mathcal{F} \cup \{G_{h-\max}\})$, and if $\mathcal{F} \not\subseteq \mathsf{LSM}$ we can replace $\#\mathsf{BIS}$ by $\#\mathsf{SAT}$.

We will want to use variable weights that are arbitrary powers of two, so it is convenient to hide W at this point. By Lemma 5.26 there is a set of simple weightings $\mathcal{G}$ of weight-functions in $\mathcal{F}$, and a set of simple weightings $\mathcal{G}'$ of $G_{h-\max}$, such that $\#\mathsf{CSP}^{W}_{=2}(\mathcal{F} \cup \{G_{h-\max}\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}_{=2}(\mathcal{G} \cup \mathcal{G}')$. Let

$$P = \{(2^{p_0}, 2^{p_1}) \mid p_0, p_1 \in \mathbb{Z}\}$$

Let $\mathcal{G}''$ be the set of simple weightings G' of G satisfying $G'_{h-\max} \in \mathcal{G}'$. In other words, for all arity 1 weight-functions U, W, if the weight-function defined by $G_{h-\max}(x,y)U(x)W(y)$ is in $\mathcal{G}'$, then the weight-function defined by $G(x,y)U(x)W(y)$ is in $\mathcal{G}''$. Note that $|\mathcal{G}''| = |\mathcal{G}'|$ is finite. By Lemma 5.23,

$$\#\mathsf{CSP}_{=2}(\mathcal{G} \cup \mathcal{G}') \leq_{\mathrm{AP}} \#\mathsf{CSP}^{P}_{=2}(\mathcal{G} \cup \mathcal{G}'')$$

We will show that

$$\#\mathsf{CSP}^{P}_{=2}(\mathcal{G} \cup \mathcal{G}'') \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\{(1,2),(1,1),(2,1)\}}_{=2}(\mathcal{G} \cup \mathcal{G}'' \cup \{\mathrm{EQ}_2\}) \tag{5.4}$$

We are given an instance (V, C, w) of $\#\mathsf{CSP}^{P}_{=2}(\mathcal{G} \cup \mathcal{G}'')$ and error parameter ε. For each $v \in V$ there exists an integer p_v such that $w(v)_1/w(v)_0 = 2^{p_v}$. Let V' be the set with one variable v_i for each $v \in V$ and each $0 \leq i \leq |p_v|$. Define $w' \colon V' \to W$ by $w'(v_0) = (1,1)$ and for all $i > 0$,

$$w'(v_i) = \begin{cases} (1,2) & \text{if } p_v < 0 \\ (2,1) & \text{if } p_v > 0. \end{cases}$$

Modify C as follows to obtain a new list of constraints C': for each $v \in V$, insert constraints $\langle (v_0, v_1), \mathrm{EQ}_2 \rangle, \cdots \langle (v_{|p_v|-1}, v_{|p_v|}), \mathrm{EQ}_2 \rangle$ and replace the two occurrences of v by v_0 and $v_{|p_v|}$. Note that configurations $\mathbf{x}'$ of V' satisfy $\mathrm{wt}_{V',C',w'}(\mathbf{x}') = 0$ unless there exists $\mathbf{x} \in \{0,1\}^V$ such that $x'_{v_i} = x_v$ for all v, i, and in this case $\mathrm{wt}_{V,C,w}(\mathbf{x}') = \mathrm{wt}_{V,C,w}(\mathbf{x})/K$ where $K = \prod_{v \in V} \min(w(v)_0, w(v)_1)$. Hence $Z_{V',C',w'} = Z_{V,C,w} K$. And $Z_{V',C',w'}$ can be approximated by the oracle, passing through ε. Multiplying the result by K gives the AP-reduction (5.4).

To finish, let F be a pinning of a weight-function in $\mathcal{F}$ such that $\mathrm{supp}(F)$ is a copy of EQ_2. Then $F(x,y) = \mathrm{EQ}_2(x,y)F(x,x)$ for all $x, y \in \{0,1\}$ so F is a simple weighting of EQ_2. By Lemma 5.26 there is a finite set W' (which we can assume contains $(0,1)$ and $(1,0)$) such that

$$\#\mathsf{CSP}^{\{(1,2),(1,1),(2,1)\}}_{=2}(\mathcal{G} \cup \mathcal{G}'' \cup \{\mathrm{EQ}_2\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W'}_{=2}(\mathcal{F} \cup \{F\})$$

and $\#\mathsf{CSP}^{W'}_{=2}(\mathcal{F}\cup\{F\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W'}_{=2}(\mathcal{F})$ by pinning (Lemma 5.35). $\square$

5.5 Expressive power of terraced functions

Recall that F is *terraced* if for all partial configurations $\mathbf{p}$ such that the pinning $F(\mathbf{p},\cdot)$ is identically zero, for all $i,j \in \operatorname{dom}\mathbf{p}$ the pinnings $F(\mathbf{p}\oplus\{i\},\cdot)$ and $F(\mathbf{p}\oplus\{j\},\cdot)$ are parallel. In this section we show that we cannot hope to extend Theorem 5.10 by expressing non-terraced functions starting with terraced functions.

Theorem 5.14. No non-terraced function can be defined by a (≤ 2)-formula that only uses terraced functions.

Proof. Given a (≤ 2)-formula φ, we can add useless constraints $\langle (x), U\rangle$ where $U(0) = U(1) = 1$ to bring the degree of every variable to two without affecting $[\![\varphi]\!]$. Because U is terraced, this means we can assume that φ is in fact a $(=2)$-formula.

φ consists of a vertex set V, a constraint list C, and distinct external variables $v_1,\cdots,v_m$. We can split up the definition of $[\![\varphi]\!]$ in the following way. Define $V' = \{(c,i) \mid c = \langle (s_1,\ldots,s_k),F\rangle \in C, 1\leq i\leq k\}$ and define $P\colon \{0,1\}^{V'} \to \mathbb{Q}_{\geq 0}$ by

$$P(\mathbf{x}) = \prod_{c=\langle (s_1,\ldots,s_k),F\rangle\in C} F(\mathbf{x}(c,1),\ldots,\mathbf{x}(c,k)) \qquad (\mathbf{x}\in\{0,1\}^{V'}).$$

Recall that in a $(=2)$-formula the external variables are used exactly once; for each $1\leq j\leq m$ let c_j denote the constraint using v_j, and let i_j denote the position of v_j in the scope of c_j (so if $c_j = \langle (s_1,\ldots,s_k),F\rangle \in C$ then $s_{i_j} = v_j$). For all $\mathbf{x}\in\{0,1\}^{\{(c_1,i_1),\ldots,(c_k,i_k)\}}$ we have

$$[\![\varphi]\!]\,(\mathbf{x}(c_1,i_1),\ldots,\mathbf{x}(c_k,i_k)) = \sum P(\mathbf{x})$$

where $\sum$ denotes the sum over all extensions of $\mathbf{x}$ to V' such that $\mathbf{x}(c,i) = \mathbf{x}(c',j)$ whenever an internal variable v appears in the i'th position in the constraint c and in the j'th position in the constraint c'.

P is a tensor product of terraced functions. To show that P is terraced, it suffices to show that for all terraced functions $F\colon\{0,1\}^{U_F}\to\mathbb{Q}_{\geq 0}$ and $G\colon\{0,1\}^{U_G}\to\mathbb{Q}_{\geq 0}$ the tensor product $F\otimes G$ is terraced (U_F and U_G are assumed disjoint). Consider a partial configuration $\mathbf{p}$ of $U_F\cup U_G$ such that $(F\otimes G)(\mathbf{p},\cdot)$ is identically zero, and let $i,j\in\operatorname{dom}\mathbf{p}$. The comma notation may be misleading here: if we let $\mathbf{p}_F$ be the restriction of $\mathbf{p}$ to $U_F\cap\operatorname{dom}\mathbf{p}$, and let $\mathbf{p}_G$ be the restriction of $\mathbf{p}$ to $U_G\cap\operatorname{dom}\mathbf{p}$, the hypothesis is that $F(\mathbf{p}_F,\mathbf{x}_F)G(\mathbf{p}_G,\mathbf{x}_G)$ is zero for all $\mathbf{x}_F\in\{0,1\}^{U_F\setminus\operatorname{dom}\mathbf{p}}$ and all $\mathbf{x}_G\in\{0,1\}^{U_G\setminus\operatorname{dom}\mathbf{p}}$.

Note that this implies that $F(\mathbf{p}_F,\cdot)$ is identically zero or $G(\mathbf{p}_G,\cdot)$ is identically zero. We deduce that $(F\otimes G)(\mathbf{p}\oplus\{i\},\cdot)$ is parallel to $(F\otimes G)(\mathbf{p}\oplus\{j\},\cdot)$:

- When $(i,j)\in U_F\times U_F$, note that if $F(\mathbf{p}_F,\cdot)$ is identically zero then $F(\mathbf{p}_F\oplus\{i\},\cdot)$ is parallel to $F(\mathbf{p}_F\oplus\{j\},\cdot)$, which implies that $(F\otimes G)(\mathbf{p}\oplus\{i\},\cdot)$ is parallel to $(F\otimes G)(\mathbf{p}\oplus\{j\},\cdot)$. And if $G(\mathbf{p}_G,\cdot)$ is identically zero then $(F\otimes G)(\mathbf{p}\oplus\{i\},\cdot)$ and $(F\otimes G)(\mathbf{p}\oplus\{j\},\cdot)$ are identically zero.

- When $(i,j) \in U_G \times U_G$, note that if $G(\mathbf{p}_G, \cdot)$ is identically zero then $G(\mathbf{p}_G \oplus \{i\}, \cdot)$ is parallel to $G(\mathbf{p}_G \oplus \{j\}, \cdot)$, which implies that $(F \otimes G)(\mathbf{p} \oplus \{i\}, \cdot)$ is parallel to $(F \otimes G)(\mathbf{p} \oplus \{j\}, \cdot)$. And if $F(\mathbf{p}_F, \cdot)$ is identically zero then $(F \otimes G)(\mathbf{p} \oplus \{i\}, \cdot)$ and $(F \otimes G)(\mathbf{p} \oplus \{j\}, \cdot)$ are identically zero.

- When $(i,j) \in U_F \times U_G$ or $(i,j) \in U_G \times U_F$, since either $F(\mathbf{p}_F, \cdot)$ or $G(\mathbf{p}_G, \cdot)$ is identically zero, either $(F \otimes G)(\mathbf{p} \oplus \{i\}, \cdot)$ or $(F \otimes G)(\mathbf{p} \oplus \{j\}, \cdot)$ is identically zero.

We have shown that any tensor product of terraced functions is terraced, and hence that P is terraced. To show that $[\![\varphi]\!]$ is terraced, it suffices to show that the property of being terraced is preserved under summing two variables. This comes down to the following claim. We are given a function $F\colon \{0,1\} \times \{0,1\} \times \{0,1\}^U \times \{0,1\}^{U'} \to \mathbb{Q}_{\geq 0}$ and a configuration $\mathbf{p} \in \{0,1\}^U$. We are given that F is terraced (when considered as an arity $2 + |U| + |U'|$ function), and that $G(\mathbf{p}, \cdot)$ is identically zero where $G\colon \{0,1\}^{U \cup U'} \to \mathbb{Q}_{\geq 0}$ is the function defined by

$$G(\mathbf{p}, \mathbf{x}) = \sum_{t=0}^{1} F(t, t, \mathbf{p}, \mathbf{x}) \qquad (\mathbf{p} \in \{0,1\}^U, \mathbf{x} \in \{0,1\}^{U'}.)$$

We wish to show that $G(\mathbf{p} \oplus \{i\}, \cdot)$ is parallel to $G(\mathbf{p} \oplus \{j\}, \cdot)$. There are two cases.

- $F(s,t,\mathbf{p},\mathbf{x}) = 0$ for all $s,t \in \{0,1\}$ and all $\mathbf{x} \in \{0,1\}^{U'}$. In this case, because F is terraced there are numbers λ, μ not both zero satisfying

  $$\lambda F(s,t,\mathbf{p} \oplus \{i\}, \mathbf{x}) = \mu F(s,t,\mathbf{p} \oplus \{j\}, \mathbf{x}) \text{ for all } \mathbf{x} \in \{0,1\}^{U'} \text{ and all } s,t \in \{0,1\}.$$

  Therefore $\lambda G(\mathbf{p} \oplus \{i\}, \mathbf{x}) = \mu G(\mathbf{p} \oplus \{j\}, \mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^{U'}$.

- $F(s,t,\mathbf{p},\mathbf{x}) > 0$ for some $s,t \in \{0,1\}$ and some $\mathbf{x} \in \{0,1\}^{U'}$. Since $G(\mathbf{p},\mathbf{x}) = 0$ we have $s \neq t$. Let $H\colon \{0,1\}^{U'} \to \mathbb{Q}_{\geq 0}$ be the pinning $F(s,t,\mathbf{p},\cdot)$. Since F is terraced and $F(0,0,\mathbf{p},\cdot)$ is identically zero, $F(0,0,\mathbf{p} \oplus \{i\}, \cdot)$ and $F(0,0,\mathbf{p} \oplus \{j\}, \cdot)$ are constant multiples of H. Since F is terraced and $F(1,1,\mathbf{p},\cdot)$ is identically zero, $F(1,1,\mathbf{p} \oplus \{i\}, \cdot)$ and $F(1,1,\mathbf{p} \oplus \{j\}, \cdot)$ are also constant multiples of H. So $G(\mathbf{p} \oplus \{i\}, \cdot)$ and $G(\mathbf{p} \oplus \{j\}, \cdot)$ are constant multiples of H, which implies that $G(\mathbf{p} \oplus \{i\}, \cdot)$ is parallel to $G(\mathbf{p} \oplus \{j\}, \cdot)$.

We have shown that the class of terraced functions is closed under summing two variables, which implies that $[\![\varphi]\!]$ is terraced. □

5.6 Degree three and higher

In this section we will study $\#\mathsf{CSP}^W_{\leq k}(\mathcal{F})$ for $k > 2$.

Lemma 5.46. *For all $\lambda > 4$, there is no FPRAS for $\#\mathsf{CSP}^{\{(1,\lambda)\}}_{=3}(\mathrm{NAND})$ or $\#\mathsf{CSP}^{\{(\lambda,1)\}}_{=3}(\mathrm{OR})$ unless RP=NP.*

Proof. $\#\mathsf{CSP}_{=3}^{\{(1,\lambda)\}}(\text{NAND})$ has an FPRAS if and only if $\#\mathsf{CSP}_{=3}^{\{(\lambda,1)\}}(\text{OR})$ has an FPRAS, because the two problems just reverse the roles of 0 and 1 in the domain. Lemma 5.6 says that there is no FPRAS for $\#\mathsf{CSP}_{=3}^{\{(1,\lambda)\}}(\text{NAND})$ unless RP = NP. $\square$

Lemma 5.47. *Let $R \subseteq \{0,1\}^V$ be a relation that is pinning-minimal subject to not being degenerate. Then R has arity 2 or $R = \{\mathbf{x}, \overline{\mathbf{x}}\}$ for some $\mathbf{x} \in \{0,1\}^V$.*

Proof. Pick $v \in V$. For any $H \subseteq \{0,1\}^V$ we will denote the pinnings of H by $\{v \mapsto 0\}$ and $\{v \mapsto 1\}$ respectively by H_0 and H_1.

For any product G of relations $G' \subseteq \{0,1\}^{\{v\}}$ and $G'' \subseteq \{0,1\}^{V\setminus\{v\}}$, the pinnings G_0 and G_1 are equal if $G' = \{0,1\}^{\{v\}}$, and otherwise one of them is empty; in other words, G_0 and G_1 are parallel. So for all partial configurations $\mathbf{p}$ such that $R_0(\mathbf{p},\cdot)$ and $R_1(\mathbf{p},\cdot)$ are not parallel, $R(\mathbf{p},\cdot)$ is non-degenerate and hence $\operatorname{dom}\mathbf{p} = \emptyset$ by minimality of R. Since R_0 and R_1 are products of arity 1 relations but R is not, R cannot be $R_0 \times \{0,1\}^{\{v\}}$ or $R_0 \times \{0\}^{\{v\}}$ or $R_1 \times \{1\}^{\{v\}}$; this implies that R_0 and R_1 are not parallel. Hence the weight-function pair (R_0, R_1) is pinning-minimal subject to not being parallel. By Lemma 5.28, $R_0 \cup R_1 = \{\mathbf{y}, \overline{\mathbf{y}}\}$ for some $\mathbf{y}$, which can be chosen to be in R_0.

Let $\mathbf{x} = \mathbf{y} \cup \{v \mapsto 0\}$ be the extension of $\mathbf{y}$ to V with $x_v = 0$. If R_0 and R_1 are $\{\mathbf{y}\}$ and $\{\overline{\mathbf{y}}\}$ respectively then $R = \{\mathbf{x}, \overline{\mathbf{x}}\}$, so we are done. Otherwise R_0 or R_1 is $\{\mathbf{y}, \overline{\mathbf{y}}\}$, but R_0 and R_1 are degenerate by minimality of R. So $\{\mathbf{y}, \overline{\mathbf{y}}\}$ is a copy of a product of arity 1 relations, $\{0,1\}^a \times \{0\}^b \times \{1\}^c$ for some $a, b, c \geq 0$. Taking cardinalities we have $2 = 2^a$ so $a = 1$, and $b = c = 0$ because $y_u \neq 1 - y_u$ for all $u \in V \setminus \{v\}$. Hence R has arity 2. $\square$

Theorem 5.15. Let $\mathcal{F}$ be a finite weighted constraint language and assume that not every weight-function in $\mathcal{F}$ has degenerate support. There exists a finite set of variable weights W such that $\#\mathsf{CSP}^{\geq 0}(\mathcal{F})$ has an FPRAS if and only if $\#\mathsf{CSP}_{\leq 3}^{W}(\mathcal{F})$ has an FPRAS.

Proof. Let F be a weight-function in $\mathcal{F}$ whose support is non-degenerate. Let $G\colon \{0,1\}^k \to \mathbb{Q}_{\geq 0}$ be a minimal non-degenerate pinning of F. Define $H(x_1, x_2) = \sum_{x_3,\cdots,x_k} G(x_1, \cdots, x_k)$ and $R = \operatorname{supp}(H)$. By Lemma 5.47, either the arity of G is 2, or $\operatorname{supp}(G)$ equals $\{\mathbf{x}, \overline{\mathbf{x}}\}$ for some configuration $\mathbf{x}$. In either case the relation R is non-degenerate. Permuting the variables of R if necessary,

$$R \in \{\text{NAND}, \text{OR}, \text{EQ}_2, \text{NEQ}, \text{IMP}\}.$$

Observe that there exists $(X, Y) \in \{0,1\}^2 \setminus R$ with $(X \oplus 1, Y), (X, Y \oplus 1) \in R$. For all $x, y \in \{0,1\}$,

$$R(x,y) = \begin{cases} H(x,y)/H(x, Y \oplus x \oplus X \oplus 1) & \text{if } (X \oplus 1, Y \oplus 1) \notin R, \text{ and} \\ \frac{H(x,y)H(x\oplus 1, Y\oplus 1)H(X\oplus 1, y\oplus 1)}{H(X\oplus 1, Y\oplus 1)H(X, Y\oplus 1)H(X\oplus 1, Y)} & \text{if } (X \oplus 1, Y \oplus 1) \in R. \end{cases}$$

Thus R is a simple weighting of H. For any finite set $W \subseteq \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$, by using both parts of Lemma 5.26, there is a finite set W' such that

$$\#\mathsf{CSP}^{W}_{\leq 3}(\mathcal{F} \cup \{R\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W'}_{\leq 3}(\mathcal{F} \cup \{H\})$$

By substitution (Lemma 5.21) and pinning (Lemma 5.25),

$$\#\mathsf{CSP}^{W'}_{\leq 3}(\mathcal{F} \cup \{H\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W'}_{\leq 3}(\mathcal{F} \cup \{G\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W'}_{\leq 3}(\mathcal{F})$$

Therefore it suffices to show that there is a finite set $W \subseteq \mathbb{Q}_{\geq 0} \times \mathbb{Q}_{\geq 0}$ such that $\#\mathsf{CSP}^{\geq 0}(\mathcal{F})$ has an FPRAS if and only if $\#\mathsf{CSP}^{W}_{\leq 3}(\mathcal{F} \cup \{R\})$ has an FPRAS.

If $R = \mathrm{NEQ}$ then $\mathrm{EQ}_2(x, z) = \sum_y R(x, y)R(y, z)$ for all $x, z \in \{0, 1\}$. By substitution (Lemma 5.21) we have $\#\mathsf{CSP}^{W}_{\leq 3}(\mathcal{F} \cup \{\mathrm{EQ}_2\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{\leq 3}(\mathcal{F})$ for all sets W containing $(1, 1)$. So we can ignore the case $R = \mathrm{NEQ}$. If $R = \mathrm{NAND}$ or $R = \mathrm{OR}$ then by Lemma 5.46, for some finite set W there is no FPRAS for $\#\mathsf{CSP}^{W}_{\leq 3}(R)$ unless $\mathrm{RP} = \mathrm{NP}$. The remaining cases are $R = \mathrm{EQ}_2$ and $R = \mathrm{IMP}$. We will "3-simulate equality" as in [51].

By Lemma 5.24(i.) we may assume that $\mathcal{F}$ is not contained in WNEQ. It follows from [22, Theorem 14, Proposition 25] that there is a finite set S of arity 1 weight-functions such that $\#\mathsf{CSP}(\mathcal{F} \cup S) =_{\mathrm{AP}} \#\mathsf{CSP}(\mathcal{F} \cup \{\mathrm{IMP}\}) =_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}(\mathcal{F})$. Let

$$W = \{(U(0), U(1)) \mid U \in S\} \cup \{(1, 1)\}$$

We will show that $\#\mathsf{CSP}(\mathcal{F} \cup S) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{W}_{\leq 3}(\mathcal{F})$. Given an instance (V, C) of $\#\mathsf{CSP}(\mathcal{F} \cup S)$, construct an instance (V', C') of $\#\mathsf{CSP}_{\leq 3}(\mathcal{F} \cup S \cup \{R\})$ by, for each variable v of degree $d = \deg_C(v)$, replacing the d occurrences of v in C by distinct variables $v_1, \ldots, v_d$, then inserting new constraints $\langle (v_1, v_2), R\rangle, \langle (v_2, v_3), R\rangle, \cdots, \langle (v_d, v_1), R\rangle$ if $d \geq 1$. Note that $Z_{V,C} = Z_{V',C'}$, because for all $d \geq 1$ and all $x_1, \ldots, x_d \in \{0, 1\}$,

$$R(x_1, x_2)R(x_2, x_3) \cdots R(x_d, x_1) = \mathrm{EQ}_d(x_1, \ldots, x_d).$$

Finally, construct an instance (V', C'', w) of $\#\mathsf{CSP}^{W}_{\leq 3}(\mathcal{F} \cup \{R\})$ from (V', C') by replacing each constraint $\langle (v_i), U\rangle \in C$ such that $U \in S$ by a variable weight on v_i; that is, C'' is the list of constraints in C' not using weight-functions in S, and $w : V' \to W$ is defined by

$$w(v_i) = \begin{cases} (U(0), U(1)) & \text{if there is a constraint } \langle (v_i), U\rangle \in C' \text{ with } U \in S\text{, and} \\ (1, 1) & \text{otherwise.} \end{cases}$$

This is well-defined because each v_i is in at most one constraint in C' that does not use R. Moving these factors of $U(0), U(1)$ into variable weights does not affect the weight of any configuration, so $Z_{V',C'',w} = Z_{V',C'} = Z_{V,C}$. Therefore a correct AP-reduction is to query the $\#\mathsf{CSP}^W_{\leq 3}(\mathcal{F} \cup \{R\})$ oracle with (V', C'', w), passing through the error parameter, and returning the result. $\square$

5.7 A tractable region

In this section we will argue that there is a large tractable region for $\#\mathsf{CSP}^{\geq 0}_{\leq d}$. The existence of these FPRASes contrasts with the unbounded problem $\#\mathsf{CSP}^{\geq 0}$. Assuming that $\#\mathsf{BIS}$ does not have an FPRAS, $\#\mathsf{CSP}^{\geq 0}(F)$ has an FPRAS if and only if it is in FP (see Proposition 5.8). But $\#\mathsf{CSP}^{\geq 0}_{\leq d}(\mathcal{F})$ can have an FPRAS even when $\#\mathsf{CSP}^{\geq 0}_{\leq d}(\mathcal{F})$ is $\#\mathsf{P}$-hard.

Proposition 5.48. *[26, Theorem 5.3] If $\mathcal{F}$ is not a subset of* WNEQ *then $\#\mathsf{CSP}^{\geq 0}_{\leq 3}(\mathcal{F})$ is $\#\mathsf{P}$-hard to evaluate exactly.*

Proof. Define U by $U(0) = 1$ and $U(1) = 2$. Using variable weights instead of U, we have $\#\mathsf{CSP}_{\leq 3}(\mathcal{F} \cup \{U\}) \leq_{\mathrm{AP}} \#\mathsf{CSP}^{\geq 0}_{\leq 3}(\mathcal{F})$. Now we can appeal to [26, Theorem 5.3]. Their set "$\mathcal{A}$" does not contain U, and WNEQ is contained in their "$\mathcal{P}$". Hence $\#\mathsf{CSP}_{\leq 3}(F, U)$ is $\#\mathsf{P}$-hard. $\square$

This following argument is inspired by [105]. In particular we use the same quantity J.

Theorem 5.16. Let $d, k \geq 2$. Let F be an arity k weight-function with values in the range $[1, \frac{d(k-1)+1}{d(k-1)-1})$. Then $\#\mathsf{CSP}^{\geq 0}_{\leq d}(F)$ has an FPRAS.

Proof. We will use a path coupling argument on a Markov chain with heat bath dynamics. We will proceed by giving a FPAUS, by which we mean a randomised algorithm that, given an instance (V, C, w) and error parameter $\varepsilon > 0$, outputs a random configuration μ such that the total variation distance of μ from $\pi_{V,C,w}$ is at most ε where $\pi_{V,C,w}(\sigma) = \mathrm{wt}_{V,C,w}(\sigma)/Z_{V,C,w}$, and the algorithm runs in time polynomial in the size of the input and $\log(1/\varepsilon)$.

The FPAUS is to simulate a Markov chain of configurations $(\mathbf{x}_t)_{t=0,1,\cdots}$ and output $\mathbf{x}_T$ for some T to be determined later. For configurations $\mathbf{x} \in \{0,1\}^V$ and variables v we will use the notation $\mathbf{x}[v \mapsto j]$ to mean $\mathbf{x}[v \mapsto j](u) = \mathbf{x}(u)$ for $u \neq v$ and $\mathbf{x}[v \mapsto j](v) = j$. Let $\mathbf{x}_0 \in \{0,1\}^V$ be any configuration. For each $t \geq 1$ let $v_t \in V$ be distributed uniformly at random and let $\mathbf{x}_t$ be distributed according to heat bath dynamics, that is, distributed according to $\pi_{V,C,w}$ conditioned on $\mathbf{x}_t \in \{\mathbf{x}_{t-1}[v_t \mapsto 0], \mathbf{x}_{t-1}[v_t \mapsto 1]\}$. Thus

$$\Pr[\mathbf{x}_t(i) = 1 \mid \mathbf{x}_{t-1}, v_t] = \frac{\mathrm{wt}_{V,C,w}(\mathbf{x}_{t-1}[v \mapsto 1])}{\mathrm{wt}_{V,C,w}(\mathbf{x}_{t-1}[v \mapsto 0]) + \mathrm{wt}_{V,C,w}(\mathbf{x}_{t-1}[v \mapsto 1])}$$

This probability is easy to compute exactly, so each step of the Markov chain can be simulated efficiently.

Consider another Markov chain $(\mathbf{y}_t)_{t\geq 0}$ distributed in the same way as $(\mathbf{x}_t)_{t\geq 0}$, with the optimal coupling given that both chains choose the same variables v_t. So

$$\Pr[\mathbf{x}_t(v_t) \neq \mathbf{y}_t(v_t)|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, v_t] = |\Pr[\mathbf{x}_t(v_t) = 1|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, v_t] - \Pr[\mathbf{y}_t(v_t) = 1|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, v_t]|$$

Define $\beta = \beta(V, C, w) = \max_{\mathbf{x}_0, \mathbf{y}_0 : |\mathbf{x}_0 \triangle \mathbf{y}_0| = 1} \mathbb{E}[d(\mathbf{x}_1, \mathbf{y}_1)]$. Let M be the maximum value taken by F. We will establish the bound

$$\beta \leq 1 - c|V|^{-1} \tag{5.5}$$

for some $c > 0$ depending only on the parameters d, k, M. Then by the General Path Coupling Theorem of [16] the total variation distance from the stationary distribution is at most ε as long as $T \geq \log(|V|\varepsilon^{-1})/\log \beta^{-1} = \mathrm{poly}(|V|, \log \varepsilon^{-1})$. This gives the required FPAUS. Given the FPAUS, there is an FPRAS by [73, Theorem 6.4] in the same way as we discussed earlier in Theorem 3.4 – the self-reducibility is given by pinning (Lemma 5.25).

We will now bound β. Fix configurations $\mathbf{x}_0$ and $\mathbf{y}_0$ that only differ on a single variable u. For all $v_1 \in V$ define

$$E(\mathbf{x}_0, \mathbf{y}_0, v_1) = |\Pr[\mathbf{x}_1(v_1) = 1|\mathbf{x}_0, \mathbf{y}_0, v_1] - \Pr[\mathbf{y}_1(v_1) = 1|\mathbf{x}_0, \mathbf{y}_0, v_1]|$$

Define $W_{ij} = W_{ij}(\mathbf{x}_0, \mathbf{y}_0, v_1) = \mathrm{wt}_{V,C,w}(\mathbf{x}_0[u \mapsto i][v_1 \mapsto j])$ for all $i, j \in \{0, 1\}$. Then

$$\begin{aligned}
E(\mathbf{x}_0, \mathbf{y}_0, v_1) &= |\Pr[\mathbf{x}_1(v_1) = 1|\mathbf{x}_0, \mathbf{y}_0, v_1] - \Pr[\mathbf{y}_1(v_1) = 1|\mathbf{x}_0, \mathbf{y}_0, v_1]| \\
&= \left| \frac{W_{01}}{W_{00} + W_{01}} - \frac{W_{11}}{W_{10} + W_{11}} \right| \\
&= \frac{|W_{00}W_{11} - W_{01}W_{10}|}{W_{00}W_{11} + W_{01}W_{10} + W_{00}W_{10} + W_{01}W_{11}} \\
&\leq \frac{|W_{00}W_{11} - W_{01}W_{10}|}{W_{00}W_{11} + W_{01}W_{10} + 2\sqrt{W_{00}W_{10}W_{01}W_{11}}} \\
&= \frac{|\sqrt{W_{00}W_{11}} - \sqrt{W_{01}W_{10}}|}{\sqrt{W_{00}W_{11}} + \sqrt{W_{01}W_{10}}}
\end{aligned}$$

Let $v_1 \in V \setminus \{u\}$. Denote by $C' = C'(u, v_1) \subseteq C$ the list of constraints with u and v_1 in their scope. For all $i, j \in \{0, 1\}$ let $\mathbf{x}^{ij} = \mathbf{x}[u \mapsto i][v_1 \mapsto j]$, and for all $c = \langle (u_1, \ldots, u_k), F \rangle \in C'(u, v_1)$ define

$$F'_c(i, j) = F(\mathbf{x}^{ij}(u_1), \ldots, \mathbf{x}^{ij}(u_k)).$$

Define $W' = W'(u, v_1)$ by $W'_{ij} = \prod_{c \in C'} F'_c(j, k)$. The other weights depend on u or v_1 alone, so $W'_{00}W'_{11}/W'_{01}W'_{10}$ equals $W_{00}W_{11}/W_{01}W_{10}$ and

$$E(\mathbf{x}_0, \mathbf{y}_0, v_1) \leq \frac{|\sqrt{W'_{00}W'_{11}} - \sqrt{W'_{01}W'_{10}}|}{\sqrt{W'_{00}W'_{11}} + \sqrt{W'_{01}W'_{10}}}$$

For all 2-by-2 matrices G with strictly positive entries, define $J(G) = \frac{1}{4}\log\frac{G_{00}G_{11}}{G_{01}G_{10}}$. Note that the functions F'_i take values in the range $[1, M]$ so $|J(F'_i)| \leq \frac{1}{2}\log M$; also recall that tanh is non-decreasing and subadditive for positive reals, that is, $\tanh(x+y) = \frac{\tanh x + \tanh y}{1+\tanh x \tanh y} \leq \tanh(x) + \tanh(y)$. Hence

$$\begin{aligned}
E(\mathbf{x}_0, \mathbf{y}_0, v_1) &\leq \frac{|\sqrt{W'_{00}W'_{11}} - \sqrt{W'_{01}W'_{10}}|}{\sqrt{W'_{00}W'_{11}} + \sqrt{W'_{01}W'_{10}}} \\
&= \tanh|J(W')| \\
&= \tanh\left|\sum_{i\in C'(u,v_1)} J(F'_i)\right| \\
&\leq |C'(u,v_1)| \tanh\left(\frac{1}{2}\log M\right) \\
&= |C'(u,v_1)|\frac{M-1}{M+1}
\end{aligned}$$

The variable u appears in at most d constraints, each of which contributes at most $k-1$ to $\sum_{v_1}|C'(u,v_1)|$. Rearranging $M < \frac{d(k-1)+1}{d(k-1)-1}$ we get $d(k-1)\frac{M-1}{M+1} < 1$, so

$$\begin{aligned}
\mathbb{E}[d(\mathbf{x}_1, \mathbf{y}_1)] &= 1 - \frac{1}{|V|} + \frac{1}{|V|}\sum_{v_1\in V\setminus\{u\}} E(\mathbf{x}_0, \mathbf{y}_0, v_1) \\
&\leq 1 - \left(1 - d(k-1)\frac{M-1}{M+1}\right)/|V|
\end{aligned}$$

giving the required bound (5.5). □

5.8 Infinite sets of variable weights are sometimes necessary

Theorem 5.13 gives some circumstances in which the set of variable weights in Theorem 5.10 can be taken to be finite. We will show that, assuming that #PM does not have an FPRAS, the set of variable weights cannot always be taken to be finite in Theorem 5.10: the constraint language $\{\mathrm{AtMostOne}_3\}$ falls into case (iii) of Theorem 5.10, but there is no finite set W such that $\#\mathsf{PM} \leq_{\mathrm{AP}} \#\mathsf{CSP}^W_{=2}(\mathrm{AtMostOne}_3)$.

Consider a graph G and a function $c\colon E(G) \to \mathbb{Q}_{\geq 0}$ assigning an edge-weight $c(e)$ to each edge e. Recall that a matching in G is a subset M of the edge set of G such that no two edges in M share a vertex. Define the monomer-dimer partition function

$$\mathrm{Zmd}(G,c) = \sum_{\text{matchings } M}\ \prod_{e\in M} c(e).$$

[70, Corollary 4.4] states that there exists an FPRAS for the monomer-dimer partition function of arbitrary weighted graphs with edge weights presented in unary. The unary representation is just used is to bound $\max_{e\in E(G)} c(e)$ in the proof of [70, Corollary 4.3]. Thus we have:

Lemma 5.49. *[70] There exists a randomised approximation scheme for the monomer-dimer partition function of arbitrary weighted graphs with edge weights presented as fractions of binary integers, with runtime polynomial in $|V(G)|$, $\max_{e\in E(G)} c(e)$, and the inverse error parameter ε^{-1}.*

Theorem 5.17. Let $\mathrm{AtMostOne}_3 = \{\mathbf{x} \in \{0,1\}^3 \mid x_1 + x_2 + x_3 \le 1\}$. Let W be a finite subset of $\mathbb{Q}_{\ge 0} \times \mathbb{Q}_{\ge 0}$. Then $\#\mathsf{CSP}^{W}_{\le 2}(\mathrm{AtMostOne}_3)$ has an FPRAS.

Proof. We will show something slightly stronger: there is a randomised approximation scheme for $\#\mathsf{CSP}^{\ge 0}_{\le 2}(\mathrm{AtMostOne}_3)$ whose runtime is bounded by a polynomial in the size of the input (V, C, w), the inverse error parameter ε^{-1}, and in

$$\delta = \max\{w(v)_1/w(v)_0 \mid v \in V \text{ such that } w(v)_0 > 0\}.$$

Let (V, C, w) be an instance of $\#\mathsf{CSP}^{\ge 0}_{\le 2}(\mathrm{AtMostOne}_3)$ and let $\varepsilon > 0$. To deal with variables used less than twice, let $\mathrm{AtMostOne}_1$ be the arity-1 constraint $\{0, 1\}$, and for each $v \in V$ add $2 - \deg_C(c)$ constraints $\langle (v), \mathrm{AtMostOne}_1\rangle$ to C. Note that these constraints do not affect whether an assignment is satisfying.

Let G be the multigraph whose vertex set is (a set of labels for) the constraint set C, and whose edge set is V, where for each variable $v \in V$, the edge v joins the two vertices corresponding to the constraints in which v appears. If a variable is used twice in the same constraint then we get a vertex with a loop. Let $V_0 = \{v \in V \mid w(v)_0 = 0\}$. Let G' be the graph obtained from G by deleting the edge v and its endpoints, for each $v \in V_0$. Define $c \colon V \setminus V_0 \to \mathbb{Q}_{\ge 0}$ by $c(v) = w(v)_1/w(v)_0$.

The map $\mathbf{x} \mapsto V_0 \cup \{v \in V \setminus V_0 \mid w(v)_1\}$ is a bijection from the set of configurations $\mathbf{x} \colon V \to \{0, 1\}$ with $\mathrm{wt}_{V,C,w}(\mathbf{x}) > 0$ to the set of matchings M of G with $\prod_{v\in M} c(v) > 0$. Letting $K = \left(\prod_{v\in V_0} w(v)_1\right)\left(\prod_{v\in V\setminus V_0} w(v)_0\right)$, we have $\mathrm{wt}_{V,C,w}(\mathbf{x}) = K \prod_{v\in M} w'(v)$. This implies that $Z_{V,C,w} = K \cdot \mathrm{Zmd}(G', c)$.

By Lemma 5.49 there is a randomised approximation scheme that approximates $\mathrm{Zmd}(G', w')$ in time bounded by a polynomial in $|V|$ and $\max_{v\in V\setminus V_0} w'(v) \le \delta$ and ε^{-1}. Multiplying the result by K gives the required approximation to $Z_{V,C,w}$. $\square$

Chapter 6

The complexity of approximating conservative counting CSPs

(This chapter is a revised version of [34] with a modified introduction.)

We study the complexity of approximately solving the weighted counting constraint satisfaction problem $\#\mathsf{CSP}(\mathcal{F})$. In the conservative case, where $\mathcal{F}$ contains all unary functions, there is a classification known for the case in which the domain of functions in $\mathcal{F}$ is Boolean. In this chapter, we give a classification for the more general problem where functions in $\mathcal{F}$ have an arbitrary finite domain. We define the notions of *weak log-modularity* and *weak log-supermodularity*. We show that if $\mathcal{F}$ is weakly log-modular, then $\#\mathsf{CSP}(\mathcal{F})$ is in FP. Otherwise, it is at least as difficult to approximate as $\#\mathsf{BIS}$, the problem of counting independent sets in bipartite graphs. $\#\mathsf{BIS}$ is complete with respect to approximation-preserving reductions for a logically defined complexity class $\#\mathrm{RH}\Pi_1$, and is believed to be intractable. We further sub-divide the $\#\mathsf{BIS}$-hard case. If $\mathcal{F}$ is weakly log-supermodular, then we show that $\#\mathsf{CSP}(\mathcal{F})$ is as easy as a Boolean log-supermodular weighted $\#\mathsf{CSP}$. Otherwise, we show that it is NP-hard to approximate. Finally, we give a full trichotomy for the arity-2 case, where $\#\mathsf{CSP}(\mathcal{F})$ is in FP, or is $\#\mathsf{BIS}$-equivalent, or is equivalent in difficulty to $\#\mathsf{SAT}$, the problem of approximately counting the satisfying assignments of a Boolean formula in conjunctive normal form. We also discuss the algorithmic aspects of our classification.

6.1 Introduction

There has been a lot of work on classifying the computational difficulty of exactly solving $\#\mathsf{CSP}(\mathcal{F})$. For some weighted constraint languages $\mathcal{F}$, this is a computationally easy task, while for others, it is intractable. We will give a brief summary of what is known. For more details, see the surveys of Chen [33] and Lu [81].

First, suppose that the domain D is Boolean (that is, suppose that $D = \{0, 1\}$). For this case, Creignou and Hermann [40] gave a dichotomy for the case in which weights are also in $\{0, 1\}$. In this case, they showed that $\#\mathsf{CSP}(\mathcal{F})$ is in FP (the set of polynomial-time computable function problems) if all of the functions in $\mathcal{F}$ are affine, and that otherwise, it is #P-complete. Dyer, Goldberg, and Jerrum [47] extended this to the case in which weights are non-negative rationals. For this case, they showed that the problem is solvable in polynomial time if (1) every function in $\mathcal{F}$ is expressible as a product of unary functions, equalities and disequalities, or (2) every function in $\mathcal{F}$ is a constant multiple of an affine function. Otherwise, they showed the problem is complete in the complexity class $\mathrm{FP}^{\#\mathrm{P}}$. We will not deal with negative weights in this chapter. However, it is worth mentioning that these results have been extended to the case in which weights can be negative [21], to the case in which they can be complex [26], and to the related class of Holant* problems [27]. Other dichotomies are also known for Holant problems (see [81]).

Next, consider an arbitrary finite domain D. For the case in which weights are in $\{0, 1\}$, Bulatov's breakthrough result [19] showed that $\#\mathsf{CSP}(\mathcal{F})$ is always either in FP or #P-hard. A simplified version was given by Dyer and Richerby [49, 50] who introduced a new criterion called "strong balance". The dichotomy was extended to include non-negative rational weights by Bulatov et al. [20] and then to include all non-negative algebraic weights by Cai, Chen and Lu [24, 25]. Cai, Chen and Lu gave a generalised notion of balance that we will use in this work. Finally, Cai and Chen [23] extended the dichotomy to include all algebraic complex weights. The criterion for the unweighted #CSP dichotomy is known to be decidable [50] and this carries through to non-negative rational weights and non-negative algebraic weights [24]. Decidability is currently open for the complex case.

Much less is known about the complexity of approximately solving $\#\mathsf{CSP}(\mathcal{F})$. Prior to this work, there were no known complexity classifications for approximately solving $\#\mathsf{CSP}(\mathcal{F})$ for the case in which the domain D is not Boolean. Thus, this is the problem that we address in this chapter. Our main result (Theorem 6.4, below) is a complexity classification for the conservative case (where all unary weights are contained in $\mathcal{F}$). Here is an informal description of the result.

- If $\mathcal{F}$ is "weakly log-modular" (a concept we define below) then, for any finite $\mathcal{G} \subset \mathcal{F}$, $\#\mathsf{CSP}(\mathcal{G})$ is in FP.
- Otherwise, there is a finite $\mathcal{G} \subset \mathcal{F}$ such that $\#\mathsf{CSP}(\mathcal{G})$ is at least as hard to approximate as #BIS. Furthermore,
  - if $\mathcal{F}$ is "weakly log-supermodular" (again, defined below) then, for any finite $\mathcal{G} \subset \mathcal{F}$, there is a finite set $\mathcal{G}'$ of log-supermodular functions on the Boolean domain such that $\#\mathsf{CSP}(\mathcal{G})$ is as easy to approximate as $\#\mathsf{CSP}(\mathcal{G}')$;
  - otherwise, there is a finite $\mathcal{G} \subset \mathcal{F}$ such that $\#\mathsf{CSP}(\mathcal{G})$ is as hard to approximate as #SAT.

Informally, $\mathcal{F}$ is weakly log-supermodular if, for every binary function F that can be expressed using functions in $\mathcal{F}$, every projection of F onto two domain elements is log-supermodular (see Definition 6.3). Thus, in some sense, our result shows that all the difficulty of approximating conservative weighted constraint satisfaction problems arises in the Boolean case. Even when the domain D is larger, approximations which are #SAT-equivalent are #SAT-equivalent precisely because of intractable Boolean problems which arise as sub-problems.

In addition to the complexity classifications described above (FP versus #BIS-hard and "as easy as a Boolean log-supermodular problem" versus #SAT-equivalent) we also give a full trichotomy for the binary case (i.e., where all functions in $\mathcal{F}$ have arity 1 or 2).

- If $\mathcal{F}$ is weakly log-modular then, for any finite $\mathcal{G} \subset \mathcal{F}$, #CSP($\mathcal{G}$) is in FP.
- Otherwise, if $\mathcal{F}$ is weakly log-supermodular, then
  - for every finite $\mathcal{G} \subset \mathcal{F}$, #CSP($\mathcal{G}$) is as easy to approximate as #BIS and
  - there is a finite $\mathcal{G} \subset \mathcal{F}$ such that #CSP($\mathcal{G}$) is as hard to approximate as #BIS.
- Otherwise, there is a finite $\mathcal{G} \subset \mathcal{F}$ such that #CSP($\mathcal{G}$) is as hard to approximate as #SAT.

The final section of this chapter discusses the algorithmic aspects of our classification for the case in which $\mathcal{F}$ is the union of a finite, weighted constraint language $\mathcal{H}$ and the set of all unary functions. In particular, we give an algorithm that takes $\mathcal{H}$ as input and correctly makes one of the following deductions:

1. #CSP($\mathcal{G}$) is in FP for every finite $\mathcal{G} \subset \mathcal{F}$;
2. #CSP($\mathcal{G}$) is LSM-easy for every finite $\mathcal{G} \subset \mathcal{F}$ and #BIS-hard for some such $\mathcal{G}$;
3. #CSP($\mathcal{G}$) is #BIS-easy for every finite $\mathcal{G} \subset \mathcal{F}$ and #BIS-equivalent for some such $\mathcal{G}$;
4. #CSP($\mathcal{G}$) is #SAT-easy for all finite $\mathcal{G} \subset \mathcal{F}$ and #SAT-equivalent for some such $\mathcal{G}$.

Further, if every function in $\mathcal{H}$ has arity at most 2, the output is not deduction 2.

6.1.1 Previous work

The first contribution of this chapter is to show that, if $\mathcal{F}$ is weakly log-modular then, for any finite $\mathcal{G} \subset \mathcal{F}$, #CSP($\mathcal{G}$) is in FP. Otherwise, there is a finite $\mathcal{G} \subset \mathcal{F}$ for which #CSP($\mathcal{G}$) is at least as hard to approximate as #BIS. We also show that, if $\mathcal{F}$ is not weakly log-supermodular, then there is a finite $\mathcal{G} \subset \mathcal{F}$, such that #CSP($\mathcal{G}$) is #SAT-hard. This work is presented in Sections 6.2 and 6.3 below and builds on two strands of previous work.

- The hardness results build on the approximation classification in the Boolean case [22] and, in particular, on the key role played by log-supermodular functions.
- The easiness results build on the classification of the exact evaluation of #CSP($\mathcal{F}$) in the general case [24], and in particular on the key role played by "balance".

The second contribution of this chapter is to show that, if $\mathcal{F}$ is weakly log-supermodular, then, for any finite $\mathcal{G} \subset \mathcal{F}$, there is a finite set $\mathcal{G}'$ of log-supermodular functions on the Boolean domain such that $\#\mathsf{CSP}(\mathcal{G})$ is as easy to approximate as $\#\mathsf{CSP}(\mathcal{G}')$. This builds on three key studies of the complexity of optimisation CSPs by Takhanov [94, 95], Cohen, Cooper and Jeavons [35] and Kolmogorov and Živný [77, 78]. In all three cases, we use their arguments and ideas, and not merely their results. Thus, we delve into these three papers in some detail.

Our final contribution is the trichotomy for the binary case. This relies additionally on work of Rudolf and Woeginger [88] on decomposing matrices known as Monge matrices.

6.1.2 Proof outline

The previous section described each part of the classification, and where we use previous results. Here is an outline of how the these previous results are used.

The hardness results follow quickly from the Boolean case; weak log-modularity and weak log-supermodularity are conditions about Boolean subdomains, and we can pick out these problems using unary functions. If a conservative constraint language $\mathcal{F}$ is not weakly log-modular the classification in [22] easily gives a reduction from $\#\mathsf{BIS}$ to $\#\mathsf{CSP}(\mathcal{G})$ for some finite $\mathcal{G} \subset \mathcal{F}$; and if $\mathcal{F}$ instead fails to be weakly log-supermodular, we get a reduction from $\#\mathsf{SAT}$ (Theorem 6.7).

For the polynomial-time tractability result we show that if $\mathcal{F}$ is weakly log-modular then it satisfies the balance condition in [24], giving a polynomial-time algorithm for $\#\mathsf{CSP}(\mathcal{G})$ for each finite subset $\mathcal{G} \subset \mathcal{F}$. Indeed if $\mathcal{F}$ is not balanced, by definition of balance there is a pps-formula witnessing that $\mathcal{F}$ is not balanced; we show that this witness can be modified using unary functions to produce a witness that $\mathcal{F}$ is not weakly log-modular. (Theorem 6.12)

For the second contribution we need to show that if $\mathcal{F}$ is weakly log-supermodular then we can reduce $\#\mathsf{CSP}(\mathcal{G})$ to $\#\mathsf{CSP}(\mathcal{G}')$ for some finite set $\mathcal{F}'$ of log-supermodular functions on the Boolean domain. Here we use a connection to optimisation. We show that if a (translated version of) a constraint language $\mathcal{F}$ does not satisfy the tractability condition of Kolmogorov and Živný's result, then $\mathcal{F}$ is not weakly log-supermodular (Lemma 6.27). The proof of this uses the following trick. By repeating constraints k times for large k, most of the contribution to the partition function comes from the highest weight configurations:

$$\left(\sum_{\mathbf{y}} \prod_{i=1}^{m} F_i(\mathbf{y})^k\right)^{1/k} \to \max_{\mathbf{y}} \prod_{i=1}^{m} F_i(\mathbf{y}) \qquad \text{as } k \to \infty.$$

The expression on the right can then be translated into an instance of the optimisation problems studied by Kolmogorov and Živný.

Next, where Kolmogorov and Živný reduce to submodular function minimisation problems, we argue that essentially the same transformation reduces #CSPs to a problem of the form $\#\mathsf{CSP}(\mathcal{G}')$ where $\mathcal{G}'$ are log-supermodular functions on the Boolean domain, as required. (See Lemma 6.41.)

The Monge matrix decomposition then allows us to transform the $\#\mathsf{CSP}(\mathcal{G})$ into an even more special form, using only Boolean log-supermodular functions of arity 2. This gives a reduction to $\#\mathsf{BIS}$. (See Lemma 6.42.)

We get an algorithmic classification carefully applying the decidability results from the classifications we use. One complication here is that conservative constraint languages are infinite. We prove that it suffices to check a certain finite subset. (See Lemma 6.44)

6.1.3 Preliminaries and statement of results

Let D be a finite domain with $|D| \geq 2$. It will be convenient to refer to the set $\mathrm{Func}_k(D,R)$ of all functions $D^k \to R$ for some codomain R, and the set $\mathrm{Func}(D,R) = \bigcup_{k=0}^{\infty} \mathrm{Func}_k(D,R)$. Let EQ be the binary equality function defined by $\mathrm{EQ}(x,x) = 1$ and $\mathrm{EQ}(x,y) = 0$ for $x \neq y$. When $D = \{0,1\}$, we called this EQ_2 in Section 1.7.2. Recall that we also defined $\mathrm{NEQ} = \{(0,1),(1,0)\}$; this can be considered as a weight-function in $\mathrm{Func}_2(\{0,1\},\{0,1\})$.

This chapter uses a slightly different definition of $\#\mathsf{CSP}(\mathcal{F})$ from Section 1.3, in terms of atomic formulas and pps-formulas. The following definitions are from [22]. Let $\mathcal{F}$ be a subset of $\mathrm{Func}(D,R)$. Let $V = \{v_1,\dots,v_n\}$ be a set of variables. An atomic formula has the form $\varphi = G(v_{i_1},\dots,v_{i_a})$ where $G \in \mathcal{F}$, $a = a(G)$ is the arity of G, and $(v_{i_1},v_{i_2},\dots,v_{i_a}) \in V^a$ is called a "scope". Note that repeated variables are allowed. The function $F_\varphi \colon D^n \to R$ represented by the atomic formula $\varphi = G(v_{i_1},\dots,v_{i_a})$ is just $F_\varphi(\mathbf{x}) = G(\mathbf{x}(v_{i_1}),\dots,\mathbf{x}(v_{i_a}))$, where $\mathbf{x}\colon \{v_1,\dots,v_n\} \to D$ is an assignment to the variables. To simplify the notation, we write $x_j = \mathbf{x}(v_j)$ so

$$F_\varphi(\mathbf{x}) = G(x_{i_1},\dots,x_{i_a}).$$

A pps-formula ("primitive product summation formula") is a finite summation of a finite product of atomic formulas. A pps-formula ψ over $\mathcal{F}$ in variables $V' = \{v_1,\dots,v_{n+k}\}$ has the form

$$\psi = \sum_{v_{n+1},\dots,v_{n+k}} \prod_{j=1}^{m} \varphi_j,$$

where φ_j are all atomic formulas over $\mathcal{F}$ in the variables V'. (The variables V are free, and the others, $V' \setminus V$, are bound.) The formula ψ specifies a function $F_\psi \colon D^n \to R$ in the following way:

$$F_\psi(\mathbf{x}) = \sum_{\mathbf{y}\in D^k} \prod_{j=1}^{m} F_{\varphi_j}(\mathbf{x},\mathbf{y}),$$

where $\mathbf{x}$ and $\mathbf{y}$ are assignments $\mathbf{x}\colon \{v_1,\dots,v_n\} \to D$ and $\mathbf{y}\colon \{v_{n+1},\dots,v_{n+k}\} \to D$. The *functional clone* $\langle\mathcal{F}\rangle_\#$ *generated by* $\mathcal{F}$ is the set of all functions that can be represented by a pps-formula over $\mathcal{F} \cup \{\mathrm{EQ}\}$. Crucially, $\langle\langle\mathcal{F}\rangle_\#\rangle_\# = \langle\mathcal{F}\rangle_\#$ [22, Lemma 1]; we will rely on this transitivity property implicitly.

Definition 6.1. A weighted constraint language $\mathcal{F}$ is *conservative* if $\mathcal{U}_D \subseteq \mathcal{F}$, where $\mathcal{U}_D = \mathrm{Func}_1(D, \mathbb{Q}_{\geq 0})$.

Definition 6.2. A weighted constraint language $\mathcal{F}$ is *weakly log-modular* if, for all binary functions $F \in \langle \mathcal{F} \rangle_\#$ and elements $a, b \in D$,

$$\begin{aligned} &F(a,a)F(b,b) = F(a,b)F(b,a), \text{ or} \\ &F(a,a) = F(b,b) = 0, \text{ or} \\ &F(a,b) = F(b,a) = 0. \end{aligned} \tag{6.1}$$

Definition 6.3. $\mathcal{F}$ is *weakly log-supermodular* if, for all binary functions $F \in \langle \mathcal{F} \rangle_\#$ and elements $a, b \in D$,

$$F(a,a)F(b,b) \geq F(a,b)F(b,a) \quad \text{or} \quad F(a,a) = F(b,b) = 0. \tag{6.2}$$

Recall the definitions of log-supermodular functions and LSM from Section 1.5.5. It is known [22, Lemma 7] that $\langle \mathsf{LSM} \rangle_\# = \mathsf{LSM}$.

The definition of $\#\mathsf{CSP}(\mathcal{F})$ used in this chapter is as follows.

Name. $\#\mathsf{CSP}(\mathcal{F})$.

Instance. A pps-formula ψ consisting of a product of m atomic $\mathcal{F}$-formulas over n free variables $\mathbf{x}$. (Thus, ψ has no bound variables.)

Output. The value $\sum_{\mathbf{x} \in D^n} F_\psi(\mathbf{x})$ where F_ψ is the function defined by ψ.

As in [22] (and other works) we take the size of a $\#\mathsf{CSP}(\mathcal{F})$ instance to be $n + m$, where n is the number of (free) variables and m is the number of weighted constraints (atomic formulas). In unweighted versions of CSP and $\#\mathsf{CSP}$, we can just use n as the size of an instance, since the number of constraints can be bounded by a polynomial in the number of variables. However, in weighted cases, the multiplicity of constraints matters so we cannot bound m in terms of n. In this chapter, we typically denote an instance of $\#\mathsf{CSP}(\mathcal{F})$ by I and the output by $Z(I)$.

The notion of pps-definability described earlier is closely related to AP-reductions. In particular, [22, Lemma 17] shows that $G \in \langle \mathcal{F} \rangle_\#$ implies that $\#\mathsf{CSP}(\mathcal{F}, G) \leq_{\mathrm{AP}} \#\mathsf{CSP}(\mathcal{F})$. We will use this fact without comment.

We say that a counting problem $\#X$ is $\#Y$-easy if $\#X \leq_{\mathrm{AP}} \#Y$ and that it is $\#Y$-hard if $\#Y \leq_{\mathrm{AP}} \#X$. A problem $\#X$ is LSM-*easy* if there is a finite, weighted constraint language $\mathcal{F} \subset \mathsf{LSM}$ such that $\#X \leq_{\mathrm{AP}} \#\mathsf{CSP}(\mathcal{F})$.

We now state our main theorem. Note that we have only defined the problem $\#\mathsf{CSP}(\mathcal{F})$ for finite languages whereas conservative languages are, by definition, infinite.

Theorem 6.4. *Let $\mathcal{F}$ be a conservative weighted constraint language taking values in $\mathbb{Q}_{\geq 0}$.*

1. *If $\mathcal{F}$ is weakly log-modular then $\#\mathsf{CSP}(\mathcal{G})$ is in* FP *for every finite $\mathcal{G} \subset \mathcal{F}$.*

2. *If $\mathcal{F}$ is weakly log-supermodular but not weakly log-modular, then* #$\mathsf{CSP}(\mathcal{G})$ *is* LSM*-easy for every finite* $\mathcal{G} \subset \mathcal{F}$ *and* #BIS*-hard for some such* $\mathcal{G}$.

3. *If $\mathcal{F}$ is weakly log-supermodular but not weakly log-modular and consists of functions of arity at most two, then* #$\mathsf{CSP}(\mathcal{G})$ *is* #BIS*-easy for every finite* $\mathcal{G} \subset \mathcal{F}$ *and* #BIS*-equivalent for some such* $\mathcal{G}$.

4. *If $\mathcal{F}$ is not weakly log-supermodular, then* #$\mathsf{CSP}(\mathcal{G})$ *is* #SAT*-easy for every finite* $\mathcal{G} \subset \mathcal{F}$ *and* #SAT*-equivalent for some such* $\mathcal{G}$.

In particular, among conservative #CSPs, there are no new complexity classes below #BIS or above LSM; furthermore there is a trichotomy for conservative weighted constraint languages with no functions of arity greater than two.

The #BIS-hardness and #SAT-equivalence are proved in Section 6.2, where they are restated as Theorem 6.7. The membership in FP is established as Theorem 6.12 at the end of Section 6.3. LSM-easiness and #BIS-easiness are established by Theorem 6.43 at the end of Section 6.6. Algorithmic aspects are discussed in Section 6.7.

6.2 Hardness results

Our hardness results use the following result from [22], which we stated before as Proposition 5.8 but which we can now state in terms of functional clones:

Lemma 6.5. *[22, Theorem 18] Let $\mathcal{F}$ be a finite, weighted constraint language with* $D = \{0, 1\}$.

- *If* $\mathcal{F} \subset \langle \mathrm{NEQ}, \mathcal{U}_{\{0,1\}} \rangle_{\#}$ *then, for any finite* $S \subset \mathcal{U}_{\{0,1\}}$, #$\mathsf{CSP}(\mathcal{F} \cup S)$ *has an FPRAS.*

- *If* $\mathcal{F} \not\subset \langle \mathrm{NEQ}, \mathcal{U}_{\{0,1\}} \rangle_{\#}$*, then there is a finite* $S \subset \mathcal{U}_{\{0,1\}}$ *such that* #$\mathsf{CSP}(\mathcal{F} \cup S)$ *is* #BIS*-hard.*

- *If* $\mathcal{F} \not\subset \langle \mathrm{NEQ}, \mathcal{U}_{\{0,1\}} \rangle_{\#}$ *and* $\mathcal{F} \not\subset \mathsf{LSM}$*, then there is a finite* $S \subset \mathcal{U}_{\{0,1\}}$ *such that* #$\mathsf{CSP}(\mathcal{F} \cup S)$ *is* #SAT*-hard.*

(As in Proposition 5.8, we restrict to rationals for simplicity. This is justified because the relevant constructions in [22] only use field operations.)

In fact we will only use the following special case of Lemma 6.5.

Lemma 6.6. *[22, Theorem 18] Let F be a function in* $\mathrm{Func}_2(\{0, 1\}, \mathbb{Q}_{\geq 0})$.

- *If* $F \notin \langle \mathrm{NEQ}, \mathcal{U}_{\{0,1\}} \rangle_{\#}$ *then* $\{F\} \cup \mathcal{U}_{\{0,1\}}$ *is* #BIS*-hard.*

- *If* $F \notin \langle \mathrm{NEQ}, \mathcal{U}_{\{0,1\}} \rangle_{\#} \cup \mathsf{LSM}$ *then* $\{F\} \cup \mathcal{U}_{\{0,1\}}$ *is* #SAT*-hard.*

Our hardness results now follow from Lemma 6.6.

Theorem 6.7. *Let $\mathcal{F}$ be a conservative weighted constraint language taking values in $\mathbb{Q}_{\geq 0}$.*

- *If $\mathcal{F}$ is not weakly log-modular, there is a finite $\mathcal{G} \subset \mathcal{F}$ such that $\#\mathsf{CSP}(\mathcal{G})$ is $\#\mathsf{BIS}$-hard.*
- *If $\mathcal{F}$ is not weakly log-supermodular, there is a finite $\mathcal{G} \subset \mathcal{F}$ such that $\#\mathsf{CSP}(\mathcal{G})$ is $\#\mathsf{SAT}$-hard.*
- *For all finite $\mathcal{G} \subset \mathcal{F}$, $\#\mathsf{CSP}(\mathcal{G})$ is $\#\mathsf{SAT}$-easy.*

Proof. First, we establish the hardness results.

Suppose that $\mathcal{F}$ is not weakly log-modular. Let $H \in \langle \mathcal{F} \rangle_\#$ be a function violating (6.1) and let a and b be the relevant domain elements, which must be distinct. Let $\varphi\colon \{0,1\} \to D$ be a unary function with $\varphi(0) = a$ and $\varphi(1) = b$. Define $H_\varphi\colon \{0,1\}^2 \to \mathbb{Q}_{\geq 0}$ by $H_\varphi(x,y) = H(\varphi(x), \varphi(y))$. The following three equations must all fail to hold:

$$H_\varphi(0,0)H_\varphi(1,1) = H_\varphi(0,1)H_\varphi(1,0)$$
$$H_\varphi(0,0) = H_\varphi(1,1) = 0$$
$$H_\varphi(0,1) = H_\varphi(1,0) = 0.$$

By [22, Remark 14], every binary function in $\langle \mathrm{NEQ}, \mathcal{U}_{\{0,1\}} \rangle_\#$ has one of three forms: $U_1(x)U_2(y)$, $U(x)\mathrm{EQ}(x,y)$ or $U(x)\mathrm{NEQ}(x,y)$. Therefore, $H_\varphi \notin \langle \mathrm{NEQ}, \mathcal{U}_{\{0,1\}} \rangle_\#$. By Lemma 6.6 there is a finite set $S \subset \mathcal{U}_{\{0,1\}}$ such that $\#\mathsf{BIS} \leq_{\mathrm{AP}} \#\mathsf{CSP}(H_\varphi, S)$.

For each $U \in \mathcal{U}_{\{0,1\}}$, define $U_{\varphi^{-1}} \in \mathcal{U}_D$ by

$$U_{\varphi^{-1}}(x) = \begin{cases} U(0) & \text{if } x = a, \\ U(1) & \text{if } x = b, \\ 0 & \text{otherwise.} \end{cases}$$

Let $E(0) = E(1) = 1$ so that $E_{\varphi^{-1}}$ is the characteristic function of $\{a,b\} \subseteq D$. Let $S' = \{U_{\varphi^{-1}} \mid U \in S \cup \{E\}\}$. Note that $\{H\} \cup S' \subset \langle F, \mathcal{U}_D \rangle_\#$ is finite.

We describe a reduction from $\#\mathsf{CSP}(H_\varphi, S)$ to $\#\mathsf{CSP}(H, S')$. Given an instance I of $\#\mathsf{CSP}(H_\varphi, S)$, replace each use of H_φ by H, and each use of $U \in S$ by $U_{\varphi^{-1}} \in S'$, and introduce an atomic formula $E_{\varphi^{-1}}(v)$ for each variable v, to obtain a new instance I' of $\#\mathsf{CSP}(H, S')$ with $Z(I) = Z(I')$. Thus $\#\mathsf{CSP}(H, S')$ is $\#\mathsf{BIS}$-hard.

A similar argument shows that $\mathcal{F}$ is $\#\mathsf{SAT}$-hard if it is not weakly log-supermodular. In this case, we start with a function $H \in \langle \mathcal{F} \rangle_\#$ violating (6.2) on the elements $a, b \in D$. Defining φ and H_φ as above, we find that $H_\varphi \notin \mathsf{LSM}$. Since H also violates (6.1) on a, b, the argument above establishes $H_\varphi \notin \langle \mathrm{NEQ}, \mathcal{U}_{\{0,1\}} \rangle_\#$. By Lemma 6.6 there is a finite set $S \subset \mathcal{U}_{\{0,1\}}$ such that $\#\mathsf{SAT} \leq_{\mathrm{AP}} \#\mathsf{CSP}(H_\varphi, S)$. We then proceed as before.

$\#\mathsf{SAT}$-easiness follows from the construction in Section 3 of [46], which shows that any problem in #P is $\#\mathsf{SAT}$-easy. The weighted counting CSPs that we deal with here are equivalent, by [20], to unweighted ones, which are in #P. ☐

6.3 Balance and weak log-modularity

In this section we show that weak log-modularity implies tractability, by showing that every weakly log-modular weighted constraint language is balanced in the following sense.

We may associate a matrix $\mathbf{M}$ with an undirected bipartite graph $G_{\mathbf{M}}$ whose vertex partition consists of the set of rows R and columns C of $\mathbf{M}$. A pair $(r, c) \in R \times C$ is an edge of $G_{\mathbf{M}}$ if, and only if, $\mathbf{M}_{rc} \neq 0$. A *block* of $\mathbf{M}$ is a submatrix whose rows and columns form a connected component in $G_{\mathbf{M}}$. $\mathbf{M}$ has block-rank 1 if all its blocks have rank 1.

We say that a weighted constraint language $\mathcal{F}$ is *balanced* [24] if, for every function $F(x_1, \dots, x_n) \in \langle \mathcal{F} \rangle_\#$ with arity $n \geq 2$, and every k with $0 < k < n$, the $|D|^k \times |D|^{n-k}$ matrix $F((x_1, \dots, x_k), (x_{k+1}, \dots, x_n))$ has block-rank 1. (This notion reduces to Dyer and Richerby's notion of "strong balance"[50] in the unweighted case.)

A function $F \colon \{0,1\}^n \to \mathbb{R}$ has *rank 1* if it has the form $F(x_1, \dots, x_k) = U_1(x_1) \cdots U_k(x_k)$.

We associate with any function $F \colon \{0,1\}^2 \to \mathbb{R}$, the matrix $M_F \in \mathbb{R}^{2\times 2}$ defined by $(M_F)_{ij} = F(i,j)$.

Lemma 6.8. *Let $\mathbf{M} \in \mathbb{R}^{k \times k}$. Let $F \colon \{0,1\}^k \to \mathbb{R}$. Let $T \in \mathbb{R}^{2\times 2}_{\geq 0}$ be non-singular.*

1. *A 2×2 matrix $\mathbf{M}$ has block-rank 1 if and only if it has rank 1 or it has at most two non-zero entries. $F \colon \{0,1\}^2 \to \mathbb{R}$ has rank 1 if and only if* $\det M_F = 0$.

2. *$\mathbf{M}$ has block-rank 1 if and only if the matrix*

$$N_{M,\mathbf{u},\mathbf{u}',\mathbf{v},\mathbf{v}'} = \begin{pmatrix} \mathbf{M}(\mathbf{u},\mathbf{v}) & \mathbf{M}(\mathbf{u},\mathbf{v}') \\ \mathbf{M}(\mathbf{u}',\mathbf{v}) & \mathbf{M}(\mathbf{u}',\mathbf{v}') \end{pmatrix}$$

   has block-rank 1 for every $\mathbf{u}, \mathbf{u}', \mathbf{v}, \mathbf{v}'$.

3. *(Topkis's theorem) If F is strictly positive and not of rank 1, there is a function $F' \colon \{0,1\}^2 \to \mathbb{R}$ of the following form that is not of rank 1.*

$$F'(x_i, x_j) = F(c_1, \dots, c_{i-1}, x_i, c_{i+1}, \dots, c_{j-1}, x_j, c_{j+1}, \dots, c_k).$$

   Here $1 \leq i < j \leq k$, and each c_ℓ is a fixed element of $\{0,1\}$.

4. *F has rank 1 if and only if $T^{\otimes k} F$ has rank 1.*

Proof. 1. A 2×2 matrix that has block-rank 1 either has rank 1 or is diagonal or anti-diagonal so has two zeroes. Conversely, a matrix that has rank 1 has no submatrix whose rank exceeds 1, so has block-rank 1. A matrix with two or more zeroes has no 2×2 block so can only have blocks of rank 1.

For the second statement, if F has rank 1 then there are unary functions U_0 and U_1 so that $F(x,y) = U_0(x)U_1(y)$, which implies that $\det M_F = 0$. Going the other way, if F is identically 0 then it has rank 1. Otherwise, suppose $F(i,j) \neq 0$.

Let $U_0(x) = F(x, j)$ and $U_1(y) = F(i, y)/F(i, j)$. If $\det M_F = 0$ then $F(x, y) = U_0(x)U_1(y)$, so F has rank 1.

2. [50, Lemma 38].

3. Say that a strictly positive function F is *log-modular* if $f = \log F$ is modular: that is, $F(\mathbf{x} \vee \mathbf{y})F(\mathbf{x} \wedge \mathbf{y}) = F(\mathbf{x})F(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \{0, 1\}^k$. A modular function is an affine map (see for example [11, Proposition 24]), so a strictly positive log-modular function is a product of unary functions, so it has rank 1. The result is then Topkis's theorem [96] in the form stated in [22, Lemma 8].

4. If F is of the form $U_1(x_1) \cdots U_n(x_n)$ then

$$(T^{\otimes n}F)(x_1, \ldots, x_n) = (T^{\otimes 1}U_1)(x_1) \cdots (T^{\otimes 1}U_n)(x_n)$$

The reverse implication follows from $(T^{-1})^{\otimes n}T^{\otimes n}F = F$, where T^{-1} is the matrix inverse of T.

□

A function $F \colon D^n \to \mathbb{Q}_{\geq 0}$ is *essentially pseudo-Boolean* if its support is contained in a set $D_1 \times \cdots \times D_n$ with $|D_1|, \ldots, |D_n| \leq 2$. The *projection* of a relation $R \subseteq D^n$ onto indices $1 \leq i < j \leq n$ is the set of pairs $(a, b) \in D^2$ such that there exists $\mathbf{x} \in R$ with $x_i = a$ and $x_j = b$. A *generalised* NEQ is a relation of the form $\{(x_i, x_j), (y_i, y_j)\} \subset D^2$ for some $x_i \neq y_i$ and $x_j \neq y_j$.

Lemma 6.9. *Let $F \colon D^n \to \mathbb{Q}_{\geq 0}$ be an essentially pseudo-Boolean function which is not of rank 1, and assume that no binary projection of the support of F is a generalised* NEQ. *Then $\{F\} \cup \mathcal{U}_D$ is not weakly log-modular.*

Proof. Let the support of F be contained in $D_1 \times \cdots \times D_n$ where $|D_i| = 2$ for all i. Choose bijections $\rho_i \colon \{0, 1\} \to D_i$ for each $1 \leq i \leq n$. Define $F_\rho \colon \{0, 1\}^n \to \mathbb{Q}_{\geq 0}$ by

$$F_\rho(x_1, \ldots, x_n) = F(\rho_1(x_1), \ldots, \rho_n(x_n))$$

for all $x_1, \ldots, x_n \in \{0, 1\}$. Let $T = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ and note that $T^{\otimes n}F_\rho$ is strictly positive. Since F is not of rank 1, F_ρ is not of rank 1, so by Lemma 6.8 part (4), $T^{\otimes n}F_\rho$ is not of rank 1. By Lemma 6.8 part (3), there is a function $B \colon \{0, 1\}^2 \to \mathbb{Q}_{\geq 0}$ of the following form that is not of rank 1.

$$B(x_i, x_j) = (T^{\otimes n}F_\rho)(c_1, \ldots, c_{i-1}, x_i, c_{i+1}, \ldots, c_{j-1}, x_j, c_{j+1}, \ldots, c_n).$$

For all indices $k \in \{1, \ldots, n\} \setminus \{i, j\}$, define $U_k \in \mathcal{U}_D$ by $U_k(\rho_k(x_k)) = T_{c_k x_k}$ for all $x_k \in \{0, 1\}$, and $U_k(z) = 0$ if $z \notin D_k$. Define $G, H \colon D^2 \to \mathbb{Q}_{\geq 0}$ and $G_{\rho_i, \rho_j}, H_{\rho_i, \rho_i} \colon \{0, 1\}^2 \to \mathbb{Q}_{\geq 0}$ as follows. Note in these definitions that i and j are fixed, but ρ_i and ρ_j are used as subscripts in the name of some of the functions as a reminder of the bijections that are being applied to the inputs. Thus, in H_{ρ_i,ρ_i}, the bijection ρ_i is applied to both arguments, even though the function depends on both i and j.

$$\begin{aligned}
G(y_i, y_j) &= \sum \left(\prod_{k \neq i,j} U_k(y_k) \right) F(y_1, \ldots, y_n) && \text{for all } y_i, y_j \in D \\
G_{\rho_i,\rho_j}(x_i, x_j) &= \sum \left(\prod_{k \neq i,j} T_{c_k,x_k} \right) F_\rho(x_1, \ldots, x_n) && \text{for all } x_i, x_j \in \{0,1\} \\
H(y', y'') &= \sum_{y \in D} G(y', y) G(y'', y) && \text{for all } y', y'' \in D \\
H_{\rho_i,\rho_i}(x', x'') &= \sum_{x \in \{0,1\}} G_{\rho_i,\rho_j}(x', x) G_{\rho_i,\rho_j}(x'', x) && \text{for all } x', x'' \in \{0,1\}
\end{aligned}$$

where the first sum is over all $y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_{j-1}, y_{j+1}, \ldots, y_n \in D$ and the second sum is over all $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n \in \{0,1\}$.

Note that $M_{H_{\rho_i,\rho_i}} = M_{G_{\rho_i,\rho_j}} M^{\mathrm{t}}_{G_{\rho_i,\rho_j}} = T^{-1} M_B (T^{-1})^{\mathrm{t}} T^{-1} M_B^t (T^{-1})^{\mathrm{t}}$ where t denotes transpose. Taking determinants and applying Lemma 6.8 part (1) this implies that H_{ρ_i,ρ_i} is not of rank 1. Also, since T is strictly positive, the support of G_{ρ_i,ρ_j} is the binary projection of the support of F_ρ onto i and j which, by assumption, is not NEQ or EQ. Hence H_{ρ_i,ρ_i} is strictly positive but not of rank 1. Again using Lemma 6.8 part (1) we see that H is a witness that $\{F\} \cup \mathcal{U}_D$ is not weakly log-modular. □

Lemma 6.10. *Every conservative weakly log-modular weighted constraint language is balanced.*

Proof. Let $\mathcal{F}$ be a conservative weighted constraint language that is not balanced. We will show that $\mathcal{F}$ is not weakly log-modular.

By the definition of balance, there is a function $F \in \langle \mathcal{F} \rangle_\#$ of arity n and a partition $\mathbf{x} = (\mathbf{u}, \mathbf{v})$ of its n variables, such that the matrix $F(\mathbf{u}, \mathbf{v})$ is not of block-rank 1. By Lemma 6.8 part (2) there is a two-by-two submatrix $N = N_{F,\mathbf{u},\mathbf{u}',\mathbf{v},\mathbf{v}'}$ that is not of block-rank 1.

Construct an essentially pseudo-Boolean function G from F as follows. For all $1 \leq i \leq n$ let $U_i \in \langle \mathcal{U}_D \rangle_\# \subseteq \langle \mathcal{F} \rangle_\#$ be the indicator function of $D^{i-1} \times D_i \times D^{n-i}$, where $D_i = \{u_i, u_i'\}$ for all $1 \leq i \leq k$ and $D_i = \{v_i, v_i'\}$ for all $k < i \leq n$. Let $G = F \prod_{i=1}^n U_i$. Then $N_{G,\mathbf{u},\mathbf{u}',\mathbf{v},\mathbf{v}'} = N$ is not of block-rank 1, and G is essentially pseudo-Boolean.

If the binary projection of the support of G onto two indices i, j is a generalised NEQ $\{(x_i, x_j), (y_i, y_j)\}$, construct $(G', \rho(\mathbf{u}), \rho(\mathbf{u}'), \rho(\mathbf{v}), \rho(\mathbf{v}'))$ from $(G, \mathbf{u}, \mathbf{u}', \mathbf{v}, \mathbf{v}')$ as follows. Let $\rho \colon D^n \to D^{n-1}$ be the projection operator sending $\mathbf{x}$ to $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ and let $G'(\mathbf{x}) = \sum_{\rho(\mathbf{x}')=\mathbf{x}} G(\mathbf{x}')$ for all $\mathbf{x} \in D^{n-1}$. Note that, for all $\mathbf{x} \in D^n$, $G(\mathbf{x}) \neq G'(\rho(\mathbf{x}))$ implies that $G(\mathbf{x}) = 0$ because $G(\mathbf{x}) = 0$ unless $x_i \neq x_j$. Note that N has at least three non-zero entries by Lemma 6.8 part (1). So the corresponding three pairs out of $((\mathbf{u},\mathbf{v})_i, (\mathbf{u},\mathbf{v})_j)$, $((\mathbf{u},\mathbf{v}')_i, (\mathbf{u},\mathbf{v}')_j)$, $((\mathbf{u}',\mathbf{v})_i, (\mathbf{u}',\mathbf{v})_j)$, and $((\mathbf{u}',\mathbf{v}')_i, (\mathbf{u}',\mathbf{v}')_j)$ must each be either (x_i, x_j) or (y_i, y_j). But then the fourth pair must also be (x_i, x_j) or (y_i, y_j), which implies that $N_{G',\rho(\mathbf{u}),\rho(\mathbf{u}'),\rho(\mathbf{v}),\rho(\mathbf{v}')} = N$. Also, G' is essentially pseudo-Boolean, and G' is obtained by summing the i'th variable, so $G' \in \langle G \rangle_\#$.

Repeating this process if necessary, we obtain $(G', \mathbf{x}, \mathbf{x}', \mathbf{y}, \mathbf{y}')$ such that G' is an essentially pseudo-Boolean function in $\langle \mathcal{F}, \mathcal{U}_D \rangle_\# = \langle \mathcal{F} \rangle_\#$ and none of the binary projections of the support of G' is a generalised NEQ, and $N_{G',\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}$ is not of block-rank 1. So, in particular, G' is not of rank 1. By Lemma 6.9, $\{G'\} \cup \mathcal{U}_D$ is not weakly log-modular, so $\langle \mathcal{F} \rangle_\#$ is not weakly log-modular. □

We now return to Theorem 6.4 and prove the tractable case. The proof relies on an important theorem of Cai, Chen and Lu about the complexity of exact evaluation.

Lemma 6.11. *[24] Let $\mathcal{F}$ be a finite, weighted constraint language taking non-negative algebraic real values. If $\mathcal{F}$ is balanced, then $\#\mathsf{CSP}(\mathcal{F})$ is in* FP*, and otherwise $\#\mathsf{CSP}(\mathcal{F})$ is* #P*-hard.*

Theorem 6.12. *Let $\mathcal{F}$ be a conservative weighted constraint language taking values in $\mathbb{Q}_{\geq 0}$. If $\mathcal{F}$ is weakly log-modular then, for any finite $\mathcal{G} \subset \mathcal{F}$, $\#\mathsf{CSP}(\mathcal{G}) \in$ FP.*

Proof. By Lemma 6.10, $\mathcal{F}$ is balanced. Hence, every finite $\mathcal{G} \subset \mathcal{F}$ is balanced, which implies that $\#\mathsf{CSP}(\mathcal{G})$ is in FP by Lemma 6.11. □

6.4 Valued clones, valued CSPs and relational clones

To define valued clones, we use the same set-up as Section 6.1.3 except that summation is replaced by minimisation and product is replaced by sum. Let D be a finite domain with $|D| \geq 2$ and let R be a codomain with $\{0, \infty\} \subseteq R$, where ∞ obeys the following rules for all $x \in R$: $x + \infty = \infty$, $x \leq \infty$ and $\min\{x, \infty\} = x$. Let Φ be a subset of $\mathrm{Func}(D, R)$ and let $V = \{v_1, \ldots, v_n\}$ be a set of variables. For each atomic formula $\varphi = G(v_{i_1}, \ldots, v_{i_a})$ we use the notation f_φ to denote the function represented by φ, so $f_\varphi(\mathbf{x}) = G(x_{i_1}, \ldots, x_{i_a})$.

A psm-formula ("primitive sum minimisation formula") is a minimisation of a sum of atomic formulas. A psm-formula ψ over Φ in variables $V' = \{v_1, \ldots, v_{n+k}\}$ has the form

$$\psi = \min_{v_{n+1},\ldots,v_{n+k}} \sum_{j=1}^{m} \varphi_j, \tag{6.3}$$

where φ_j are all atomic formulas over Φ in the variables V'. The formula ψ specifies a function $f_\psi \colon D^n \to R$ in the following way:

$$f_\psi(\mathbf{x}) = \min_{\mathbf{y} \in D^k} \sum_{j=1}^{m} f_{\varphi_j}(\mathbf{x}, \mathbf{y}), \tag{6.4}$$

where $\mathbf{x}$ and $\mathbf{y}$ are assignments $\mathbf{x} \colon \{v_1, \ldots, v_n\} \to D$ and $\mathbf{y} \colon \{v_{n+1}, \ldots, v_{n+k}\} \to D$.

The *valued clone* $\langle\Phi\rangle_V$ *generated by* Φ is the set of all functions that can be represented by a psm-formula over $\Phi \cup \{\mathrm{eq}\}$, where eq is the binary equality function on D given by $\mathrm{eq}(x, x) = 0$ and $\mathrm{eq}(x, y) = \infty$ for $x \neq y$.

We next introduce valued constraint satisfaction problems (VCSPs), which are optimisation problems. In the work of Kolmogorov and Živný [77], the codomain is $R = \mathbb{Q}_{\geq 0} \cup \{\infty\}$. For reasons which will be clear below, it is useful for us to extend the codomain to include irrational numbers. This will not cause problems because, with the exception of Theorem 6.33 we use only formal calculations from their papers, not complexity results. For Theorem 6.33, we avoid irrational numbers and, in fact, restrict to cost functions taking values in $\{0, \infty\} \subset R$. Furthermore, all the real numbers we use are either rationals or their logarithms so are efficiently computable.

Let $\overline{\mathbb{R}}_{\geq 0} = \mathbb{R}_{\geq 0} \cup \{\infty\}$ be the set of non-negative real numbers together with ∞.

Definition 6.13. A *cost function* is a function $D^k \to \overline{\mathbb{R}}_{\geq 0}$. A *valued constraint language* is a set of cost functions $\Phi \subseteq \mathrm{Func}(D, \overline{\mathbb{R}}_{\geq 0})$.

Given a valued constraint language Φ, VCSP(Φ) is the problem of taking an instance ψ, a psm-formula consisting of a sum of m atomic Φ-formulas over n free variables $\mathbf{x}$ and computing the value

$$\mathrm{minCost}(\psi) = \min_{\mathbf{x} \in D^n} f_\psi(\mathbf{x})\,,$$

where f_ψ is the function defined by ψ.

We typically use the notation of Kolmogorov and Živný. An instance is usually denoted by the letter I. In this case, we use f_I to denote the function specified by the psm-formula corresponding to instance I, so the value of the instance is denoted by $\mathrm{minCost}(I)$. The psm-formula corresponding to I is a sum of atomic formulas (since all of the variables are free variables). We refer to each of these atomic formulas as a *valued constraint* and we represent these by the multiset T of all valued constraints in the instance I. For each valued constraint $t \in T$ we use k_t to denote its arity, f_t to denote the function represented by the corresponding atomic formula, and σ_t to denote its scope, which is given as a tuple $(i(t, 1), \ldots, i(t, k_t)) \in \{1, \ldots, n\}^{k_t}$ containing the indices of the variables in the scope. Thus,

$$f_I(\mathbf{x}) = \sum_{t \in T} f_t(x_{i(t,1)}, \ldots, x_{i(t,k_t)})\,. \tag{6.5}$$

For convenience, we use $\mathbf{x}[\sigma_t]$ as an abbreviation for the tuple $(x_{i(t,1)}, \ldots, x_{i(t,k_t)})$. In this abbreviated notation, the function defined by instance I may be written $f_I(\mathbf{x}) = \sum_{t \in T} f_t(\mathbf{x}[\sigma_t])$.

Now, let $[0, 1]_{\mathbb{Q}} = [0, 1] \cap \mathbb{Q}$. For reasons which will be clear below, it will be useful to work with weight functions in $\mathrm{Func}(D, [0, 1]_{\mathbb{Q}})$. For such a weight function F, let the cost function $\ell(F) \in \mathrm{Func}(D, \overline{\mathbb{R}}_{\geq 0})$ be the function defined by

$$(\ell(F))(\mathbf{x}) = \begin{cases} -\ln F(\mathbf{x}) & \text{if } F(\mathbf{x}) > 0 \\ \infty & \text{if } F(\mathbf{x}) = 0. \end{cases}$$

For example, $\ell(\mathrm{EQ}) = \mathrm{eq}$. For a weighted constraint language $\mathcal{F} \subseteq \mathrm{Func}(D, [0,1]_{\mathbb{Q}})$, let $\ell(\mathcal{F})$ be the valued constraint language defined by $\ell(\mathcal{F}) = \{\ell(F) \mid F \in \mathcal{F}\}$.

There is a natural bijection between instances of $\#\mathsf{CSP}(\mathcal{F})$ and $\mathrm{VCSP}(\ell(\mathcal{F}))$, obtained by replacing each function F_t in the former by the function $f_t = \ell(F_t)$ in the latter, keeping the scopes unchanged. Note that $f_I(\mathbf{x}) = -\ln F_I(\mathbf{x})$, for any assignment $\mathbf{x}$, with the convention $-\ln 0 = \infty$.

Definition 6.14. A valued constraint language is *conservative* if it contains all arity-1 cost functions $D \to \overline{\mathbb{R}}_{\geq 0}$.

The mapping $F \mapsto \ell(F)$ from $\mathrm{Func}(D, [0,1]_{\mathbb{Q}})$ to $\mathrm{Func}(D, \overline{\mathbb{R}}_{\geq 0})$ is not surjective because there are real numbers that are not the logarithm of any rational. For the same reason, the valued constraint language $\ell(\mathcal{F})$ is not conservative (for any weighted constraint language $\mathcal{F}$). Finally, note that we have only defined $\ell(F)$ for $F \in \mathrm{Func}(D, [0,1]_{\mathbb{Q}})$. The obvious extension to $F \in \mathrm{Func}(D, \mathbb{Q}_{\geq 0})$ would produce negative-valued cost functions and we wish to avoid this since Kolmogorov and Živný [77] do not allow it.

Definition 6.15. A cost function is *crisp* [36] if $f(\mathbf{x}) \in \{0, \infty\}$ for all $\mathbf{x}$.

Definition 6.16. For any cost function f, let $\mathrm{Feas}(f)$ be the relation defined by $\mathrm{Feas}(f) = \{\mathbf{x} \mid f(\mathbf{x}) < \infty\}$.

Thus, any cost function f can be associated with its underlying relation. Similarly, we can represent any relation by a crisp cost function f for which $f(\mathbf{x}) = 0$ if and only if $\mathbf{x}$ is in the relation. A *crisp constraint language* is a set of relations, which we always represent as crisp cost functions, not as functions with codomain $\{0, 1\}$. For a valued constraint language Φ, the crisp constraint language $\mathrm{Feas}(\Phi)$ is given by $\mathrm{Feas}(\Phi) = \{\mathrm{Feas}(f) \mid f \in \Phi\}$.

Definition 6.17. A crisp constraint language is *conservative* if it includes all arity-1 relations.

A *relational clone* is simply a crisp constraint language $\mathrm{Feas}(\langle\Phi\rangle_V)$ for a valued constraint language Φ.

Lemma 6.18. *Suppose* $\Phi \subseteq \mathrm{Func}(D, \overline{\mathbb{R}}_{\geq 0})$. *Then* $\langle \mathrm{Feas}(\Phi)\rangle_V = \mathrm{Feas}(\langle\Phi\rangle_V)$.

Proof. The mapping $\rho\colon \overline{\mathbb{R}}_{\geq 0} \to \{0, \infty\}$ defined by $\rho(\infty) = \infty$ and $\rho(x) = 0$, for all $x < \infty$, is a homomorphism of semirings, from $(\overline{\mathbb{R}}_{\geq 0}, \min, +)$ to $(\{0, \infty\}, \min, +)$. $\square$

6.5 STP/MJN multimorphisms and weak log-supermodularity

In [77, Corollary 12], Kolmogorov and Živný give a tractability criterion for conservative VCSPs. In particular, they show that the VCSP associated with a conservative valued constraint language Φ is tractable iff Φ has an STP/MJN multimorphism.

We define STP/MJN multimorphisms below. In this section, we show (see Theorem 6.30 below) that, if a weighted constraint language $\mathcal{F} \in \mathrm{Func}(D, [0,1]_{\mathbb{Q}})$ is weakly log-supermodular, then the corresponding valued constraint language $\ell(\mathcal{F})$ has an STP/MJN multimorphism. In Section 6.6, this will enable us to use such a multimorphism (via the work of Kolmogorov and Živný[77] and Cohen, Cooper and Jeavons [35]) to prove #BIS-easiness and LSM-easiness of the weighted counting CSP.

Our proof of Theorem 6.30 relies on work by Kolmogorov and Živný [77] and Takhanov [94]. We start with some general definitions. Most of these are from [77], but some care is required since some of the definitions in [77] differ from those in [35].

Definition 6.19. A *k-ary operation on D* is a function from D^k to D. An *operation on D* is a k-ary operation, for some k.

We drop the “on D” when the domain D is clear from the context.

Definition 6.20. A k-tuple $\langle \rho_1, \ldots, \rho_k \rangle$ of k-ary operations $\rho_1, \ldots, \rho_k$ is *conservative* if, for every tuple $\mathbf{x} = (x_1, \ldots, x_k) \in D^k$, the multisets $\{\{x_1, \ldots, x_k\}\}$ and $\{\{\rho_1(\mathbf{x}), \ldots, \rho_k(\mathbf{x})\}\}$ are equal.

Note that we have now defined conservative operations and conservative constraint languages (weighted, valued and crisp). There are connections between these notions of “conservative” but we do not need these here.

Definition 6.21. $\langle \rho_1, \ldots, \rho_k \rangle$ is a *multimorphism* of an arity-r cost function f if, for all $\mathbf{x}^1, \ldots, \mathbf{x}^k \in D^r$, we have:

$$\sum_{i=1}^{k} f(\rho_i(x_1^1, \ldots, x_1^k), \ldots, \rho_i(x_r^1, \ldots, x_r^k)) \leq \sum_{i=1}^{k} f(\mathbf{x}^i).$$

Definition 6.22. $\langle \rho_1, \ldots, \rho_k \rangle$ is a multimorphism of a valued constraint language Φ if it is a multimorphism of every $f \in \Phi$.

These definitions imply the following.

Observation 1. If $\langle \rho_1, \ldots, \rho_k \rangle$ is conservative, then it is a multimorphism of every unary cost function f.

Definition 6.23. Suppose $M \subseteq D^2$. A pair $\langle \sqcap, \sqcup \rangle$ of binary operations is a *symmetric tournament pair* (STP) on M if it is conservative and both operations are commutative on M. We say that it is an STP if it is an STP on D^2.

Definition 6.24. Suppose $M \subseteq D^2$. A triple $\langle \texttt{Mj1}, \texttt{Mj2}, \texttt{Mn3} \rangle$ of ternary operations is an *MJN on M* if it is conservative and, for all triples $(a, b, c) \in D^3$ with $\{\{a, b, c\}\} = \{\{x, x, y\}\}$ where x and y are distinct and $(x, y) \in M$, we have $\texttt{Mj1}(a, b, c) = \texttt{Mj2}(a, b, c) = x$ and $\texttt{Mn3}(a, b, c) = y$.

Definition 6.25. An *STP/MJN multimorphism* of a valued constraint language Φ consists of a pair of operations $\langle \sqcap, \sqcup \rangle$ and a triple of operations $\langle \texttt{Mj1}, \texttt{Mj2}, \texttt{Mn3} \rangle$, both of which are multimorphisms of Φ, for which, for some symmetric subset M of D^2, $\langle \sqcap, \sqcup \rangle$ is an STP on M and $\langle \texttt{Mj1}, \texttt{Mj2}, \texttt{Mn3} \rangle$ is an MJN on $\{(a, b) \in D^2 \setminus M \mid a \neq b\}$.

Definition 6.26. $\Phi \subseteq \mathrm{Func}(D, \overline{\mathbb{R}}_{\geq 0})$ is *weakly submodular* if, for all binary functions $f \in \langle \Phi \rangle_V$ and elements $a, b \in D$,

$$f(a, a) + f(b, b) \leq f(a, b) + f(b, a) \quad \text{or} \quad f(a, a) = f(b, b) = \infty. \tag{6.6}$$

Note that the definition of weak submodularity for cost functions is a restatement of Kolmogorov and Živný's "Assumption 3". It is not trivial that weak log-supermodularity for $\mathcal{F}$ is related to weak submodularity for $\ell(\mathcal{F})$. Expressibility for VCSP is different from expressibility for #CSP and, specifically, we cannot expect $\langle \ell(\mathcal{F}) \rangle_V = \ell(\langle \mathcal{F} \rangle_\#)$ to hold in general. However, the following is suitable for our purposes.

Lemma 6.27. *Suppose* $\mathcal{F} \subseteq \mathrm{Func}(D, [0, 1]_\mathbb{Q})$ *and let* $\Phi = \ell(\mathcal{F})$. *If* $\mathcal{F}$ *is weakly log-supermodular then* Φ *is weakly submodular.*

Proof. We prove the contrapositive. Suppose $f \in \langle \Phi \rangle_V$ is a binary function that witnesses the fact that Φ is not weakly submodular according to Definition 6.26, specifically,

$$f(a, a) + f(b, b) > f(a, b) + f(b, a) \quad \text{and} \quad \min\{f(a, a), f(b, b)\} < \infty.$$

Since $f \in \langle \Phi \rangle_V$, we may express f in the form

$$f(\mathbf{x}) = \min_{\mathbf{y}} g(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{y}} \sum_{i=1}^{m} g_i(\mathbf{x}, \mathbf{y}),$$

where the $g_i \in \Phi$ are atomic. For $k \in \mathbb{N}$, define

$$F^{(k)}(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{i=1}^{m} G_i(\mathbf{x}, \mathbf{y})^k,$$

where each G_i is such that $g_i = \ell(G_i)$. Note that $F^{(k)} \in \langle \mathcal{F} \rangle_\#$, and

$$F^{(k)}(\mathbf{x})^{1/k} \to \max_{\mathbf{y}} \prod_{i=1}^{m} G_i(\mathbf{x}, \mathbf{y}), \quad \text{as } k \to \infty.$$

Now

$$\max_{\mathbf{y}} \prod_{i=1}^{m} G_i(\mathbf{x}, \mathbf{y}) = \max_{\mathbf{y}} \exp\Big(- \sum_{i=1}^{m} g_i(\mathbf{x}, \mathbf{y}) \Big)$$
$$= \exp\Big(- \min_{\mathbf{y}} \sum_{i=1}^{m} g_i(\mathbf{x}, \mathbf{y}) \Big)$$
$$= \exp(-f(\mathbf{x}))$$

and

$$\exp(-f(a,a)) \exp(-f(b,b)) < \exp(-f(a,b)) \exp(-f(b,a)).$$

Thus $F(a,a)F(b,b) < F(a,b)F(b,a)$ where $F = F^{(k)}$ for some sufficiently large k. Also, $\min\{f(a,a), f(b,b)\} < \infty$ implies that $\max\{F(a,a), F(b,b)\} > 0$. These properties of F imply that $\mathcal{F}$ is not weakly log-supermodular, according to Definition 6.3. $\square$

Let Γ be a crisp constraint language. A *majority polymorphism* of Γ is a ternary operation ρ such that $\rho(a,a,b) = \rho(a,b,a) = \rho(b,a,a) = a$ for all $a, b \in D$ and for all arity-k relations $R \in \Gamma$ we have

$$\mathbf{x}, \mathbf{y}, \mathbf{z} \in R \implies (\rho(x_1, y_1, z_1), \dots, \rho(x_k, y_k, z_k)) \in R.$$

Let $N(a,b,c,d)$ be the relation $\{(a,c), (b,c), (a,d)\}$. The existence of such a relation in $\langle\Gamma\rangle_V$ indicates that Γ is not "strongly balanced" in the terminology of [50]. Note that, on the Boolean domain, $N(0,1,0,1)$ is NAND.

Theorem 6.28. *(Takhanov) Let Γ be a conservative relational clone with domain D. At least one of the following holds.*

- *There are distinct $a, b \in D$ such that $N(a,b,a,b) \in \Gamma$.*
- *There are distinct $a, b \in D$ such that $\{(a,a,a), (a,b,b), (b,a,b), (b,b,a)\} \in \Gamma$.*
- *For some $k \geq 1$, there are $a_0, \dots, a_{2k}, b_0, \dots, b_{2k} \in D$ such that, for each $0 \leq i \leq 2k$, $a_i \neq b_i$ and, for each $0 \leq i \leq 2k-1$,*

$$N(a_i, b_i, a_{i+1}, b_{i+1}) \in \Gamma \text{ and } N(a_{2k}, b_{2k}, a_0, b_0) \in \Gamma.$$

- *Γ has a majority polymorphism.*

Proof. This formulation is essentially [94, Theorem 9.1] except for the last bullet point. As stated in the proof of that theorem, the first two conditions both fail if and only if the "necessary local conditions" [94, Definition 3.5] hold. Unfortunately for us, Takhanov uses the term "functional clone" differently to how we use it, so the reader will need to take this into account to understand the local conditions. However, we do not need the detail, here. Takhanov's proof of the NP-hard case of his Theorem 3.7 (at the end of his Section 4) shows the following: Given the necessary local conditions, the third condition fails only if a certain graph T_F is bipartite. If T_F is bipartite then [94, Theorem 5.5] establishes a majority polymorphism. □

Lemma 6.29. *If* $\Phi \subseteq \mathrm{Func}(D, \overline{\mathbb{R}}_{\geq 0})$ *is conservative and weakly submodular,* $\Gamma = \langle \mathrm{Feas}(\Phi) \rangle_V$ *has a majority polymorphism.*

Proof. Since Φ is conservative (Definition 6.14), so is Γ (Definition 6.17). We will now show that the first three bullets of Theorem 6.28 contradict the premise of the lemma, so the fourth must hold.

The first bullet-point is easily ruled out. Suppose the given relation is in Γ. By Lemma 6.18, there is a binary function $f \in \langle \Phi \rangle_V$ such that $\mathrm{Feas}(f) = N(a, b, a, b)$. This function has $f(b, b) = \infty$ and $f(a, a), f(a, b), f(b, a) < \infty$, and hence violates (6.6).

For the second bullet-point, by Lemma 6.18 we have a 3-place function $g \in \langle \Phi \rangle_V$ which is finite precisely on $\{(a, a, a), (a, b, b), (b, a, b), (b, b, a)\}$. Now let M be a sufficiently large constant and let u be the unary function defined by

$$u(z) = \begin{cases} M & \text{if } z = a, \\ 0 & \text{if } z = b, \\ \infty & \text{otherwise.} \end{cases}$$

Let

$$f(x, y) = \min_{z \in D} \{g(x, y, z) + u(z)\}.$$

Then $f(a, a) = M + g(a, a, a)$, $f(b, b) = M + g(b, b, a)$, $f(a, b) = g(a, b, b)$ and $f(b, a) = g(b, a, b)$. Clearly, f violates (6.6) for sufficiently large M.

Finally, let us consider the third bullet-point. By Lemma 6.18 we have binary functions $g_0, g_1, \ldots, g_2k \in \langle \Phi \rangle_V$ where the underlying relation of g_i is $N(a_i, b_i, a_{i+1}, b_{i+1})$ for $0 \leq i < 2k$ and the underlying relation of g_{2k} is $N(a_{2k}, b_{2k}, a_0, b_0)$. Define

$$f(x, y) = \min \big\{ g_0(x, z_0) + u_0(z_0) + g_1(z_0, z_1) + u_1(z_1) + \\ \cdots + u_{2k-1}(z_{2k-1}) + g_{2k}(z_{2k-1}, y) \mid (z_0, \ldots, z_{2k-1}) \in D^{2k} \big\},$$

where $u_i(a_i) = M$, $u_i(b_i) = 0$, and $u_i(z) = \infty$ if $z \notin \{a_i, b_i\}$. Note that $f(a_0, a_0) \geq kM$, $f(b_0, b_0) \geq (k + 1)M$ and $f(a_0, b_0), f(b_0, a_0) \leq kM + (2k + 1)m$, where m is the largest finite value taken by any of $g_0, \ldots, g_{2k}$. So f violates (6.6) for sufficiently large M.

So we are left with the remaining possibility that Γ has a majority polymorphism. □

We can now prove the main result of this section.

Theorem 6.30. *Let* $\mathcal{F}$ *be a weighted constraint language such that* $\mathrm{Func}_1(D, [0, 1]_{\mathbb{Q}}) \subseteq \mathcal{F} \subseteq \mathrm{Func}(D, [0, 1]_{\mathbb{Q}})$*. If* $\mathcal{F}$ *is weakly log-supermodular then* $\ell(\mathcal{F})$ *has an STP/MJN multimorphism.*

Proof. Let $\Phi = \ell(\mathcal{F}) \cup \mathrm{Func}_1(D, \overline{\mathbb{R}}_{\geq 0})$. We will show that Φ has an STP/MJN multimorphism. By Definitions 6.25 and 6.22, this is also an STP/MJN multimorphism of the subset $\ell(\mathcal{F})$.

By Lemma 6.27, $\ell(\mathcal{F})$ is weakly submodular. Now, $\ell(\mathcal{F})$ contains $\ell(\mathrm{Func}_1(D, [0,1]_{\mathbb{Q}}))$. Thus, for every unary function $u \in \mathrm{Func}_1(D, \overline{\mathbb{R}}_{\geq 0})$ and every $\varepsilon \in (0,1)$, there is a unary function $u_\varepsilon \in \ell(\mathcal{F})$ such that, for all $x \in \{0,1\}$, $|u(x) - u_\varepsilon(x)| < \varepsilon$. From the definition of valued clones, and continuity, we deduce that, for every binary function $f \in \langle \Phi \rangle_V$ and every $\varepsilon > (0,1)$, there is an $f_\varepsilon \in \langle \ell(\mathcal{F}) \rangle_V$ such that, for all $x, y \in \{0,1\}$, $|f(x,y) - f_\varepsilon(x,y)| < \varepsilon$. Since $\ell(\mathcal{F})$ is weakly submodular, we conclude from the definition of weak submodularity (Definition 6.26) that Φ is weakly submodular.

In [77, §6.1–6.4], Kolmogorov and Živný show how to construct an STP/MJN multimorphism of Φ under "Assumptions 1–3". Assumption 1 is that Φ is conservative, which is true by construction. Assumption 3 is that Φ is weakly submodular. This is given as a premise of our lemma. Assumption 2 is that $\Gamma = \mathrm{Feas}(\Phi)$ has a majority polymorphism, which follows from Assumptions 1 and 3 by Lemma 6.29. (Assumption 2 states that Φ has a majority polymorphism. In our terminology, this means that $\mathrm{Feas}(\Phi)$ has a majority polymorphism.) $\square$

6.6 LSM-easiness and #BIS-easiness

Our aim is to show that if $\ell(\mathcal{F})$ has an STP/MJN multimorphism then $\mathcal{F}$ is LSM-easy. This will involve using the arguments of [35] and [77], but we try, as much as possible, to avoid going into the details of their proofs. We start by generalising the notion of an STP multimorphism.

Definition 6.31. Let f be an arity-k cost function. A *generalised STP multimorphism of* f is a pair $\langle \sqcap, \sqcup \rangle$, defined as follows. For $1 \leq i \leq k$, $\sqcap_i$ and $\sqcup_i$ are operations on the set $D_i = \{a \in D \mid \exists \mathbf{x} : x_i = a \text{ and } f(\mathbf{x}) < \infty\}$, and $\langle \sqcap_i, \sqcup_i \rangle$ is an STP of $\{f\}$.

The operation $\sqcap$ is the binary operation on $D_1 \times \cdots \times D_k$ defined by applying $\sqcap_1, \ldots, \sqcap_k$ component-wise. Similarly, $\sqcup$ is defined by applying $\sqcup_1, \ldots, \sqcup_k$ component-wise. We require that, for all $\mathbf{x}, \mathbf{y} \in D^k$, $f(\sqcup(\mathbf{x}, \mathbf{y})) + f(\sqcap(\mathbf{x}, \mathbf{y})) \leq f(\mathbf{x}) + f(\mathbf{y})$. Equivalently, we require

$$f(\sqcup_1(x_1, y_1), \ldots, \sqcup_k(x_k, y_k)) + f(\sqcap_1(x_1, y_1), \ldots, \sqcap_k(x_k, y_k)) \leq f(\mathbf{x}) + f(\mathbf{y}).$$

Where it is clearer, we use infix notation for operations such as $\sqcap$ and $\sqcup$.

Theorem 6.32 (Kolmogorov and Živný)**.** *Suppose* Φ_0 *is a finite, valued constraint language which has an STP/MJN multimorphism. Then there is a polynomial-time algorithm that takes an instance* I *of* $\mathrm{VCSP}(\Phi_0)$ *and returns a generalised STP multimorphism* $\langle \sqcap, \sqcup \rangle$ *of* f_I*. The pair* $\langle \sqcap, \sqcup \rangle$ *depends only on the STP/MJN multimorphism of* Φ_0 *and on the relation* $\mathrm{Feas}(f_I)$ *underlying* f_I*. It does not depend in any other way on* I*.*

Proof. This theorem is closely related to Theorem 11 in [77]. To prove Theorem 6.32, we just take the proof of Kolmogorov and Živný's Theorem 11 (refer to [77, §7]), but we stop before the final step. Specifically, we stop when their Lemma 35 has been proved. At this point, the existence of the pair $\langle \sqcap, \sqcup \rangle$, as specified above, has been established.

Note that Kolmogorov and Živný restrict to rationals, whereas we allow real numbers. This is not a problem because their algorithm does not require access to the functions in Φ_0 themselves. Instead, it only requires access to the relations in $\mathrm{Feas}(\Phi_0)$ and to the STP/MJN multimorphism that Φ_0 satisfies. These are both finite amounts of data, which can be hardwired into the algorithm, whose input is just the instance I, which is a symbolic expression. $\square$

We will also use the following algorithmic consequence of [77, Theorem 11]. We restrict to crisp cost functions because this is all that we use and we wish to avoid issues with number systems.

Theorem 6.33 (Kolmogorov and Živný)**.** *Suppose that Φ_0 is a finite, crisp constraint language that has an STP/MJN multimorphism. Then there is a polynomial-time algorithm for* $\mathrm{VCSP}(\Phi_0)$.

For our eventual construction, we would like $\langle \sqcap, \sqcup \rangle$ to induce a generalised STP multimorphism of f_t for each individual valued constraint t in the instance. We do not know whether this is true of the generalised STP multimorphism provided by Kolmogorov and Živný's algorithm, but something sufficiently close to this is true.

Definition 6.34. For an instance I, a valued constraint t and a length-k_t vector $\mathbf{a}$, define

$$R_{I,t}(\mathbf{a}) = \begin{cases} 0, & \text{if there exists } \mathbf{x} \text{ with } \mathbf{x}[\sigma_t] = \mathbf{a} \text{ and } f_I(\mathbf{x}) < \infty; \\ \infty, & \text{otherwise,} \end{cases}$$

and define $f'_t = f_t + R_{I,t}$.

Thus, f'_t is a "trimmed" version of f_t whose domain is precisely the k_t-tuples of values that can actually arise in feasible solutions to instance I. We will see that if the scope σ_t contains variables with indices $i(t,1), \ldots, i(t,k_t)$, then

$$\big\langle \sqcap[\sigma_t], \sqcup[\sigma_t] \big\rangle = \big\langle (\sqcap_{i(t,1)}, \ldots, \sqcap_{i(t,k_t)}), (\sqcup_{i(t,1)}, \ldots, \sqcup_{i(t,k_t)}) \big\rangle$$

is a generalised STP multimorphism of f'_t, even though it might not necessarily be a generalised STP multimorphism of f_t.

Note that Theorem 6.33 has the following consequence.

Corollary 6.35. *Let Φ_0 be a finite, valued constraint language that has an STP/MJN multimorphism. There is a polynomial-time algorithm that takes an instance I of* $\mathrm{VCSP}(\Phi_0)$, *a valued constraint t and a length-k_t vector* $\mathbf{a}$ *and returns a truth table for f'_t.*

The truth table produced by the algorithm of Corollary 6.35 is finite since all valued constraints in Φ_0 are finite.

Theorem 6.36 (An extension to Theorem 6.32). *Suppose Φ_0 is a finite, valued constraint language which has an STP/MJN multimorphism. Consider the algorithm from Theorem 6.32 which takes an instance I of* VCSP(Φ_0) *(in the form (6.5)) and returns a generalised STP multimorphism $\langle \sqcap, \sqcup \rangle$ of f_I. Then, for all $t \in T$, $\big\langle \sqcap[\sigma_t], \sqcup[\sigma_t] \big\rangle$ is a generalised STP multimorphism of f'_t.*

Proof. Focus on a particular valued constraint t of I. Let $k = k_t$ be the arity of f_t, and for brevity denote $\sqcap[\sigma_t]$ and $\sqcup[\sigma_t]$ by $\sqcap'$ and $\sqcup'$, respectively. Without loss of generality assume $\sigma_t = (1, 2, \ldots, k)$. We wish to show that

$$f'_t(\mathbf{a} \sqcap' \mathbf{b}) + f'_t(\mathbf{a} \sqcup' \mathbf{b}) \leq f'_t(\mathbf{a}) + f'_t(\mathbf{b}) \tag{6.7}$$

for all $\mathbf{a}, \mathbf{b} \in D_1 \times \cdots \times D_k$. If either $f'_t(\mathbf{a}) = \infty$ or $f'_t(\mathbf{b}) = \infty$, then we are done. Otherwise, by construction of f'_t, there exist $\mathbf{x}$ and $\mathbf{y}$ such that $\mathbf{a} = \mathbf{x}[\sigma_t]$, $\mathbf{b} = \mathbf{y}[\sigma_t]$, and $f_I(\mathbf{x}), f_I(\mathbf{y}) < \infty$. Notice that $f'_t(\mathbf{a}) = f_t(\mathbf{a}) < \infty$ and $f'_t(\mathbf{b}) = f_t(\mathbf{b}) < \infty$, also by construction of f'_t. Now consider the augmented instance I_N of I with N extra copies of the valued constraint t. We have

$$\begin{aligned} f_{I_N}(\mathbf{x}) &= f_I(\mathbf{x}) + N f_t(\mathbf{a}) \\ f_{I_N}(\mathbf{y}) &= f_I(\mathbf{y}) + N f_t(\mathbf{b}) \\ f_{I_N}(\mathbf{x} \sqcap \mathbf{y}) &= f_I(\mathbf{x} \sqcap \mathbf{y}) + N f_t(\mathbf{a} \sqcap' \mathbf{b}) \\ f_{I_N}(\mathbf{x} \sqcup \mathbf{y}) &= f_I(\mathbf{x} \sqcup \mathbf{y}) + N f_t(\mathbf{a} \sqcup' \mathbf{b}). \end{aligned} \tag{6.8}$$

Since $\mathrm{Feas}(f_{I_N}) = \mathrm{Feas}(f_I)$, from Theorem 6.32, $\langle \sqcap, \sqcup \rangle$ is also a generalised STP multimorphism of f_{I_N}, i.e.,

$$f_{I_N}(\mathbf{x} \sqcap \mathbf{y}) + f_{I_N}(\mathbf{x} \sqcup \mathbf{y}) \leq f_{I_N}(\mathbf{x}) + f_{I_N}(\mathbf{y}).$$

Combining this with (6.8), we obtain

$$f_t(\mathbf{a} \sqcap' \mathbf{b}) + f_t(\mathbf{a} \sqcup' \mathbf{b}) + O(N^{-1}) \leq f_t(\mathbf{a}) + f_t(\mathbf{b}) + O(N^{-1}). \tag{6.9}$$

Since (6.9) remains true as $N \to \infty$ but f_t is independent of N, we conclude that

$$f_t(\mathbf{a} \sqcap' \mathbf{b}) + f_t(\mathbf{a} \sqcup' \mathbf{b}) \leq f_t(\mathbf{a}) + f_t(\mathbf{b}).$$

Since $\mathbf{a} \sqcap' \mathbf{b}$ and $\mathbf{a} \sqcup' \mathbf{b}$ extend to feasible solutions $\mathbf{x} \sqcap \mathbf{y}$ and $\mathbf{x} \sqcup \mathbf{y}$, it follows that $f'_t(\mathbf{a} \sqcap' \mathbf{b}) = f_t(\mathbf{a} \sqcap' \mathbf{b})$ and $f'_t(\mathbf{a} \sqcup' \mathbf{b}) = f_t(\mathbf{a} \sqcup' \mathbf{b})$. The required inequality (6.7) follows immediately. ∎

To make use of Theorem 6.36, we will use the following definitions.

Definition 6.37. Given a finite, valued constraint language $\Phi_0 \subset \mathrm{Func}(D, \overline{\mathbb{R}}_{\geq 0})$, let Φ_0' be the set of functions of the form $f + R$, for $f \in \Phi_0 \cap \mathrm{Func}_k(D, \overline{\mathbb{R}}_{\geq 0})$, $R \in \mathrm{Func}_k(D, \{0, \infty\})$ and $k \in \mathbb{N}$.

Note that Φ_0' is finite because $\mathrm{Func}_k(D, \{0, \infty\})$ is finite for any finite k.

Definition 6.38. Two n-variable instances I and I' of $\mathrm{VCSP}(\Phi)$ are *equivalent* if $f_I(\mathbf{x}) = f_{I'}(\mathbf{x})$ for all $\mathbf{x} \in D^n$.

Definition 6.39. [35] A function $f \colon D_1 \times \cdots \times D_r \to \overline{\mathbb{R}}_{\geq 0}$ is *domain-reduced* if, for each $i \in \{1, \ldots, r\}$, and for each $a \in D_i$, there is an $\mathbf{x} \in D^n$ such that $x_i = a$ and $f(\mathbf{x}) < \infty$.

Lemma 6.40. *Suppose Φ_0 is a finite, valued constraint language which has an STP/MJN multimorphism. Consider an instance I of* $\mathrm{VCSP}(\Phi_0)$. *There is an equivalent instance I' of* $\mathrm{VCSP}(\Phi_0')$ *and a generalised STP multimorphism $\langle \sqcap, \sqcup \rangle$ of $f_{I'}$ which induces a generalised STP multimorphism of f_t for each valued constraint t of I'. Both I' and $\langle \sqcap, \sqcup \rangle$ are polynomial-time computable (given I). Moreover, each operation $\sqcap_i$ and $\sqcup_i$ induces a total order.*

Proof. We first show how to construct an equivalent instance I' and a generalised STP multimorphism of $f_{I'}$ which induces generalised STP multimorphisms on the valued constraints. To obtain I', start from the instance I and use Corollary 6.35 to replace each valued constraint $f_t(\mathbf{x}[\sigma_t])$ with $f_t'(\mathbf{x}[\sigma_t])$. This operation clearly preserves the set of feasible solutions and their costs. Then use the algorithm from Theorem 6.36 to construct the generalised STP multimorphism $\langle \sqcap, \sqcup \rangle$.

In the remainder of the proof, we construct a new generalised STP multimorphism by modifying $\langle \sqcap, \sqcup \rangle$ to ensure that it is composed of total orders, as required. Consider the following claim.

Claim: Suppose that D is a domain. Given a set of functions $\Phi \subseteq \mathrm{Func}(D, \overline{\mathbb{R}}_{\geq 0})$, let $\mathcal{P}$ be an instance of $\mathrm{VCSP}(\Phi)$ *with variable set $\{v_1, \ldots, v_n\}$. Let $D_i = \{a \in D \mid \exists \mathbf{x} : x_i = a$ and $f_{\mathcal{P}}(\mathbf{x}) < \infty\}$. Suppose that $\langle \sqcap, \sqcup \rangle$ is a generalised STP multimorphism of $\mathcal{P}$. Then there is a generalised STP multimorphism $\langle \sqcap', \sqcup' \rangle$ of $\mathcal{P}$ in which each $\sqcap_i'$ is a total order on D_i (hence $\sqcup_i'$ is the reversal of this total order). Furthermore, for any set $J = \{i_1, \ldots, i_j\} \subseteq \{1, \ldots, n\}$ and any domain-reduced function $f \colon D_{i_1} \times \cdots \times D_{i_j} \to \overline{\mathbb{R}}_{\geq 0}$ for which $\langle \sqcap_J, \sqcup_J \rangle$ is a multimorphism, $\langle \sqcap_J', \sqcup_J' \rangle$ is also a multimorphism of f. The multimorphism $\langle \sqcap', \sqcup' \rangle$ is polynomial-time computable.*

This claim is proved (but not explicitly stated) in the proof of [35, Theorem 8.2]. The basic method is as follows. $\mathcal{P}$ is augmented with extra (redundant) valued constraints using unary and binary crisp cost functions. The binary crisp cost functions are used to enforce consistency so that when $\sqcap_i$ is modified to form a total order on D_i, a compatible modification is made to each other $\sqcap_j$. Once $\sqcap$ and $\sqcup$ are constructed, it is proved by induction that every relevant function f has the property specified in the claim. The induction is on the arity of f.

To prove the lemma, we use the claim with $\Phi = \Phi_0'$, $\mathcal{P} = I'$ and, for each valued constraint t of I', $f = f_t'$. □

Lemma 6.41. *If $\mathcal{F} \subseteq \mathrm{Func}(D, [0,1]_{\mathbb{Q}})$ and $\ell(\mathcal{F})$ has an STP/MJN multimorphism, then $\#\mathsf{CSP}(\mathcal{G})$ is* LSM*-easy for every finite $\mathcal{G} \subset \mathcal{F}$.*

Proof. Let $\mathcal{G}$ be a finite subset of $\mathcal{F}$. To any instance $I_{\#}$ of $\#\mathsf{CSP}(\mathcal{G})$ there corresponds an instance $I = \ell(I_{\#})$ of VCSP(Φ_0), where $\Phi_0 = \ell(\mathcal{G})$: for each weighted constraint t, the function F_t is mapped to $f_t = \ell(F_t)$ while the scope σ_t remains unchanged. Using Lemma 6.40, we may construct an equivalent instance I' of VCSP(Φ'_0) on the domain $D_1 \times \cdots \times D_n$ and a generalised STP multimorphism $\langle \sqcap, \sqcup \rangle$ of that instance, where each $\sqcap_i$ is a total order. $\langle \sqcap, \sqcup \rangle$ induces a generalised STP multimorphism of each f_t.

We now construct an instance I'' over the Boolean domain that is equivalent to I' and hence to I. For each i, $1 \le i \le n$, introduce a set of $|D_i| + 1$ Boolean variables $V_i = \{z_{i,a} \mid a \in D_i^+\}$, where $D_i^+ = D_i \cup \{\bot\}$. Extend the total order on D_i^+ by placing $\bot$ below all elements of D_i. Define a nested sequence of subsets of D_i^+ by $U_{i,a} = \{b \in D_i^+ \mid b < a\}$. The idea is that each domain element $a \in D_i$ is represented by the truth assignment that assigns 1 to all variables in $U_{i,a}$, and 0 to the others. Consider the constraint asserting that only these $|D_i|$ particular assignments to V_i are allowed. This constraint can be represented by the crisp cost function f that assigns $f(\mathbf{x}) = 0$ to these assignments and $f(\mathbf{x}) = \infty$ to all others. Note that $F(x) = \exp(-f(x))$ is log-supermodular.

Note that we can use the same relation for any pair of sets D_i and D_j with $|D_i| = |D_j|$ — if D_i and D_j have different total orders then the relation is applied to the variables in D_i^+ in a different order than to the variables in D_j^+. If we add these crisp valued constraints then there is a natural bijection between $D_1 \times \cdots \times D_n$ and feasible assignments to Boolean variables $V_1 \cup \cdots \cup V_n$. The variable $z_{i,a}$ where a is the smallest (respectively largest) element of D_i^+ always takes on the value 1 (respectively 0), and so these variables are redundant. However, their introduction simplifies the description of some constructions later in the proof.

Consider a valued constraint in I' of arity k that imposes the function $f' \in \Phi'_0$, and, without loss of generality, assume that its scope is the first k variables $x_1, \ldots, x_k$. Add a corresponding valued constraint f'' to I'' with $f'' \colon 2^{V_1 \cup \cdots \cup V_k} \to \overline{\mathbb{R}}_{\ge 0}$ defined as follows, where for convenience we are viewing f'' as a function on subsets of $V_1 \cup \cdots \cup V_k$ rather than as a function of $|V_1| + \cdots + |V_k|$ Boolean variables:

$$f''(A) = \begin{cases} f'(a_1, \ldots, a_k), & \text{if } A = U_{1,a_1} \cup \cdots \cup U_{k,a_k} \text{ for some } (a_1, \ldots, a_k); \\ \infty, & \text{otherwise.} \end{cases}$$

We claim f'' is submodular, i.e., $f''(A \cap B) + f''(A \cup B) \le f''(A) + f''(B)$. If either $f''(A) = \infty$ or $f''(B) = \infty$ there is nothing to prove. So $A = U_{1,a_1} \cup \cdots \cup U_{k,a_k}$ and

$B = U_{1,b_1} \cup \cdots \cup U_{k,b_k}$ for some $(a_1, \ldots, a_k), (b_1, \ldots, b_k) \in D_1 \times \cdots \times D_k$. Then

$$\begin{aligned}
&f''(A \cap B) + f''(A \cup B) \\
&= f''\big((U_{1,a_1} \cap U_{1,b_1}) \cup \cdots \cup (U_{k,a_k} \cap U_{k,b_k})\big) \\
&\quad + f''\big((U_{1,a_1} \cup U_{1,b_1}) \cup \cdots \cup (U_{k,a_k} \cup U_{k,b_k})\big) \\
&= f''(U_{1,a_1 \sqcap_1 b_1} \cup \cdots \cup U_{k,a_k \sqcap_k b_k}) \\
&\quad + f(U_{1,a_1 \sqcup_1 b_1} \cup \cdots \cup U_{k,a_k \sqcup_k b_k}) \\
&= f'(a_1 \sqcap_1 b_1, \ldots, a_k \sqcap_k b_k) + f'(a_1 \sqcup_1 b_1, \ldots, a_k \sqcup_k b_k) \\
&\leq f'(a_1, \ldots, a_k) + f'(b_1, \ldots, b_k) \\
&= f''(A) + f''(B).
\end{aligned}$$

Now take stock. We have an instance I'' of Boolean VCSP, which is equivalent to I' and hence to I. It has at most $n(|D|+1)$ Boolean variables and it has n more valued constraints than I. The number of distinct valued constraints in Φ_0'' is $|\Phi_0''| \leq |\Phi_0'| + |D|$; note that these come from a fixed set of cost functions independent of the instance I and hence of $I_\#$ itself.

Now map the VCSP instance I'' back to $\#\mathsf{CSP}$ to yield an instance $I_\#''$ over the Boolean domain in which every valued constraint comes from a certain fixed set of cost functions $\mathcal{F}_0'' \subset \mathsf{LSM}$. Specifically, $I'' = \ell(I_\#'')$ and $\Phi_0'' = \ell(\mathcal{F}_0'')$. Since I'' is equivalent to I, there is a bijection between the non-zero terms of $Z(I_\#)$ and $Z(I_\#'')$ that preserves weights, and hence $Z(I_\#) = Z(I_\#'')$. □

Lemma 6.41 shows that, if $\ell(\mathcal{F})$ has an STP/MJN multimorphism, then $\#\mathsf{CSP}(\mathcal{G})$ is LSM-easy for every finite $\mathcal{G} \subset \mathcal{F}$. Lemma 6.42 below strengthens the result by showing that $\#\mathsf{CSP}(\mathcal{G})$ is $\#\mathsf{BIS}$-easy. The strengthening applies when the weight functions in $\mathcal{F}$ have arity at most two.

In order to do the strengthening, we need to generalise the notion of a binary submodular function to cover binary functions over larger domains. A matrix M has the Monge property if, for every pair of rows r and r', and every pair of columns s and s',

$$M_{r \wedge r' \, s \wedge s'} + M_{r \vee r' \, s \vee s'} \leq M_{r \wedge r' \, s \vee s'} + M_{r \vee r' \, s \wedge s'},$$

where the $\wedge$ and $\vee$ operators are the minimum and maximum operators, respectively. To apply this concept here, suppose that f is a function with domain $D_i \times D_j$. Given orders on D_i and D_j, let $D_i(\ell)$ and $D_j(\ell)$ denote the ℓ'th element of D_i and D_j, respectively. We say that a function f with domain $D_i \times D_j$ is Monge (with respect to the given orders) if the $|D_i| \times |D_j|$ matrix M_f is Monge), where, as in Section 6.3, $(M_F)_{k\ell} = F(D_i(k), D_j(\ell))$. We say that a function $F : D_i \times D_j \to [0,1]_\mathbb{Q}$ is *log-anti-Monge* (with respect to the given orders) if the function $\ell(F)$ is Monge (with respect to the same orders). The "anti" in the term "log-anti-Monge" comes from the fact that $(\ell(F))(\mathbf{x}) = -\ln F(\mathbf{x})$, and the minus sign reverses the inequality in the definition of Monge. Thus, the relationship between

Monge and log-anti-Monge is analogous to the relationship between submodular and log-supermodular. The Monge property generalises submodularity and the property of being log-anti-Monge generalises the notion of log-supermodularity.

Lemma 6.42. *If $\mathcal{F} \subseteq \mathrm{Func}(D, [0,1]_{\mathbb{Q}})$ is a weighted constraint language whose weight functions have arity at most two and $\ell(\mathcal{F})$ has an STP/MJN multimorphism, then $\#\mathsf{CSP}(\mathcal{G})$ is $\#\mathsf{BIS}$-easy for every finite $\mathcal{G} \subset \mathcal{F}$.*

Proof. Let $\mathcal{G}$ be a finite subset of $\mathcal{F}$. We use exactly the same construction as in the previous lemma, but go further and show that every weight function F'' appearing in instance $I''_{\#}$ is expressible in terms of unary weight functions in $\mathcal{U}_{\{0,1\}}$, and the binary weight function IMP defined by $\mathrm{IMP}(0,0) = \mathrm{IMP}(0,1) = \mathrm{IMP}(1,1) = 1$ and $\mathrm{IMP}(1,0) = 0$. Moreover, unary weight functions in $\mathcal{U}_{\{0,1\}}$ (even those taking irrational values) can be approximated sufficiently closely by polynomial-sized pps-formulas using IMP [22, Lemma 36]. This will complete the proof, since $\#\mathsf{CSP}(\mathrm{IMP}) \leq_{\mathrm{AP}} \#\mathsf{BIS}$ by [47, Theorem 5].

The task then, is to show that every weight function F'' in instance $I''_{\#}$ is expressible in terms of unary weight functions in $\mathcal{U}_{\{0,1\}}$ and IMP. We do this by considering, in turn, the different types of weight functions arising in $I''_{\#}$. The n relations (crisp cost functions) that were introduced in I'' to impose a total order on the variables in the sets V_i are clearly implementable in terms of $\mathrm{imp} = \ell(\mathrm{IMP})$.

Every other weight function F'' is associated with a cost function f'' in I'' that is an implementation over the Boolean domain of a cost function f' from I'. Since $f' \in \Phi_0$, it has arity at most 2. Our goal is to show that the function $F'(\mathbf{x}) = \ell^{-1}(f'(\mathbf{x})) = \exp(-f'(\mathbf{x}))$ is expressible in terms of unary weight functions in $\mathcal{U}_{\{0,1\}}$ and IMP. If f' is unary, this is immediate, so suppose f' is binary.

To fix the notation, suppose that f' is a function $f' \colon D_i \times D_j \to \overline{\mathbb{R}}_{\geq 0}$. We can assume without loss of generality that D_i and D_j are disjoint (otherwise, rename some elements). Also, D_i and D_j are ordered according to the linear order induced by $\sqcap$. Since $\langle \sqcap, \sqcup \rangle$ induces a generalised STP multimorphism of f' (see Definition 6.31), the function f' is Monge (with respect to this order). We start by considering two special cases.

- First, suppose that F' is strictly positive. (That is, there is no (x,y) with $f'(x,y) = \infty$ so the range of f' is contained $\mathbb{R}_{\geq 0}$). Rudolf and Woeginger [88] have shown that every nonnegative Monge matrix is expressible as a positive linear combination of certain simple Monge matrices. Translated to our setting by applying ℓ^{-1}, this says that F' is expressible as a product of certain simple basis functions, namely: (i) the unary functions

$$B_a(x) = \begin{cases} \alpha, & \text{if } x = a; \\ 1, & \text{otherwise,} \end{cases}$$

for all $a \in D_i$ (with a similar set of unary functions defined over D_j), and (ii) the binary functions

$$B_{a,b}(x,y) = \begin{cases} \alpha, & \text{if } x \le a \text{ and } y \ge b; \\ 1, & \text{otherwise,} \end{cases}$$

for all $(a,b) \in (D_i, D_j)$ (with a similar set of binary function defined by replacing $x \le a$ and $y \ge b$ by $x \ge a$ and $y \le b$). In both cases (i) and (ii), α is an arbitrary constant in the range $(0,1)$.

Using the Boolean variables $V_i \cup V_j$, the basis function $B_a(x)$ may be implemented as $U_\alpha(z_{i,a})\, U_{1/\alpha}(z_{i,a^-})$, where

$$U_\beta(z) = \begin{cases} \beta, & \text{if } z = 0; \\ 1, & \text{if } z = 1, \end{cases}$$

and a^- is the element immediately below a in the total order on D_i. Also, the basis function $B_{a,b}(x,y)$ may be implemented as $\mathrm{IMP}(z_{i,a}, y)\, \mathrm{IMP}(z_{j,b^-}, y) \times U_\alpha(z_{i,a})\, U_{1/\alpha-1}(y)$, where y is a new variable.

- Second, suppose that the range of F' is $\{0,1\}$ (that is, f' is a crisp cost function). As in the proof of Lemma 6.41, let $U_{i,a} = \{b \in D_i^+ \mid b < a\}$. Let F^* be the set of subsets of $V_i \cup V_j$ defined as follows.

$$F^* = \{U_{i,a} \cup U_{j,b} \mid F'(a,b) = 1\}$$

F' is log-anti-Monge so

$$F'(x \wedge x', y \wedge y')F'(x \vee x', y \vee y') \ge F'(x,y)F'(x',y').$$

Thus, F^* is closed under unions and intersections. Hence by [48, Corollary 18], F^* is a conjunction of implications and constants, and hence can be implemented using IMP and unary weights.

To finish the proof, we use a construction from [22, Lemma 23]. Assume without loss of generality that $D_i = \{1, \dots, |D_i|\}$ and $D_j = \{1, \dots, |D_j|\}$ with the usual total order. Define $H'\colon D_i \times D_j \to [0,1]_{\mathbb{Q}}$ by putting $H'(x,y) = F'(x,y)$ for all $x \in D_i$, $y \in D_j$ except that, if $F'(1,1) = 0$, we put $H'(1,1) = 1$. Note that H' is log-anti-Monge (by construction, since F' is log-anti-Monge) and $H'(1,1) \ne 0$. Define $G\colon D_i \times D_j \to \mathbb{Q}_{\ge 0}$ by

$$G(x,y) = \max\{H'(x^*, y^*)\mu^{x+y-x^*-y^*} \mid x^* \le x \text{ and } y^* \le y\},$$

where

$$\mu = \max\{m \geq 0 \mid H'(x^*, y^*)m^{x+y-x^*-y^*} \leq H'(x,y) \\ \text{for all } (x,y),(x^*,y^*) \in D_i \times D_j \\ \text{with } H'(x,y) > 0 \text{ and } x^* \leq x \text{ and } y^* \leq y\}.$$

From the definition of μ, we can see that $\mu > 0$, so G is strictly positive. We will show that G is log-anti-Monge. Let $x, x' \in D_i$ and $y, y' \in D_j$. There exist $x^* \leq x$ and $x'^* \leq x'$ and $y^* \leq y$ and $y'^* \leq y'$ such that

$$\begin{aligned} G(x,y)G(x',y') &= H'(x^*,y^*)H'(x'^*,y'^*)\mu^{x+y+x'+y'-x^*-y^*-x'^*-y'^*} \\ &\leq H'(x^* \wedge x'^*, y^* \wedge y'^*)\mu^{x \wedge x' - x^* \wedge x'^* + y \wedge y' - y^* \wedge y'^*} \\ &\quad \cdot H'(x^* \vee x'^*, y^* \vee y'^*)\mu^{x \vee x' - x^* \vee x'^* + y \vee y' - y^* \vee y'^*} \\ &\leq G(x \wedge x', y \wedge y')G(x \vee x', y \vee y'). \end{aligned}$$

Now, from the definition of G and μ we can see that $H'(x,y) = R_{H'}(x,y)G(x,y)$, where $R_{H'}$ is the support of H' (considered as a zero-one valued function). Thus, by the definition of H', $F'(x,y) = R_{F'}(x,y)G(x,y)$. Then, by the above constructions, F' can be implemented in terms of IMP and unary functions. (The first of the cases above shows how to implement G, which is strictly positive; the second shows how to implement R, whose range is $\{0,1\}$). □

To use Lemmas 6.41 and 6.42, we need to perform some scaling. For any k-ary weight function in $F \in \mathcal{F}$, let $m_F = \max\{f(\mathbf{x}) \mid \mathbf{x} \in D^k\}$. Let

$$\Lambda(F) = \begin{cases} F/m_F & \text{if } m_F > 1 \\ F & \text{otherwise} \end{cases}$$

and let $\Lambda(\mathcal{F}) = \{\Lambda(F) \mid F \in \mathcal{F}\}$. Note that $\Lambda(F)$ always takes values in $[0,1]_{\mathbb{Q}}$ and that, since $\mathcal{F}$ is conservative, $\mathrm{Func}_1(D, [0,1]_{\mathbb{Q}}) \subseteq \Lambda(\mathcal{F})$.

We return, once more, to the proof of Theorem 6.4.

Theorem 6.43. *Let $\mathcal{F}$ be a weakly log-supermodular, conservative weighted constraint language taking values in $\mathbb{Q}_{\geq 0}$.*

- *For any finite $\mathcal{G} \subset \mathcal{F}$, there is a finite $\mathcal{G}' \subset$ LSM such that $\#\mathsf{CSP}(\mathcal{G}) \leq_{\mathrm{AP}} \#\mathsf{CSP}(\mathcal{G}')$.*
- *If $\mathcal{F}$ consists of functions of arity at most two, then $\#\mathsf{CSP}(\mathcal{G})$ is $\#\mathsf{BIS}$-easy for any finite $\mathcal{G} \subset \mathcal{F}$.*

Proof. By Theorem 6.30, $\ell(\lambda(\mathcal{F}))$ has an STP/MJN multimorphism. The result follows from Lemmas 6.41 and 6.42 and the fact that $\#\mathsf{CSP}(\mathcal{F}) \leq_{\mathrm{AP}} \#\mathsf{CSP}(\Lambda(\mathcal{F}))$. □

Theorem 6.4, our classification of the complexity of approximating $\#\mathsf{CSP}(\mathcal{F})$, now follows from Theorems 6.7, 6.12 and 6.43.

6.7 Algorithmic aspects

Finally, we consider the algorithmic aspects of the classification of Theorem 6.4. Intuitively, there is an algorithm that determines the complexity of $\#\mathsf{CSP}$ with constraints from a finite language $\mathcal{H}$ plus unary weights because weak log-modularity is essentially equivalent to balance and weak log-supermodularity is essentially equivalent to the existence of a STP/MJN multimorphism. Balance and the existence of STP/MJN multimorphisms depend only on certain finite parts of the weighted constraint language so balance is decidable by [24] and the existence of STP/MJN multimorphisms can be determined by brute force, or by using more sophisticated methods from [77].

We need to determine whether the infinite language $\mathcal{H}\cup\mathcal{U}_D$ is balanced. Fortunately, it suffices to check whether $\mathcal{H}\cup\mathcal{U}'_D$ is balanced, where $\mathcal{U}'_D = \mathrm{Func}_1(D,\{1,2\})$, which is finite. (Note that it is not enough to test whether $\mathcal{H}$ is balanced; also, there is nothing special about 1 and 2: any pair of distinct, positive rationals would do. In fact, $|\mathcal{U}'_D| = 2^{|D|}$ and there are sets of size $|D|$ which would suffice, but we do not need this here.)

Lemma 6.44. *Let $\mathcal{H}$ be a finite, weighted constraint language taking values in $\mathbb{Q}_{\geq 0}$. The following are equivalent: (1) $\mathcal{H}\cup\mathcal{U}'_D$ is balanced; (2) every finite subset of $\mathcal{H}\cup\mathcal{U}_D$ is balanced; and (3) $\mathcal{H}\cup\mathcal{U}_D$ is balanced.*

Proof. (2) and (3) are equivalent because any pps-formula contains only a finite number of atomic formulas. (2) trivially implies (1), since $\mathcal{U}'_D$ is finite. It remains to show that (1) implies (2) so, towards this goal, suppose that $\mathcal{H}\cup\mathcal{U}'_D$ is balanced. We must show that every finite subset of $\mathcal{H}\cup\mathcal{U}_D$ is balanced. Suppose that such a subset contains r functions in $\mathcal{U}_D\setminus\mathcal{H}$.

Let $\{F_1,\dots,F_r\}$ be unary functions such that $F_i(d) = a_{i,d}$ ($i\in\{1,\dots,r\}$, $d\in D$) and let $\mathcal{G} = \mathcal{H}\cup\{F_1,\dots,F_r\}$. We may consider the $a_{i,d}$ as formal variables and treat a function $G\in\langle\mathcal{G}\rangle_\#$ with free variables $\mathbf{x}$ as a function of both $\mathbf{x}$ and the $a_{i,d}$. We will show that, for any function G and any interpretation of the $a_{i,d}$ (i.e., any instantiation of the function symbols F_i as concrete functions $D\to\mathbb{Q}_{\geq 0}$), the matrices associated with G have block-rank 1, thus establishing that $\mathcal{G}$ is balanced.

So, consider any $G\in\langle\mathcal{G}\rangle_\#$ with arity $n\geq 2$ and choose any k with $1\leq k<n$. We will show that the $D^k\times D^{n-k}$ matrix $M_G(\mathbf{x},\mathbf{y})$ has block-rank 1 for any value of the $a_{i,d}$. By Lemma 6.8 part (2), it suffices to show that every 2×2 submatrix induced by rows $\mathbf{x},\mathbf{x}'$ and columns $\mathbf{y},\mathbf{y}'$ has block-rank 1. By Lemma 6.8 part (1), this happens if, and only if, every such submatrix has rank 1 or at least two zero entries, which happens if, and only if, the multivariate polynomial

$$p = G(\mathbf{x},\mathbf{y}')G(\mathbf{x}',\mathbf{y})G(\mathbf{x}',\mathbf{y}')\big[G(\mathbf{x},\mathbf{y})G(\mathbf{x}',\mathbf{y}') - G(\mathbf{x}',\mathbf{y})G(\mathbf{x},\mathbf{y}')\big]$$

is zero for all values of $\mathbf{x}, \mathbf{x}', \mathbf{y}, \mathbf{y}'$ and for all values of the $a_{i,d}$. (Note that, if the submatrix defined by a pair of rows and columns does not have block-rank 1 but has exactly one zero, then only one of the four possible choices for $\mathbf{x}, \mathbf{x}', \mathbf{y}, \mathbf{y}'$ will make p non-zero.)

We now fix $\mathbf{x}, \mathbf{x}', \mathbf{y}, \mathbf{y}'$ and consider p as a function of just the $a_{i,d}$. Our goal is to show that (for every choice of $\mathbf{x}, \mathbf{x}', \mathbf{y}, \mathbf{y}'$), p is identically 0.

Consider first the case where every $a_{i,d}$ is a power of two. Here, every atomic formula $F_i(z)$ defines the same function as some product $U_1(z) \cdots U_\ell(z)$ of atomic formulas from $\mathcal{U}'_D$ so G is equivalent to some function in $\langle \mathcal{H} \cup \mathcal{U}'_D \rangle_\#$. But $\mathcal{H} \cup \mathcal{U}'_D$ is balanced by assumption, so $p = 0$ whenever every $a_{i,d}$ is a power of two. Therefore, $p = 0$ over a space that is a product of infinite sets. It follows from the Schwartz–Zippel lemma or from [1, Theorem 1.2] that the only polynomial with this property is the zero polynomial, so p is the zero polynomial and $\mathcal{H} \cup \{F_1, \dots, F_r\}$ is balanced for any set $\{F_1, \dots, F_r\}$ of unary weights. $\square$

Theorem 6.45. *There is an algorithm that, given a finite, weighted constraint language $\mathcal{H}$ taking values in $\mathbb{Q}_{\geq 0}$, correctly makes one of the following deductions, where $\mathcal{F} = \mathcal{H} \cup \mathcal{U}_D$:*

1. $\#\mathsf{CSP}(\mathcal{G})$ *is in* FP *for every finite $\mathcal{G} \subset \mathcal{F}$;*
2. $\#\mathsf{CSP}(\mathcal{G})$ *is* LSM*-easy for every finite $\mathcal{G} \subset \mathcal{F}$ and* $\#\mathsf{BIS}$*-hard for some such $\mathcal{G}$;*
3. $\#\mathsf{CSP}(\mathcal{G})$ *is* $\#\mathsf{BIS}$*-easy for every finite $\mathcal{G} \subset \mathcal{F}$ and* $\#\mathsf{BIS}$*-equivalent for some such $\mathcal{G}$;*
4. $\#\mathsf{CSP}(\mathcal{G})$ *is* $\#\mathsf{SAT}$*-easy for every finite $\mathcal{G} \subset \mathcal{F}$ and* $\#\mathsf{SAT}$*-equivalent for some such $\mathcal{G}$.*

If every function in $\mathcal{H}$ has arity at most 2, the output is not deduction 2.

Proof. We reduce the problem to determining whether $\mathcal{H} \cup \mathcal{U}'_D$ is balanced, whether $\ell(\mathcal{H})$ has an STP/MJN multimorphism and whether $\mathcal{H}$ contains only functions of arity at most 2. Balance of finite languages is decidable [24]. An STP/MJN multimorphism consists of two operations $D^2 \to D$ and three operations $D^3 \to D$, which must have certain easily checked properties with respect to each of the functions in $\ell(\mathcal{H})$. Thus, we can determine the existence of an STP/MJN multimorphism by brute force, checking each possible collection of five operations, or by using the methods of Kolmogorov and Živný [77]. It is clearly decidable whether $\mathcal{H}$ contains a function of arity greater than 2.

By Lemma 6.44, if $\mathcal{H} \cup \mathcal{U}'_D$ is balanced, then so is any finite $\mathcal{G} \subset \mathcal{H} \cup \mathcal{U}_D$. Therefore, by Lemma 6.11, $\#\mathsf{CSP}(\mathcal{G})$ can be solved exactly in FP so we output deduction 1. From this point, we assume that $\mathcal{H} \cup \mathcal{U}'_D$ is not balanced.

Since $\mathcal{H} \cup \mathcal{U}'_D$ is not balanced, nor is $\mathcal{H} \cup \mathcal{U}_D$ (Lemma 6.44). Therefore, $\mathcal{H} \cup \mathcal{U}_D$ is not weakly log-modular (Lemma 6.10) so there is a finite $\mathcal{G} \subset \mathcal{H} \cup \mathcal{U}_D$ such that $\#\mathsf{CSP}(\mathcal{G})$ is $\#\mathsf{BIS}$-hard (Theorem 6.7).

$\ell(\Lambda(\mathcal{H} \cup \mathcal{U}_D))$ has an STP-MJN multimorphism if, and only if, $\ell(\Lambda(\mathcal{H}))$ does (Observation 1), and $\ell(\Lambda(\mathcal{H}))$ is a finite language so we can determine whether it has an STP-MJN multimorphism by exhaustive search. If $\ell(\Lambda(\mathcal{H} \cup \mathcal{U}_D))$ has an STP-MJN multimorphism, then, for all finite $\mathcal{G} \subset \Lambda(\mathcal{H} \cup \mathcal{U}_D)$, $\#\mathsf{CSP}(\mathcal{G})$ is LSM-easy (Lemma 6.41). Since any function in $\Lambda(\mathcal{H} \cup \mathcal{U}_D)$ is a scalar multiple of some function in $\mathcal{H} \cup \mathcal{U}_D$, $\#\mathsf{CSP}(\mathcal{G})$ is also LSM-easy for all finite $\mathcal{G} \subset \mathcal{H} \cup \mathcal{U}_D$. We output deduction 2, unless every function in $\mathcal{H}$ has arity at most 2, in which case $\#\mathsf{CSP}(\mathcal{G})$ is $\#\mathsf{BIS}$-easy for all finite $\mathcal{G} \subset \mathcal{H} \cup \mathcal{U}_D$ (Lemma 6.42) and we output deduction 3.

On the other hand, if $\ell(\Lambda(\mathcal{H} \cup \mathcal{U}_D))$ has no STP-MJN multimorphism, then $\Lambda(\mathcal{H} \cup \mathcal{U}_D)$ is not weakly log-supermodular (Theorem 6.30). Because Λ is just a rescaling, $\mathcal{H} \cup \mathcal{U}_D$ is also not weakly log-supermodular. Therefore, there is a finite $\mathcal{G} \subset \mathcal{H} \cup \mathcal{U}_D$ such that $\#\mathsf{CSP}(\mathcal{G})$ is $\#\mathsf{SAT}$-equivalent (Theorem 6.7 again). We output deduction 4. $\square$

Chapter 7

LSM is not generated by binary functions

(This chapter is based on part of [22].)

Recall that a function $F\colon \{0,1\}^n \to \mathbb{R}_{\geq 0}$ is *log-supermodular* if

$$F(\mathbf{x} \vee \mathbf{y})F(\mathbf{x} \wedge \mathbf{y}) \geq F(\mathbf{x})F(\mathbf{y}) \text{ for all } \mathbf{x}, \mathbf{y} \in \{0,1\}^n.$$

In Section 6.1.3 we defined $\langle \mathcal{F} \rangle_\#$ as the set of functions that are "pps-definable" by $\mathcal{F}$. The important property is that $\#\mathsf{CSP}(\mathcal{F}) \leq_{\mathrm{AP}} \#\mathsf{CSP}(\mathcal{G})$ if $\mathcal{F} \subset \langle \mathcal{G} \rangle_\#$, so we can rule out a natural class of AP-reductions $\#\mathsf{CSP}(\mathcal{F}) \leq_{\mathrm{AP}} \#\mathsf{CSP}(\mathcal{G})$ by showing $\mathcal{F} \not\subset \langle \mathcal{G} \rangle_\#$.

We denote the set of log-supermodular functions by LSM, and we denote the set of log-supermodular functions of arity at most k by LSM_k. In [22] it was shown that $\langle \mathsf{LSM} \rangle_\# = \mathsf{LSM}$ and $\langle \mathsf{LSM}_2 \rangle_\# = \langle \mathsf{LSM}_3 \rangle_\#$ and $\langle \mathsf{LSM}_3 \rangle_\# \subsetneqq \langle \mathsf{LSM}_4 \rangle_\#$. This situation mirrors known results about valued CSPs [106]. In this chapter we present the proof that $\langle \mathsf{LSM}_3 \rangle_\# \subsetneqq \langle \mathsf{LSM}_4 \rangle_\#$.

In fact, in [22] it was shown that limits can also give AP-reductions, subject to some requirements about computational efficiency. Let $\langle \mathcal{F} \rangle_{\#,\omega}$ denote the set of functions $F\colon \{0,1\}^n \to \mathbb{R}_{\geq 0}$ such that there is a finite set $S \subseteq \mathcal{F}$ such that there are $F_1, F_2, \dots \in \langle S \rangle_\#$ satisfying

$$\max_{\mathbf{x} \in \{0,1\}^n} |F(\mathbf{x}) - F_n(\mathbf{x})| \to 0 \text{ as } n \to \infty.$$

A pps_ω**-definable functional clone** is a set of the form $\langle \mathcal{F} \rangle_{\#,\omega}$. Equivalently (by [22, Lemma 2]), it is a set $\mathcal{F}$ such that $\langle \mathcal{F} \rangle_{\#,\omega} = \mathcal{F}$.

We show the technically stronger statement $\langle \mathsf{LSM}_3 \rangle_{\#,\omega} \subsetneqq \langle \mathsf{LSM}_4 \rangle_{\#,\omega}$, which rules out a class of reductions based on pps-definability and taking limits.

7.1 Notation

We again use the Func notation, denoting the set of functions $\{0,1\}^k \to \mathbb{R}_{\geq 0}$ by $\mathrm{Func}_k(\{0,1\}, \mathbb{R}_{\geq 0})$ and denoting $\bigcup_{k \geq 0} \mathrm{Func}_k(\{0,1\}, \mathbb{R}_{\geq 0})$ by $\mathrm{Func}(\{0,1\}, \mathbb{R}_{\geq 0})$. Recall the definitions of $\oplus$ and $F^\star$ from Section 1.7.2.

7.2 Non-negativity of Fourier coefficients

First we consider the class[1] $\mathcal{P}$ of functions $F \in \mathrm{Func}(\{0,1\}, \mathbb{R}_{\geq 0})$ for which the *Fourier transform*[2] $\widehat{F}$ has nonnegative coefficients, where

$$\widehat{F}(\mathbf{y}) \;=\; \frac{1}{2^n} \sum_{\mathbf{w}\in\{0,1\}^n} (-1)^{y_1 w_1 + \cdots + y_n w_n} F(\mathbf{w}) \qquad\qquad (\mathbf{y} \in \{0,1\}^n). \tag{7.1}$$

Thus $F \in \mathcal{P}$ if and only if $\widehat{F} \in \mathrm{Func}(\{0,1\}, \mathbb{R}_{\geq 0})$. See [43] for further information. We will use (7.1) and the *convolution theorem*: for all $F, G \in \mathrm{Func}_n(\{0,1\}, \mathbb{R}_{\geq 0})$ we have

$$\widehat{FG}(\mathbf{x}) \;=\; \sum_{\mathbf{y}\in\{0,1\}^n} \widehat{F}(\mathbf{y})\widehat{G}(\mathbf{x} \oplus \mathbf{y}) \qquad\qquad (\mathbf{x}, \mathbf{y} \in \{0,1\}^n). \tag{7.2}$$

See for example [43, Section 2.3] for a proof of the dual statement.

To show that $\mathcal{P}$ is closed under pps-formula evaluation, it is useful to restrict to atomic formulas where variables are not repeated within a scope.

Lemma 7.1. *Let $\mathcal{F} \subseteq \mathrm{Func}(\{0,1\}, \mathbb{R}_{\geq 0})$. For all pps-formulas ψ over $\mathcal{F}$ there is another pps-formula ψ' over $\mathcal{F} \cup \{\mathrm{EQ}_2\}$ such that $F_\psi = F_{\psi'}$ and no atomic formula of ψ' contains a repeated variable.*

Proof. Given ψ obtain ψ' as follows. For each variable v_i that is used $d_i \geq 2$ times in total in ψ, replace the uses of v_i by new distinct variables $v_i^1, \cdots, v_i^{d_i}$, multiply by atomic formulas $\mathrm{EQ}_2(v_i, v_i^j)$ for $1 \leq j \leq d_i$, then sum over these new variables v_i^j. ☐

Lemma 7.2. *$\mathcal{P}$ is closed under addition, summation, products and limits. Moreover, $\mathcal{P}$ is a pps_ω-definable functional clone.*

Proof. If $F, G \in \mathcal{P}$, then $\widehat{F+G} = \widehat{F} + \widehat{G}$ is clearly non-negative, and $\widehat{FG}$ is nonnegative by the convolution theorem (7.2). For summation, by induction we only need to consider summing over the last variable. So, let $H(\mathbf{x}) = \sum_t F(\mathbf{x}, t)$. Then it follows easily from (7.1) that $\widehat{H}(\mathbf{y}) = 2\widehat{F}(\mathbf{y}, 0) \geq 0$ for all $\mathbf{y}$. For limits note that if $F_n \to F$ then $\widehat{F_n} \to \widehat{F}$, and a limit of non-negative functions is non-negative.

Let ψ be a pps-formula over $\mathcal{P} \cup \{\mathrm{EQ}_2\}$. We will argue that that $F_\psi \in \mathcal{P}$. By Lemma 7.1 there is a pps-formula ψ' over $\mathcal{P} \cup \{\mathrm{EQ}_2\}$ such that $F_\psi = F_{\psi'}$ and such that no atomic formula of ψ contains a repeated variable. The functions F_φ defined by atomic formulas $\varphi = G(v_{i_1}, \cdots, v_{i_k})$ of ψ' are therefore "expansions": permutations of the function $G' \in \mathrm{Func}_n(\{0,1\}, \mathbb{R}_{\geq 0})$, $n \geq k$, defined by

$$G'(\mathbf{x}, \mathbf{x}') \;=\; G(\mathbf{x}) \qquad\qquad (\mathbf{x} \in \{0,1\}^k \text{ and } \mathbf{x}' \in \{0,1\}^{n-k}). \tag{7.3}$$

[1] Not to be confused with the class $\mathcal{P}$ used in Section 4.3.

[2] This is the same as the Hadamard transform mentioned in Section 3.1.1, but we use a different normalisation here.

It therefore suffices to check that $\mathcal{P}$ is closed under expansions. Let G' be the expansion defined by (7.3). Then, for all $\mathbf{y} \in \{0,1\}^k$ and $\mathbf{y}' \in \{0,1\}^{n-k}$, we have $\widehat{G'}(\mathbf{y},\mathbf{y}') = \widehat{G}(\mathbf{y})$ if $\mathbf{y}' = 0^k$, and $\widehat{G'}(\mathbf{y},\mathbf{y}') = 0$ otherwise, and hence $G' \in \mathcal{P}$. Note that $\widehat{\mathrm{EQ}_2} = \frac{1}{2}\mathrm{EQ}_2$, so $\mathrm{EQ}_2 \in \mathcal{P}$. Thus $\mathcal{P}$ is a functional clone, but it is also closed under limits. □

7.3 A class containing binary log-supermodular functions

Let $\mathcal{C}$ be the class of functions $F \in \mathrm{Func}(\{0,1\},\mathbb{R}_{\geq 0})$ such that $G^\star \in \mathcal{P}$ for every pinning $G = F(\mathbf{c},\cdot)$. Note, in particular, that if $U \in \mathrm{Func}_1(\{0,1\},\mathbb{R}_{\geq 0})$, $U^\star(z) = U(0)U(1)$, a nonnegative constant. Therefore we have $\mathrm{Func}_1(\{0,1\},\mathbb{R}_{\geq 0}) \subseteq \mathcal{C}$ and, to establish that $F \in \mathcal{C}$, we need only check pinnings of F of arity at least 2.

Lemma 7.3. *$\mathcal{C}$ is a* pps_ω*-definable functional clone.*

Proof. As in Lemma 7.2 we will check that $\mathcal{C}$ is closed under "expansions", products, summations, and limits. But a pinning of an expansion (or product, summation, or limit) of functions in $\mathcal{C}$ is an expansion (or product, summation, or limit) of pinnings of functions in $\mathcal{C}$, which are necessarily in $\mathcal{C}$ because $\mathcal{C}$ is closed under pinnings. So it suffices to check the $\mathcal{C}$ condition for trivial pinnings, for example to check closure under products it suffices to show that $F, G \in \mathcal{C}$ implies $(FG)^\star \in \mathcal{P}$.

Let $G \in \mathcal{C}$ have arity k, let $n \geq k$, let G' be the function defined by (7.3). Note that $G'\overline{G'}$ is an expansion of $G\overline{G}$, so $G'\overline{G'} \in \mathcal{P}$ and $G' \in \mathcal{C}$. We have $\mathrm{EQ}_2 \in \mathcal{C}$, since $\mathrm{EQ}_2\overline{\mathrm{EQ}_2} = \mathrm{EQ}_2 \in \mathcal{P}$. Closure under product follows from Lemma 7.2 and the observation that $(FG)^\star = F^\star G^\star$. For summation, by induction we may consider only summing over the last variable. Then, if $H(\mathbf{x}) = \sum_t F(\mathbf{x},t)$, where F has arity $k+1$, then

$$H^\star(\mathbf{x}) = \textstyle\sum_t F(\mathbf{x},t)\sum_t \overline{F}(\mathbf{x},t) = (F_0)^\star(\mathbf{x}) + (F_1)^\star(\mathbf{x}) + \sum_t F^\star(\mathbf{x},t).$$

where F_0 and F_1 are the pinnings defined by $F_i(x_1,\dots,x_k) = F(x_1,\dots,x_k,i)$. We have $(F_0)^\star, (F_1)^\star \in \mathcal{P}$ by the pinning assumption, and the arity k function $\sum_t F^\star(x_1,\dots,x_k,t)$ is in $\mathcal{P}$ by Lemma 7.2. Thus $H^\star$ is the sum of three functions in $\mathcal{P}$, and so, using Lemma 7.2 again, $H^\star \in \mathcal{P}$. Finally note that $\mathcal{C}$ is closed under limits: if $F_n \to F$ as $n \to \infty$ then $F_n^\star \to F^\star$, but $\mathcal{P}$ is closed under limits. □

Lemma 7.4. $\langle \mathsf{LSM}_2\rangle_{\#,\omega} \subseteq \mathcal{C}$.

Proof. Let $F \in \mathsf{LSM}_2$. Note that $\widehat{F^\star}(0,0) = (F(0,0)F(1,1) + F(0,1)F(1,0))/2$, and $\widehat{F^\star}(0,1) = \widehat{F^\star}(1,0) = 0$, and $\widehat{F^\star}(1,1) = (F(0,0)F(1,1) - F(0,1)F(1,0))/2 \geq 0$. So $F^\star \in \mathcal{P}$, and hence $F \in \mathcal{C}$. Thus $\mathsf{LSM}_2 \subseteq \mathcal{C}$ and, since $\mathcal{C}$ is a pps_ω-definable functional clone, $\langle \mathsf{LSM}_2\rangle_{\#,\omega} \subseteq \mathcal{C}$. □

Lemma 7.5. $\langle \mathsf{LSM}_2\rangle_{\#,\omega} \subsetneq \langle \mathsf{LSM}_4\rangle_{\#,\omega}$.

Proof. Since $\mathsf{LSM}_2 \subseteq \mathcal{C}$ by Lemma 7.4, we need only exhibit a function $F \in \mathsf{LSM}_4$ which is not in $\mathcal{C}$. Define $F : \{0,1\}^4 \to \mathbb{R}_{>0}$ by

$$F(x_1, x_2, x_3, x_4) = \begin{cases} 4, & \text{if } x_1 + x_2 + x_3 + x_4 = 4; \\ 2, & \text{if } x_1 + x_2 + x_3 + x_4 = 3; \\ 1, & \text{otherwise.} \end{cases}$$

To show $F \in \mathsf{LSM}_4$, by the symmetry of F and Lemma 5.36, it suffices to show that the three arity 2 pinnings $F(0,0,x_3,x_4)$, $F(0,1,x_3,x_4)$ and $F(1,1,x_3,x_4)$ are log-supermodular. This is equivalent to the inequalities $1 \times 1 \geq 1 \times 1$, $2 \times 1 \geq 1 \times 1$, and $4 \times 1 \geq 2 \times 2$ respectively, which clearly hold.

To show that $F \notin \mathcal{C}$, we need only use (7.1) to calculate

$$\widehat{F^{\star}}(1,1,1,1) \;=\; \frac{4 \times 1 - 4 \times 2 + 6 \times 1 - 4 \times 2 + 4 \times 1}{2^4} \;=\; -\frac{1}{8} \;<\; 0.$$

Indeed $F \in \langle \mathsf{LSM}_4 \rangle_{\#,\omega}$ but $F \notin \mathcal{C}$. By Lemma 7.4, $\langle \mathsf{LSM}_2 \rangle_{\#,\omega} \subseteq \langle \mathsf{LSM}_4 \rangle_{\#,\omega} \cap \mathcal{C} \subsetneq \langle \mathsf{LSM}_4 \rangle_{\#,\omega}$. $\square$

Chapter 8

Conclusions

We studied the complexity of approximately evaluating various classes of partition functions, by finding FPRASes and AP-reductions, and by studying notions of expressibility. Apart from the results themselves, we have demonstrated various new techniques and ideas:

- Contour arguments can be used to analyse a gadget in an NP-hardness proof. Baker's approximation techniques for planar graph problems can be applied to partition functions. (Chapter 2)
- Jerrum and Sinclair's cycle-unwinding argument can be interpreted as an argument about Holant problems; perfect matchings constraints generalise to windable functions. The same class of functions puts a limit on which functions can be expressed in the context of counting perfect matchings. (Chapter 3)
- While #BIS has appeared in previous #CSP classifications, #PM has an important role in degree-two #CSP classifications. (Chapters 4 and 5).
- Feder's arguments about degree-two CSPs using non-delta-matroids can be adapted to degree-two #CSPs using non-terraced functions. (Chapter 5).
- Arguments coming from the study of VCSPs, as well as arguments coming from the study of the exact evaluation of #CSPs, can sometimes be adapted to the context of approximate evaluation of #CSPs. (Chapter 6)
- The expressibility of #CSPs can be studied using the Fourier transform. (Chapter 7)

8.1 Questions for future research

In Chapter 2 we ruled out FPRASes for evaluating the partition function of the hard-core model on planar graphs when the fugacity is at least 312. This number seems likely to be far from optimal. The difficulty here is obtaining the inequality $p^{=} > p^{\neq}$ (Lemma 2.14). It seems plausible that we only really need the existence of a phase transition, which is known to occur for square lattices for $\lambda > 5.3646$, and conjectured to occur for λ greater than about 3.79 [6]. A converse result would also be interesting — is there an FPRAS for the partition function of the hard-core model on (unbounded degree) planar graphs, at sufficiently low fugacities?

In Chapter 3 we gave FPRASes for Holant problems using windable strictly terraced functions. Is there an FPRAS for Holant problems using a larger class of functions? We also initiated the study of expressibility for $\#\mathsf{PM}$, and characterised which arity three functions have matchings circuits. Is there a characterisation of which arity four functions have matchings circuits? Does every windable function have a matchings circuit?

In Chapter 4 we started to classify unweighted degree-two #CSPs with a single arity three Boolean relation. Are there more AP-reductions between these problems? We also looked at the restriction of $\#\mathsf{Downsets}$ to dags of maximum degree three. What about other restrictions? Is there an FPRAS for $\#\mathsf{Downsets}$ restricted to dags that have depth 2 and maximum degree 6, in other words degree-six $\#\mathsf{BIS}$ (see Remark 4.18)? What about depth 3 and maximum degree 3?

In Chapter 5 we classified degree-two Boolean #CSPs with variable weights, for unweighted constraint languages. As discussed after the statement of Theorem 5.3, this classification could be refined. For example, is $\#\mathsf{CSP}^{\geq 0}_{\leq 2}(R)$ is NP-hard to approximate, if $R = \{(x, y, z) \in \{0, 1\} \mid x + y + z \in \{0, 2, 3\}\}$? We explored various extensions of this classification: allowing weighted constraint languages, and restricting the weights. There are many unclassified problems here, even with higher degree bounds.

In Chapter 6 we studied conservative #CSPs, which of course leaves open the study of non-conservative #CSPs. In Chapter 7 we showed $\mathsf{LSM}_4 \not\subset \langle \mathsf{LSM}_2 \rangle_\#$. Is there a characterisation of which arity four functions are in $\langle \mathsf{LSM}_2 \rangle_\#$? There is an analogous characterisation for the setting of VCSPs [106]. Other questions concerning the structure of LSM are asked in [22].

Appendix A

Index of definitions

These lists do not include anything that solely appears within a proof, or within a short discussion.

A.1 Notation

$\{0,1\}^{k+J}$	Section 3.2
$\mathbf{R}$ (boldface)	Section 3.2
$[\![\varphi]\!]$ (circuits)	Section 3.2.1
$[\![\varphi]\!]$ (K-formulas)	Section 5.2.1
$[\![G]\!]$	Section 3.7
$\langle\mathcal{F}\rangle_{\#}$	Section 6.1.3
$\langle\Phi\rangle_V$	Section 6.4
$F(\mathbf{p},\cdot)$	Section 1.7.2
FG	Section 1.7.2
$T\mathcal{F}$	Section 4.3
$T\mathcal{F}^1$, $T\mathcal{F}^2$	Section 5.4
$F\times G$	Section 1.7.1
$F\otimes G$	Section 5.1.6
$F^{\otimes J}$	Section 1.7.2
$\overline{\mathbf{x}}$, $F\overline{F}$, $F^\star$	Section 1.7.2
e^*	Section 2.3.5
$\oplus$	Section 1.7.2
$F_\oplus$	Section 3.4
$\triangle$ (for sets)	symmetric difference of sets
$\triangle$ (for configurations)	Definition 5.1
$\wedge$	Section 1.5.5
$\vee$	Section 1.5.5
$\{i\mapsto c\}$	Section 5.1.6
$\mathbf{x}[\sigma_t]$	Section 6.4

$I[\varphi/F]$	Definition 5.19
$[0,1]_{\mathbb{Q}}$	Section 6.4
$\leq_{\mathrm{AP}}$, $=_{\mathrm{AP}}$	Section 1.2
$\#\mathsf{BIS}$	Section 1.2.1
$\#\mathsf{CSP}(\mathcal{F})$	Section 1.3, Section 6.1.3 (with pps-formulas)
$\#\mathsf{CSP}_{\leq d}(\Gamma)$	Section 1.7.2
$\#\mathsf{CSP}_{=2}(\Gamma)$	Section 4.1
$\#\mathsf{CSP}_{\leq 2}^{\geq 0}(\Gamma)$	Section 1.6
$\#\mathsf{CSP}_K^W(\mathcal{F})$, $\#\mathsf{CSP}_{=2}^{\geq 0}(\mathcal{F})$, etc	Section 5.1.3 (generalises $\#\mathsf{CSP}_{=2}$, $\#\mathsf{CSP}_{\leq d}$, and $\#\mathsf{CSP}_{\leq 2}^{\geq 0}$)
$\#\mathsf{Downsets}$	Section 4.5 (poset variant in Example 1.1)
$\#\mathsf{FugacityWeightedPM}$	Section 3.7.2
$\#\mathsf{P}$	Section 1.2
$\#\mathsf{ParityNAE}$	Section 1.6
$\#\mathsf{PM}$	Section 1.2.1
$\#\mathsf{SAT}$	Section 1.2.1
$\Lambda(\mathcal{F})$	before the statement of Theorem 6.43
π	Section 3.3.5
$\rho(\Gamma)$	Section 3.3.5
$\boldsymbol{\sigma}_\nu$	Section 2.3.1
$\theta(F)$	Section 3.5.2
$\mu_{k,d}$, $\mu_{k,d}^0$, $\mu_{k,d}^1$	Section 2.3.4
Ω, Ω_k	Section 3.3.5
$\mathcal{A}$	Section 4.3
$\mathrm{AtMostOne}_3$	statement of Theorem 5.17
$\mathcal{B}_x(\sigma)$	Section 2.3.7
$b(\sigma)$	Section 2.1
$B_{x,m}$	Section 2.3.1
BPP	see [69, Chapter 7]
$\mathbb{C}$	Section 1.7.2
$\mathcal{C}$	Section 7.3
$c(\sigma)$	Section 2.1
C_ν	Section 2.3
$C_{k,d}$	Section 2.3.4
C_ν^*	Section 2.3.5
$\deg_C(v)$	Section 1.7.2
$\mathrm{d}(\mathbf{x},\mathbf{y})$	Section 3.2
$\mathsf{DegreeFourPlanarHardCore}$	Section 2.2
$\mathsf{DegreeFourPlanarTwoSpin}(\beta,\gamma,\lambda)$	Section 2.2
dom	Definition 5.9

$p^{=}$,$p^{\neq}$	Section 2.3.3
PIN_0, PIN_1	Section 1.7.2
$\mathsf{PlanarCubicIS}$	Section 2.4
$\mathsf{PlanarLogTwoSpin}(\beta,\gamma,\lambda)$	Section 2.2
PM_3	Section 1.7.2
$\mathbb{Q}$, $\mathbb{Q}_{\geq 0}$	Section 1.7.2
$\mathbb{R}$, $\mathbb{R}_{\geq 0}$	Section 1.7.2
$\overline{\mathbb{R}}_{\geq 0}$	Section 6.4
$R(g)$	Section 2.3.5
RP	see [69, Chapter 7]
$\mathrm{supp}(F)$	before Theorem 5.10
$T_{k,d}$,$T^1_{k,d}$,$T^0_{k,d}$	Section 2.3.4
$\mathcal{U}_D$	Definition 6.1
$U_{x,h}$	Section 2.3.7
WNEQ	Section 5.7
$\mathrm{wt}_{V,C}(\mathbf{x})$	Section 1.3
$\mathrm{wt}_{V,C,w}(\mathbf{x})$	Section 5.1.3
$\mathrm{wt}_{\varphi}(\mathbf{x})$	Section 3.2.1
$\mathrm{wt}_G(\mathbf{x})$	Section 3.7
$\mathbb{Z}$	Section 1.7.2
$Z_{V,C}$	Section 1.3
$Z_{V,C,w}$	Section 5.1.3
$Z_{\beta,\gamma,\lambda}$	Section 2.1
$Z_k(\varphi)$	Section 3.2.1

A.2 Terminology

#P-complete	Section 1.2
$*$-adjacent, $*$-distance, $*$-path	Section 2.3.7
$*$-diameter	Section 2.3.7
distance	Section 3.2
2-decomposition	Section 3.3
adjacent (to a contour)	Section 2.3.5
affine relation	Section 1.7.2
affine signature	Section 4.3
AP-reduction	Section 1.2
assignment	Section 3.2.1
(k-)assignment	Section 3.2.1
balanced	Section 6.3
basically binary	Definition 5.2

FPRAS	Section 1.2
FPTAS	deterministic FPRAS; mentioned in Section 2.1.1
functional clone	Section 6.1.3
generalised Fibonacci signature	Section 4.3
generalised STP multimorphism	Definition 6.31
graph fragment	Section 3.2.1
hard	Section 6.1.3
indecomposable	Section 5.1.6
(K) intersects $(U_{x,h})$	Section 2.3.7
join	Section 1.5.5
length (of a contour)	Section 2.3.5
level	Section 2.5
local	Section 2.3.7
log-anti-Monge	before the statement of Theorem 6.42
log-supermodular	Section 1.5.5
majority polymorphism	before statement of Theorem 6.28
matchings circuit	Section 3.7
(h-)maximisation	Definition 5.22
meet	Section 1.5.5
MJN	Definition 6.24
Monge	before the statement of Theorem 6.42
multimorphism	Definition 6.21, Definition 6.22
operation	Definition 6.19
outerplanar	see [7]
orthogonal	Section 4.3
parallel	Definition 5.9
parity relations	Section 3.2
parity-0 and parity-1 ones	Section 2.3.2
parity-0 and parity-1 terminals	Section 2.3.4
partial configuration	Definition 5.9
partition function	Section 1.1
phase	Definition 2.16
pinning	Section 1.7.2
pinning-minimal	Section 5.1.6
POTM	Section 1.2
pps-formula	Section 6.1.3
functional clone	Section 6.1.3
pps_ω-definable functional clone	introduction to Chapter 7
PRAS	Section 2.2
psm-formula	Section 6.4

Bibliography

[1] N. Alon, *Combinatorial Nullstellensatz*, Combin. Probab. Comput. **8** (1999), no. 1-2, 7–29, Recent trends in combinatorics (Mátraháza, 1995).

[2] Brenda S. Baker, *Approximation algorithms for NP-complete problems on planar graphs*, J. ACM **41** (1994), 153–180.

[3] Antar Bandyopadhyay and David Gamarnik, *Counting without sampling: Asymptotics of the log-partition function for certain statistical physics models*, Random Struct. Algorithms **33** (2008), no. 4, 452–479.

[4] David Bartl, *A very short algebraic proof of the Farkas Lemma*, Math. Methods Oper. Res. **75** (2012), no. 1, 101–104.

[5] J. van den Berg and J.E. Steif, *Percolation and the hard-core lattice gas model*, Stochastic Processes and their Applications **49** (1994), no. 2, 179–197.

[6] Antonio Blanca, David Galvin, Prasad Tetali, and Dana Randall, *Phase Coexistence and Slow Mixing for the Hard-Core Model on Z$\hat{2}$*, arXiv:1211.6182 [math.CO], 2012.

[7] Hans L. Bodlaender, *A partial k-arboretum of graphs with bounded treewidth*, Theoret. Comput. Sci. **209** (1998), no. 1-2, 1–45.

[8] Magnus Bordewich, *On the approximation complexity hierarchy*, Approximation and Online Algorithms (Klaus Jansen and Roberto Solis-Oba, eds.), Lecture Notes in Computer Science, vol. 6534, Springer Berlin Heidelberg, 2011, pp. 37–46.

[9] Magnus Bordewich, Martin Dyer, and Marek Karpinski, *Path coupling using stopping times and counting independent sets and colorings in hypergraphs*, Random Structures & Algorithms **32** (2008), no. 3, 375–399.

[10] Christian Borgs, Jennifer T. Chayes, Jeong Han Kim, Alan Frieze, Prasad Tetali, Eric Vigoda, and Van Ha Vu, *Torpid mixing of some Monte Carlo Markov chain algorithms in statistical physics*, Proceedings of the 40th Annual Symposium on Foundations of Computer Science (Washington, DC, USA), FOCS '99, IEEE Computer Society, 1999, pp. 218–229.

[11] E. Boros and P. L. Hammer, *Pseudo-Boolean optimization*, Discrete Applied Mathematics **123** (2002), no. 1–3, 155–225.

[12] Glencora Borradaile, *Baker's technique: Designing approximation schemes for planar graphs*, http://teach.glencora.org/index.php?title=Baker's_technique:_Designing_approximation_schemes_for_planar_graphs.

[13] André Bouchet and William H. Cunningham, *Delta-matroids, jump systems, and bisubmodular polyhedra*, SIAM J. Discrete Math. **8** (1995), no. 1, 17–32.

[14] A. Z. Broder, *How hard is it to marry at random? (on the approximation of the permanent)*, Proceedings of the eighteenth annual ACM symposium on Theory of computing (New York, NY, USA), STOC '86, ACM, 1986, pp. 50–58.

[15] R. Bubley and M. Dyer, *Graph orientations with no sink and an approximation for a hard case of #SAT*, Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (New Orleans, LA, 1997) (New York), ACM, 1997, pp. 248–257.

[16] ———, *Path coupling: A technique for proving rapid mixing in markov chains*, Proceedings of the 38th Annual Symposium on Foundations of Computer Science (Washington, DC, USA), IEEE Computer Society, 1997, pp. 223–.

[17] A. A. Bulatov, *Tractable conservative constraint satisfaction problems*, Proceedings of 18th Annual IEEE Symposium on Logic in Computer Science (LICS), IEEE, 2003, pp. 321–330.

[18] ———, *A dichotomy theorem for constraint satisfaction problems on a 3-element set*, J. ACM **53** (2006), no. 1, 66–120.

[19] ———, *The complexity of the counting constraint satisfaction problem*, Proceedings of 35th International Colloquium on Automata, Languages and Programming (ICALP) (Part I), LNCS, vol. 5125, Springer, 2008, pp. 646–661.

[20] A. A. Bulatov, M. E. Dyer, L. A. Goldberg, M. Jalsenius, M. R. Jerrum, and D. M. Richerby, *The complexity of weighted and unweighted #CSP*, Journal of Computer and System Sciences **78** (2012), no. 2, 681–688.

[21] A. A. Bulatov, M. E. Dyer, L. A. Goldberg, M. Jalsenius, and D. M. Richerby, *The complexity of weighted Boolean #CSP with mixed signs*, Theoretical Computer Science **410** (2009), no. 38–40, 3949–3961.

[22] Andrei A. Bulatov, Martin E. Dyer, Leslie Ann Goldberg, Mark Jerrum, and Colin McQuillan, *The expressibility of functions on the Boolean domain, with applications to Counting CSPs*, J. ACM (2013), To appear; preprint at arXiv:1108.5288 [cs.CC].

[23] J.-Y. Cai and X. Chen, *Complexity of counting CSP with complex weights*, Proceedings of 44th ACM Symposium on Theory of Computing (STOC), ACM, 2012, pp. 909–920.

[24] J.-Y. Cai, X. Chen, and P. Lu, *Non-negative weighted #CSPs: An effective complexity dichotomy*, arXiv:1012.5659 [cs.CC]; although this is unpublished, a short version appeared in [25], 2010.

[25] ———, *Non-negatively weighted #CSP: An effective complexity dichotomy*, Proceedings of IEEE Conference on Computational Complexity, 2011, pp. 45–54.

[26] J.-Y. Cai, P. Lu, and M. Xia, *Holant problems and counting CSP*, STOC, 2009, pp. 715–724.

[27] ———, *Dichotomy for Holant* problems of Boolean domain*, Proceedings of 22nd ACM–SIAM Symposium on Discrete Algorithms (SODA), 2011, pp. 1714–1728.

[28] Jin-Yi Cai and Vinay Choudhary, *Some results on matchgates and holographic algorithms*, Automata, languages and programming. Part I, Lecture Notes in Comput. Sci., vol. 4051, Springer, Berlin, 2006, pp. 703–714.

[29] Jin-Yi Cai, Sangxia Huang, and Pinyan Lu, *From Holant to #CSP and Back: Dichotomy for Holant^c Problems*, Algorithmica **64** (2012), 511–533, 10.1007/s00453-012-9626-6.

[30] Jin-Yi Cai and Michael Kowalczyk, *Spin systems on graphs with complex edge functions and specified degree regularities*, Computing and combinatorics, Lecture Notes in Comput. Sci., vol. 6842, Springer, Heidelberg, 2011, pp. 146–157.

[31] Jin-Yi Cai, Pinyan Lu, Heng Guo, and Tyson Williams, *A complete dichotomy rises from the capture of vanishing signatures*, STOC, 2013, To appear.

[32] Jin-Yi Cai, Pinyan Lu, and Mingji Xia, *Holographic algorithms by Fibonacci gates and holographic reductions for hardness*, Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science (Washington, DC, USA), FOCS '08, IEEE Computer Society, 2008, pp. 644–653.

[33] X. Chen, *Guest column: Complexity dichotomies of counting problems*, SIGACT News **42** (2011), no. 4, 54–76.

[34] Xi Chen, Martin Dyer, Leslie Ann Goldberg, Mark Jerrum, Pinyan Lu, and Colin McQuillan David Richerby, *The complexity of approximating conservative counting CSPs*, STACS, 2013, full version at arXiv:1208.1783 [cs.CC], pp. 148–159.

[35] D. A. Cohen, M. C. Cooper, and P. G. Jeavons, *Generalising submodularity and Horn clauses: Tractable optimization problems defined by tournament pair multimorphisms*, Theoretical Computer Science **401** (2008), no. 1–3, 36–51.

[36] D. A. Cohen, M. C. Cooper, P. G. Jeavons, and A. A. Krokhin, *Soft constraints: Complexity and multimorphisms*, Proceedings of 9th International Conference on Principles of Constraint Programming (CP2003), LNCS, vol. 2833, Springer, 2003, pp. 244–258.

[37] D.A. Cohen, P. Creed, P.G. Jeavons, and S. Živný, *An algebraic theory of complexity for valued constraints: Establishing a Galois connection*, Proceedings of the 36th International Symposium on Mathematical Foundations of Computer Science (MFCS'11), Lecture Notes in Computer Science, vol. 6907, Springer, 2011, pp. 231–242.

[38] Henry Cohn, Robin Pemantle, and James Propp, *Generating a random sink-free orientation in quadratic time*, Electron. J. Combin. **9** (2002), no. 1, Research Paper 10, 13 pp. (electronic).

[39] Gérard Cornuéjols, *General factors of graphs*, J. Combin. Theory Ser. B **45** (1988), no. 2, 185–198.

[40] Nadia Creignou and Miki Hermann, *Complexity of generalized satisfiability counting problems*, Information and Computation **125** (1996), no. 1, 1–12.

[41] Víctor Dalmau and Daniel K. Ford, *Generalized Satisfability with Limited Occurrences per Variable: A Study through Delta-Matroid Parity*, MFCS, 2003, pp. 358–367.

[42] Joan Davies and Colin McDiarmid, *Disjoint common transversals and exchange structures*, Journal of the London Mathematical Society **s2-14** (1976), no. 1, 55–62.

[43] Ronald de Wolf, *A Brief Introduction to Fourier Analysis on the Boolean Cube*, Graduate Surveys, no. 1, Theory of Computing Library, 2008.

[44] Persi Diaconis and Daniel Stroock, *Geometric bounds for eigenvalues of Markov chains*, Ann. Appl. Probab. **1** (1991), no. 1, 36–61.

[45] R.L. Dobrushin, *The problem of uniqueness of a Gibbsian random field and the problem of phase transitions*, Functional Analysis and Its Applications **2** (1968), 302–312 (English).

[46] M. E. Dyer, L. A. Goldberg, C. Greenhill, and M. R. Jerrum, *The relative complexity of approximate counting problems*, Algorithmica **38** (2004), no. 3, 471–500.

[47] M. E. Dyer, L. A. Goldberg, and M. R. Jerrum, *The complexity of weighted Boolean CSP*, SIAM Journal on Computing **38** (2009), no. 5, 1970–1986.

[48] ———, *An approximation trichotomy for Boolean #CSP*, Journal of Computer and System Sciences **76** (2010), no. 3–4, 267–277.

[49] M. E. Dyer and D. M. Richerby, *On the complexity of #CSP*, Proceedings of 42nd ACM Symposium on Theory of Computing (STOC), 2010, pp. 725–734.

[50] ———, *An effective dichotomy for the counting constraint satisfaction problem*, SIAM Journal on Computing (to appear), arXiv:1003.3879 [cs.CC].

[51] Martin E. Dyer, Leslie Ann Goldberg, Markus Jalsenius, and David Richerby, *The complexity of approximating bounded-degree Boolean #CSP*, STACS, 2010, pp. 323–334.

[52] Tomás Feder, *Fanout limitations on constraint systems*, Theor. Comput. Sci. **255** (2001), no. 1-2, 281–293.

[53] Michael E. Fisher, *On the Dimer Solution of Planar Ising Models*, Journal of Mathematical Physics **7** (1966), no. 10, 1776–1781.

[54] C. M. Fortuin, P. W. Kasteleyn, and J. Ginibre, *Correlation inequalities on some partially ordered sets*, Comm. Math. Phys. **22** (1971), 89–103.

[55] Andreas Galanis, Qi Ge, Daniel Štefankovič, Eric Vigoda, and Linji Yang, *Improved inapproximability results for counting independent sets in the hard-core model*, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Lecture Notes in Computer Science, vol. 6845, Springer Berlin / Heidelberg, 2011, pp. 567–578.

[56] Andreas Galanis, Daniel Štefankovič, and Eric Vigoda, *Inapproximability of the Partition Function for the Antiferromagnetic Ising and Hard-Core Models*, arXiv:1203.2226 [cs.DM], 2012.

[57] Anna Gambin, *On approximating the number of bases of exchange preserving matroids*, Mathematical foundations of computer science 1999 (Szklarska Poręba), Lecture Notes in Comput. Sci., vol. 1672, Springer, Berlin, 1999, pp. 332–342.

[58] M. R. Garey, David S. Johnson, and Larry J. Stockmeyer, *Some simplified NP-complete graph problems*, Theor. Comput. Sci. **1** (1976), no. 3, 237–267.

[59] Leslie Ann Goldberg and Mark Jerrum, *Inapproximability of the Tutte polynomial*, Information and Computation **206** (2008), no. 7, 908–929.

[60] ———, *Approximating the partition function of the ferromagnetic Potts model*, Proceedings of 37th International Colloquium on Automata, Languages and Programming (ICALP) (Part I), LNCS, vol. 6198, Springer, 2010, pp. 396–407.

[61] ———, *Approximating the partition function of the ferromagnetic Potts model*, J. ACM **59** (2012), no. 5, 1–25.

[62] ———, *Inapproximability of the Tutte polynomial of a planar graph*, Computational Complexity **21** (2012), no. 4, 605–642.

[63] Leslie Ann Goldberg, Mark Jerrum, and Colin McQuillan, *Approximating the partition function of planar two-state spin systems*, arXiv:1208.4987 [cs.CC], submitted.

[64] Leslie Ann Goldberg, Mark Jerrum, and Mike Paterson, *The computational complexity of two-state spin systems*, Random Struct. Algorithms **23** (2003), no. 2, 133–154.

[65] Catherine S. Greenhill, *Making Markov chains less lazy*, arXiv:1203.6668v2 [math.CO], 2012.

[66] H. Guo and T. Williams, *The Complexity of Planar Boolean #CSP with Complex Weights*, arXiv:1212.2284 [cs.CC], 2012.

[67] Sangxia Huang and Pinyan Lu, *A Dichotomy for Real Weighted Holant Problems*, CCC (2012), 96–106.

[68] E. T. Jaynes, *Information theory and statistical mechanics*, Phys. Rev. (2) **106** (1957), 620–630.

[69] Mark Jerrum, *Counting, sampling and integrating: algorithms and complexity*, Lectures in Mathematics ETH Zürich, Birkhäuser Verlag, Basel, 2003.

[70] Mark Jerrum and Alistair Sinclair, *Approximating the permanent*, SIAM J. Comput. **18** (1989), no. 6, 1149–1178.

[71] ______, *Polynomial-time approximation algorithms for the Ising model*, SIAM J. Comput. **22** (1993), no. 5, 1087–1116.

[72] Mark Jerrum, Alistair Sinclair, and Eric Vigoda, *A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries*, J. ACM **51** (2004), no. 4, 671–697.

[73] Mark Jerrum, Leslie G. Valiant, and Vijay Vazirani, *Random generation of combinatorial structures from a uniform distribution*, Theoret. Comput. Sci. **43** (1986), no. 2–3, 169–188.

[74] Richard M. Karp and Michael Luby, *Monte-Carlo algorithms for the planar multiterminal network reliability problem*, Journal of Complexity **1** (1985), no. 1, 45–64.

[75] Ross Kindermann and J. Laurie Snell, *Markov random fields and their applications*, Contemporary Mathematics, vol. 1, American Mathematical Society, Providence, R.I., 1980.

[76] Ker-I Ko, *Some observations on the probabilistic algorithms and NP-hard problems*, Inform. Process. Lett. **14** (1982), no. 1, 39–43.

[77] V. Kolmogorov and S. Živný, *The complexity of conservative valued CSPs*, arXiv:1110.2809 [cs.CC]; although this is unpublished, a short version appeared in [78], 2011.

[78] ———, *The complexity of conservative valued CSPs*, Proceedings of 23rd ACM–SIAM Symposium on Discrete Algorithms (SODA), 2012, pp. 750–759.

[79] Liang Li, Pinyan Lu, and Yitong Yin, *Correlation decay up to uniqueness in spin systems*, SODA, 2013, pp. 67–84.

[80] Richard J. Lipton and Robert Endre Tarjan, *Applications of a planar separator theorem*, SIAM J. Comput. **9** (1980), no. 3, 615–627.

[81] P. Lu, *Complexity dichotomies of counting problems*, Electronic Colloquium on Computational Complexity (ECCC) **18** (2011), 93.

[82] Yongyi Mao and Frank R. Kschischang, *On factor graphs and the Fourier transform*, IEEE Transactions on Information Theory **51** (2005), no. 5, 1635–1649.

[83] Colin McQuillan, *Degree two approximate Boolean #CSPs with variable weights*, arXiv:1204.5714 [cs.CC], submitted, 2012.

[84] ———, *Approximating Holant problems by winding*, arXiv:1301.2880 [cs.CC], submitted, 2013.

[85] J. Provan and M. Ball, *The complexity of counting cuts and of computing the probability that a graph is connected*, SIAM Journal on Computing **12** (1983), no. 4, 777–788.

[86] Dana Randall, *Slow mixing of glauber dynamics via topological obstructions*, Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm (New York, NY, USA), SODA '06, ACM, 2006, pp. 870–879.

[87] Ricardo Restrepo, Jinwoo Shin, Prasad Tetali, Eric Vigoda, and Linji Yang, *Improved mixing condition on the grid for counting and sampling independent sets*, Probability Theory and Related Fields (2012), 1–25.

[88] R. Rudolf and G. J. Woeginger, *The cone of Monge matrices: extremal rays and applications*, Mathematical Methods of Operations Research **42** (1995), no. 2, 161–168.

[89] Thomas J. Schaefer, *The complexity of satisfiability problems*, Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, Calif., 1978), ACM, New York, 1978, pp. 216–226.

[90] Alistair Sinclair, *Improved bounds for mixing rates of Markov chains and multicommodity flow*, Combin. Probab. Comput. **1** (1992), no. 4, 351–370.

[91] Alistair Sinclair, Piyush Srivastava, and Marc Thurley, *Approximation algorithms for two-state anti-ferromagnetic spin systems on bounded degree graphs*, Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '12, SIAM, 2012, pp. 941–953.

[92] Allan Sly, *Computational transition at the uniqueness threshold*, FOCS, 2010, pp. 287–296.

[93] Allan Sly and Nike Sun, *The computational hardness of counting in two-spin models on d-regular graphs*, FOCS, 2012, pp. 361–369.

[94] R. Takhanov, *A dichotomy theorem for the general minimum cost homomorphism problem*, arXiv:0708.3226 [cs.LG]; although this is unpublished, a short version appeared in [95].

[95] ______, *A dichotomy theorem for the general minimum cost homomorphism problem*, Proceedings of 27th International Symposium on Theoretical Aspects of Computer Science (STACS), LIPIcs, vol. 5, Liebniz-Zentrum für Informatik, 2010, pp. 657–668.

[96] D. M. Topkis, *Minimizing a submodular function on a lattice*, Operations Research **26** (1978), 305–321.

[97] Leslie G. Valiant, *The complexity of computing the permanent*, Theoret. Comput. Sci. **8** (1979), no. 2, 189–201.

[98] ______, *Accidental algorithms*, FOCS, 2006, pp. 509–517.

[99] ______, *Holographic algorithms*, SIAM Journal on Computing **37** (2008), no. 5, 1565–1594.

[100] Dror Weitz, *Counting independent sets up to the tree threshold*, Proceedings of the thirty-eighth annual ACM symposium on Theory of computing (New York, NY, USA), STOC '06, ACM, 2006, pp. 140–149.

[101] Tomoyuki Yamakami, *A dichotomy theorem for the approximate counting of complex-weighted bounded-degree Boolean CSPs*, Theor. Comput. Sci. **447** (2012), 120–135.

[102] ______, *Approximate counting for complex-weighted Boolean constraint satisfaction problems*, Inf. Comput. **219** (2012), 17–38.

[103] ______, *Approximation complexity of complex-weighted degree-two counting constraint satisfaction problems*, Theor. Comput. Sci. **461** (2012), 86–105.

[104] Y. Yin and C. Zhang, *Approximate Counting via Correlation Decay on Planar Graphs*, SODA, 2013, pp. 47–66.

[105] Jinshan Zhang, Heng Liang, and Fengshan Bai, *Approximating partition functions of the two-state spin system*, Information Processing Letters **111** (2011), no. 14, 702–710.

[106] Stanislav Živný, David A. Cohen, and Peter G. Jeavons, *The Expressive Power of Binary Submodular Functions*, Discrete Applied Mathematics **157** (2009), no. 15, 3347–3358.