Counting Constraint Satisfaction Problems*

Mark Jerrum

School of Mathematical Sciences, Queen Mary, University of London, London, UK
m.jerrum@qmul.ac.uk

Abstract

This chapter surveys counting Constraint Satisfaction Problems (counting CSPs, or #CSPs) and their computational complexity. It aims to provide an introduction to the main concepts and techniques, and present a representative selection of results and open problems. It does not cover holants, which are the subject of a separate chapter.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Approximation algorithms, Computational complexity, Constraint satisfaction problems, Counting problems, Partition functions

Digital Object Identifier 10.4230/DFU.Vol7.15301.205

1 Introduction

In this chapter, we shall be working within the usual CSP framework, or natural extensions of it, but our interest will be in *counting* assignments that satisfy all the constraints, rather than just determining whether one exists. We can make a swift entry into the topic by minimally adapting the classical CSP framework. Having done this, we shall briefly discuss the limitations of this simple-minded approach, and how the model can be refined to encompass a wider range of situations.

A Constraint Satisfaction Problem (CSP) typically has a finite *domain* D, which we identify with $\{0,\dots,q-1\}$, or possibly $\{1,\dots,q\}$, for some positive integer q. Keeping close to classical decision CSPs, a *constraint language* Γ is a set of relations of various arities on D. Given a finite constraint language Γ, an *instance* of #CSP(Γ), the counting CSP with constraint language Γ, is specified by a set of *variables* $X=\{x_1,\dots,x_n\}$ and a set $C=\{(R_1,\mathbf{x}_1),\dots,(R_m,\mathbf{x}_m)\}$ of *constraints*. Each constraint is a pair: a relation $R_i\in\Gamma$ of some arity k, and a *scope* $\mathbf{x}_i=(x_{s_{i,1}},\dots,x_{s_{i,k}})$, which is a k-tuple of variables from X. An *assignment* σ is a mapping from X to D. The assignment σ is said to be *satisfying*, or to *satisfy the instance* (X,C), if the scope of every constraint is mapped to a tuple that is in the corresponding relation, that is to say, σ satisfies the formula $\bigwedge_{i=1}^{m}(\sigma(\mathbf{x}_i)\in R_i)$, where $\sigma(\mathbf{x}_i)=(\sigma(x_{s_{i,1}}),\dots,\sigma(x_{s_{i,k}}))$.

Given an instance (X,C) of a CSP with constraint language Γ, the *decision problem* CSP(Γ) asks us to determine whether any assignment σ exists that satisfies (X,C). The *counting problem* #CSP(Γ) asks us to determine the *number* of assignments that satisfy (X,C).

By varying the constraint language Γ we obtain infinite families of decision problems CSP(Γ) and counting problems #CSP(Γ). We wish to classify these problems according to their computational complexity. A simple observation is that the counting CSP cannot be easier than its decision twin, but can be harder. For example, consider the binary relation on

* This work was partially supported by the EPSRC grant EP/N004221/1.

The Constraint Satisfaction Problem: Complexity and Approximability. *Dagstuhl Follow-Ups*, Volume 7, ISBN 978-3-95977-003-3.
Editors: Andrei Krokhin and Stanislav Živný; pp. 205–231

Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

$\{0,1\}$ defined by NAND $= \{(0,0),(0,1),(1,0)\}$, and the constraint language $\Gamma = \{\text{NAND}\}$ consisting of this single relation. Since the relation NAND is symmetric, we can view an instance of CSP(Γ) or #CSP(Γ) as an undirected graph G whose vertices represent variables and whose edges represent scopes. The decision problem CSP(Γ) asks whether G contains an independent set; this decision problem is trivial, since every graph has the empty set of vertices as an independent set. In contrast, #CSP(Γ) is the problem of counting independent sets in G, which is #P-complete. As we shall see, even estimating the number of independent sets in G within relative error $1 \pm \varepsilon$ is computational intractable, assuming RP $\neq$ NP.

Many of the motivating examples for counting CSPs come from statistical physics. Variables represent states of atoms, say, and constraints represent interactions between pairs, triples, etc., of atoms. These interactions are not usually "hard" constraints that can be modelled by relations. Because of this, even more than in the case of valued CSPs (VCSPs), there is a strong motivation to extend the above setting for counting CSPs to include weights. With this in mind, we replace the set Γ of relations by a set of functions $\mathcal{F}$. Each function $f \in \mathcal{F}$ is of the form $f : D^k \to \mathbf{R}$, where $\mathbf{R}$ is a commutative semiring: common choices for $\mathbf{R}$ are $\mathbb{C}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, or some computationally tractable subrings of these, such as $\mathbb{Q}$ or the algebraic reals. An instance of a (weighted) counting CSP is specified by a set of variables X and a collection of functional "constraints" $\big\{(f_1, \mathbf{x}_1), \ldots, (f_m, \mathbf{x}_m)\big\}$, where $f_i \in \mathcal{F}$ is a function of arity k, and $\mathbf{x}_i = (x_{s_{i,1}}, \ldots, x_{s_{i,k}})$ is a scope. Then the required output for the counting CSP is the quantity

$$Z(X, C) = \sum_{\sigma : X \to D} \prod_{i=1}^{m} f_i(\sigma(\mathbf{x}_i)).$$

Relative to usual decision CSPs we have replaced relations by functions, conjunction by multiplication and existential quantification by summation. This is a strict extension of classical decision CSPs, which we can recover by setting $\mathbf{R}$ to be the semiring $(\{0,1\}, \vee, \wedge)$. Even VCSPs can be viewed in this light by taking $\mathbf{R} = (\mathbb{R}_{\geq 0}, \min, +)$. However, although some techniques can be traded between these CSP variants, they maintain strong identities of their own.

There is already an extensive body of work on counting CSPs, and it is not possible to cover all of it here. The most significant omission is surely holants, which are, roughly speaking, read-twice counting CSPs. Holants generalise counting CSPs; for example, the generating function of perfect matchings in a graph (the dimer model in statistical physics) can be expressed as a holant, but not as a counting CSP. Holants have a special flavour and an extensive literature of their own, and are the subject of a separate chapter in this volume.

In deciding how to organise the material in this survey, a number of different attributes can be taken into account. CSPs on a two-element domain have a special prominence and are often easier to deal with. The same is true of conservative CSPs in which unary functions are given free. Bijunctive CSPs (all functions in $\mathcal{F}$ have arity at most 2), and more specifically CSPs with one symmetric binary function are not only potentially easier to handle, but have a particular importance because of their connection with spin models in statistical physics. However, it seems to me that the clearest division in terms of the flavour of the results and the techniques employed is between exact and approximate computation. I have therefore decided to make this the high level split. Within these two parts, the material will be organised as far as is possible into special cases, such as Boolean, conservative, partition functions, etc.

2 Exact Computation

A combinatorial (i.e., unweighted) counting problem specifies a function $\Sigma^* \to \mathbb{N}$ that maps problem instances, encoded over an alphabet Σ, to natural numbers. A function $f : \Sigma^* \to \mathbb{N}$ is in the complexity class #P if there exists a nondeterministic Turing machine M such that, for all $x \in \Sigma^*$, the number of accepting computations of M on input $x \in \Sigma^*$ is equal to $f(x)$. The relative complexity of counting problems is customarily investigated using Turing reductions. Thus, a problem f is said to be #P-*hard* if every problem $g \in$ #P is polynomial-time Turing reducible to f. If, in addition, $f \in$ #P, then the problem f is said to be #P-*complete*. It is clear that #CSP(Γ) is in #P for any finite set of relations Γ, since we can construct a Turing machine M that, given an instance (X, C) of #CSP(Γ), nondeterministically chooses an assignment $\sigma : X \to D$ to the variables X, and accepts if all constraints C are satisfied by σ. The complexity class #P was introduced by Valiant, who also made the initial exploration of the phenomenon of #P-completeness [54].

To make sense of a counting CSP with real or complex weights as a computational problem, we need to restrict the real or complex numbers to some suitable subfield, say rational, algebraic or polynomial-time computable. A weighted counting CSP is no longer a member of #P, for the banal reason that it does not in general produce integer outputs. Nevertheless, at least if we restrict attention to finite sets of functions $\mathcal{F}$ taking rational or algebraic values, it will be the case that #CSP($\mathcal{F}$) $\in \mathsf{FP}^{\#\mathsf{P}}$, i.e., that #CSP($\mathcal{F}$) is solvable by a polynomial-time deterministic Turing machine with a #P-oracle. Also, we can still expect #CSP($\mathcal{F}$) to be #P-hard in many cases, thus locating the computational complexity up to polynomial-time Turing reductions. For some comments on the complexity of counting problems with rational weights (in the context of the Tutte polynomial) see [37]. We will ignore the issue of representing real and complex numbers in the remainder of the survey.

2.1 Boolean #CSPs

Just as with decision CSPs, the first step historically in the exploration of counting CSPs was the resolution of the Boolean case. This was achieved by Creignou and Hermann [15]. It turns out to be helpful to identify the 2-element domain D with the 2-element field $\mathbb{F}_2$. Then we can say that a relation $R \subseteq \{0,1\}^k$ is *affine* if and only if it is the solution set to a system of linear equations over $\mathbb{F}_2$.

▶ **Theorem 1.** *Let Γ be a finite set of relations on the domain $\{0,1\}$. If every relation in Γ is affine then* #CSP(Γ) *is in* FP*; otherwise* #CSP(Γ) *is* #P*-complete.*

This result is pessimistic when compared to Schaefer's dichotomy for classical (decision) CSPs. Recall that a relation is bijunctive if it is equivalent to a conjunction of clauses with at most two literals, 0-valid (respectively, 1-valid) if it is empty or contains the all-0 (respectively, all-1) tuple, and Horn (respectively, dual-Horn) if it is equivalent to a conjunction of Horn (respectively, dual-Horn) clauses. Any of the conditions affine, bijunctive, 0/1-valid, Horn or dual-Horn is sufficient to ensure tractability of the decision problem. For counting, only affine will do. The constraint language $\Gamma = \{\mathrm{IMP}\}$ consisting solely of the implies relation $\mathrm{IMP} = \{(0,0),(0,1),(1,1)\}$ neatly illustrates the point: IMP is bijunctive, 0-valid, 1-valid, Horn and dual-Horn and yet #CSP(Γ) is #P-complete, being essentially equivalent to counting downsets in a partial order [51].

In broad brushstrokes, Creignou and Hermann's proof of Theorem 1 runs as follows. Denote by OR the relation $\mathrm{OR} = \{(0,1),(1,0),(1,1)\}$. If Γ is affine then any instance of #CSP(Γ) defines an affine relation. Thus the set of solutions to the instance (X, C) forms

an affine subspace of $\mathbb{F}_2^X$, whose dimension d can be computed by standard linear algebra techniques. The required output for the instance is then 2^d. If Γ is not affine, then one of the relations NAND, OR or IMP can (in a suitable sense) be implemented in terms of relations in Γ. Since all of #CSP($\{$NAND$\}$), #CSP($\{$OR$\}$) and #CSP($\{$IMP$\}$) are #P-complete, this completes the proof. It is an interesting exercise to translate the proof into the language of clones and Post's lattice, and also an instructive one, as it hints at what survives of the universal algebra approach, and what does not, in the passage from decision to counting. We'll sketch how this works once suitable concepts and notation have been introduced.

The next step is to add positive real weights. So now $\mathcal{F}$ is a finite set of functions of the form $\{0,1\}^k \to \mathbb{R}_{\geq 0}$ (with the arity k possibly varying), and we are interested in the complexity of #CSP($\mathcal{F}$). A dichotomy similar to Theorem 1 continues to hold. Denote by $\mathcal{P}$ the set of all functions that can be expressed as a product of nullary and unary functions, binary equality functions and binary disequality functions; these functions are said to be of *product type*. Denote by $\mathcal{A}$ the set of functions whose support is an affine relation, and whose range is a subset of $\{0, b\}$ for some $b \in \mathbb{R}_{\geq 0}$; these functions are said to be *pure affine*. In other words, to get a pure affine function we interpret an affine relation as a 0,1-function, then multiply that function by a positive constant. Dyer, Goldberg and Jerrum [21] show that #CSP($\mathcal{F}$) is in FP if $\mathcal{F} \subset \mathcal{P}$ or $\mathcal{F} \subset \mathcal{A}$. In all other cases, #CSP($\mathcal{F}$) is #P-hard.

Notice that the addition of non-negative real weights did not significantly change the statement of the dichotomy, only its proof. The first indication that something new and interesting happens when we extend the domain to negative real numbers comes with the following example. Denote by $H_2 : \{0,1\}^2 \to \mathbb{R}$ the function defined by $H_2(x,y) = -1$ if $x = y = 1$ and $H_2(x,y) = +1$ otherwise. We can interpret #CSP($\{H_2\}$) as the problem of counting induced subgraphs of a graph G which have an even (or odd) number of edges. To see this, view an instance (X, C) of #CSP($\{H_2\}$) as an undirected graph G, with vertex set X and with edges determined by the scopes of the constraints in C. An assignment to variables in X can be interpreted as the indicator function of a vertex subset $U \subseteq V(G)$ of G. If the subgraph $G[U]$ induced by U has as even (respectively, odd) number of edges then it contributes an additive $+1$ (respectively, -1) to the solution of the instance (X, C). Given that the total number of induced subgraphs is $2^{|V(G)|}$, the solution to the counting CSP easily yields the number of induced subgraphs with an even (or odd) number of edges.

It transpires that #CSP($\{H_2\}$) $\in$ FP. As we saw, the required output for an instance (X, C) is a sum of $2^{|X|}$ terms, each of them ± 1. Letting $X = \{x_1, \ldots, x_n\}$, consider the quadratic form over $\mathbb{F}_2$ defined by $Q(X) = \sum_{\{i,j\} \in S} x_i x_j$, where S is the set of all scopes of constraints in C. Note that $Q(X) = 0$ if the assignment to variables corresponds to a $+1$ term and $Q(X) = 1$ otherwise. So the number of positive terms in the sum is equal to the number of solutions to $Q(X) = 0$, and once we know the number of positive terms, we know the sum itself. Now any quadratic form over $\mathbb{F}_2$ is equivalent under linear substitutions of variables to a quadratic form over a possibly smaller number of variables, in canonical form [48, Thm 6.30]. This canonical form allows easy calculation of the number of solutions. This tractable example was first noted in the context of counting CSPs by Goldberg, Grohe, Jerrum and Thurley [35], and it, and others like it, substantially complicate the classification programme when negative or complex weights are involved.

Nevertheless, the challenges were incrementally overcome, and the Boolean #CSP dichotomy was extended to arbitrary real weights by Bulatov, Dyer, Goldberg, Jalsenius and Richerby [3], and then extended further to arbitrary complex weights by Cai, Lu and Xia [13]. Let $\mathcal{A}$ denote the set of complex functions $f(x_1, \ldots, x_k)$ that are the product of a pure affine function (defined as above, but with $b \in \mathbb{C}$ now a complex number) and a certain rotation

$\omega(x_1, \ldots, x_k)$, which takes values in the set of fourth roots of unity. Specifically, identify the Boolean domain with the field $\mathbb{F}_2$, and denote by $\mathbf{x}'$ the vector $(x_1, \ldots, x_k, 1) \in \mathbb{F}_2^{k+1}$ of arguments to f, extended to the right by the constant 1. Then there are vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{F}_2^{k+1}$, such that the value of the rotation $\omega(x_1, \ldots, x_k)$ is given by $i^{L_1(\mathbf{x}')+L_2(\mathbf{x}')+\cdots+L_n(\mathbf{x}')}$, where $i = \sqrt{-1}$, and each $L_j(\mathbf{x}')$ is the indicator function for $\mathbf{a}_j \cdot \mathbf{x}' = 1$. Note that the dot product is computed over $\mathbb{F}_2$, whereas the sum $L_1(\mathbf{x}') + L_2(\mathbf{x}') + \cdots + L_n(\mathbf{x}')$ is computed over $\mathbb{Z}$ or, equivalently, over $\mathbb{Z}_4$. This completes the definition of $\mathcal{A}$. As before let $\mathcal{P}$ be the set of all functions which can be expressed as a product of nullary and unary functions, binary equality functions and binary disequality functions (now extended to complex functions). Cai, Lu and Xia [13] proved the following dichotomy.

▶ **Theorem 2.** *Suppose $\mathcal{F}$ is a finite set of functions mapping Boolean inputs to complex numbers. If $\mathcal{F} \subset \mathcal{A}$ or $\mathcal{F} \subset \mathcal{P}$, then $\#\mathrm{CSP}(\mathcal{F})$ is in* FP*. Otherwise, $\#\mathrm{CSP}(\mathcal{F})$ is* #P*-hard.*

This wraps up the Boolean case as far as the complexity of exact computation is concerned. We not only have a dichotomy, but one that takes an explicit form which is quite easy to comprehend, even in its most general statement (Theorem 2). Unfortunately, this happy state of affairs will not continue as we move to domains of cardinality greater than two.

2.2 Graph Homomorphisms and Partition Functions

Another natural subclass of counting CSPs, and one that has important connections to other fields, is obtained by restricting the constraint language Γ to a single binary symmetric relation. It is natural to view this relation as an undirected graph H, and the instance also as an undirected graph G whose vertices correspond to variables, and whose edges correspond to the scopes of the constraints. A (graph) homomorphism from G to H is a function $\phi : V(G) \to V(H)$ such that $\{\phi(u), \phi(v)\} \in E(H)$ whenever $\{u, v\} \in E(H)$. Thus, the problem $\#\mathrm{CSP}(\Gamma)$ is equivalent to counting homomorphisms from G to H. This correspondence gives rise to the alternative names *H-homomorphisms* or *H-colourings* for this restriction of counting CSPs. In the latter case, we are thinking of the vertices of H as "colours", and viewing a homomorphism from G to H as a colouring of the vertices of G in which the colours of adjacent vertices of G are constrained by the adjacency relation of H. We see that CSPs with one symmetric relation generalise usual graph colouring, in the same way that Boolean CSPs generalise the usual CNF satisfiability problem.

It is customary to allow the fixed graph H to have loops but not parallel edges; in other words, we do not assume that the single relation in Γ is irreflexive. To focus on the irreflexive situation would exclude some interesting and natural examples. For a graph H, possibly with loops, but without parallel edges, denote by $\#H$-Col is the following problem: given a graph G, return the number of graph homomorphisms from G to H. By way of example, if K_2' is the connected graph on two vertices with a loop on one of them, and K_3 is the complete graph on three vertices with no loops, then $\#K_2'$-Col asks for the number of independent sets, in G, and $\#K_3$-Col the number of (proper, vertex) 3-colourings. The first step in the study of the complexity of $\#H$-Col was made by Dyer and Greenhill [23] who proved the following dichotomy. Say that a graph is *reflexive* if all its vertices have loops and *irreflexive* if it is loop-free.

▶ **Theorem 3.** *If every connected component of H is a reflexive complete graph or an irreflexive complete bipartite graph, then $\#H$-Col is in* FP*. Otherwise, $\#H$-Col is* #P*-complete.*

The next step is to add weights. A weighted graph H on q vertices, can be thought of as a weighted adjacency matrix, which is a symmetric $q \times q$ matrix $A = (a_{ij} : 0 \leq i, j < q)$ with non-negative real entries. In statistical physics terminology, the matrix A defines a *spin model*. We'll refer to the matrix A as the interaction matrix of the model. For an instance G, which is an undirected graph, the *partition function* Z_A of this model is defined as follows:

$$Z_A(G) = \sum_{\sigma : V(G) \to [q]} \prod_{\{u,v\} \in E(G)} a_{\sigma(u),\sigma(v)}, \tag{1}$$

where $[q] = \{0, \ldots, q-1\}$. By way of example, consider the following matrices

$$A^{\lambda}_{\text{Ising}} = \begin{pmatrix} \lambda & 1 \\ 1 & \lambda \end{pmatrix} \quad \text{and} \quad A_{\text{BIS}} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \tag{2}$$

(The second matrix may look a little mysterious, but it provides an illuminating test case, and will play an important role in the second part of this survey.) These interaction matrices yield, respectively, the partition functions of the classical Ising model (ferromagnetic in the case $\lambda > 1$ and antiferromagnetic when $\lambda < 1$) and the independent set (or hard-core) model on a bipartite graph. This correspondence between, on the one hand, partition functions of spin models in statistical physics and, on the other, weighted counting CSPs with a single symmetric binary function has led to this special case of counting CSPs being extensively studied.

As before, the generalisation to non-negative weights goes smoothly. Bulatov and Grohe [4] showed that an analogue of Theorem 3 holds for the computation of Z_A, with the role of the reflexive complete graph being taken by a rank-1 interaction matrix A, and that of the irreflexive bipartite graph being taken by an adjacency matrix in block form $\left(\begin{smallmatrix} 0 & B \\ B^{\top} & 0 \end{smallmatrix}\right)$, where B is of rank 1. Technically, we must also replace the conclusion of "#P-completeness" by "#P-hardness", as the required output is no longer an integer. Note that, unfortunately, all non-trivial spin systems, including the ones specified by A_{BIS} and $A^{\lambda}_{\text{Ising}}$ (with $\lambda \neq 1$) have hard-to-compute partition functions.

Generalising to arbitrary real weights, as we saw earlier, significantly increases the complexity. The 2×2 matrix H_2 we encountered in the previous section is one of an infinite sequence of matrices leading to tractable partition functions. Other Hadamard matrices yield tractable functions on larger domains. However, this is far from the end of the story: not every Hadamard matrix yields a tractable counting CSP, and not every tractable Boolean counting CSP arises directly from a Hadamard matrix. Nevertheless, Goldberg, Grohe, Jerrum and Thurley [35] showed that there is a dichotomy into FP and #P-hard. The characterisation is too complicated to describe here, though it is decidable. Cai, Chen and Lu [11] made the final step, generalising to complex weights. Again there is a dichotomy which is decidable (in fact in polynomial time) but too complex to describe here. It seems at this point that we have come a long way, but we'll see in Section 2.4 that we can go a quite a bit further yet.

2.3 Send in the Clones

Before studying general counting CSPs it is worth taking time to digress into functional clones, the counting analogue of relational clones, or co-clones.

Let D be a finite domain, $\mathcal{U}_k$ be the set of functions $D^k \to \mathbb{R}_{\geq 0}$, and $\mathcal{U} = \cup_{k=0}^{\infty} \mathcal{U}_k$. Denote by EQ the equality function defined by $\text{EQ}(x, y) = 1$ if $x = y$ and $\text{EQ}(x, y) = 0$ otherwise.

A set of functions $\mathcal{F} \in \mathcal{U}$ is a *functional clone* of it contains equality, and is closed under variable introduction, variable renaming, product, and summation over a variable. If $\mathcal{F} \subseteq \mathcal{U}$ is some (usually finite) set of functions, then $\langle \mathcal{F} \rangle_{\#}$ denotes the *functional clone generated by* $\mathcal{F}$, that is to say, the minimal functional clone containing $\mathcal{F}$. We say that a function $f \in \mathcal{U}$ is *pps-definable over* $\mathcal{F}$ if $f \in \langle \mathcal{F} \rangle_{\#}$. Relative to classical pp-definability, we have replaced conjunction by product and existential quantification by summation over a variable. However, the idea is the same: a function f is pps-definable over $\mathcal{F}$ if it can be "implemented" in terms of functions in $\mathcal{F}$, in just the same way as a relation is pp-definable over Γ if it can be "implemented" in terms of relations in Γ. Also, just as in the relational case, if $\langle \mathcal{F} \rangle_{\#} = \langle \mathcal{F}' \rangle_{\#}$ then #CSP($\mathcal{F}$) and #CSP($\mathcal{F}'$) are of equivalent computational complexity. For details, refer to [8], but note that there the subscript # is dropped from the notation for functional clones, since functional clones are the main object of study in that article. Note also that a more inclusive notion of functional clone is defined there which requires closure under taking limits of sequences of functions of the same arity.

Observe that even if we start in the unweighted or relational situation with a finite set of functions with range $\{0, 1\}$, then pps-definablity will soon generate more general functions and take us into the weighted situation. Thus, with $D = \{0, 1\}$,

$$g(x, y) = \sum_{z \in \{0,1\}} \mathrm{IMP}(x, z)\, \mathrm{IMP}(z, y),$$

defines the function g taking the values $g(0, 0) = g(1, 1) = 1$, $g(1, 0) = 0$ and $g(0, 1) = 2$. As an aside, the rationale for introducing limits is the following. It is fairly easy to check that the functional clone $\langle \mathcal{U}_0 \cup \{\mathrm{IMP}\} \rangle_{\#}$ contains, for all $n \in \mathbb{N}$, the unary function $f(x)$ defined by $f(0) = 2^{-n}$ and $f(1) = 1$, but not the function defined by $f(0) = 0$ and $f(1) = 1$. This is a trivial example, but it suggests that in some situations it may be reasonable to include limits.

Both the utility of functional clones and their limitations can be appreciated by reproving Theorem 1 using this technology. First a little notation. For a function $f : D^k \to \mathbb{R}_{\geq 0}$, define $\operatorname{supp} f \subseteq \{0, 1\}^k$ to be the relation $\operatorname{supp} f = \{x \in D^k : f(x) > 0\}$. Extend this notation to sets of functions $\mathcal{F} \subseteq \mathcal{U}$ via $\operatorname{supp} \mathcal{F} = \{R : R = \operatorname{supp} f, \text{ for some } f \in \mathcal{F}\}$. If $\mathcal{F}$ is a functional clone then $\operatorname{supp} \mathcal{F}$ is a relational clone, and, moreover, for any set of functions $\mathcal{F} \subseteq \mathcal{U}$ we have $\operatorname{supp} \langle \mathcal{F} \rangle_{\#} = \langle \operatorname{supp} \mathcal{F} \rangle$, where $\langle \Gamma \rangle$ denotes the relational clone generated by a set of relations Γ. To appreciate this fact, consider the homomorphism $\varphi : (\mathbb{R}_{\geq 0}, +, \times) \to (\{0, 1\}, \vee, \wedge\}$ defined by $\varphi(x) = 0$ if $x = 0$ and $\varphi(x) = 1$ if $x > 0$, and its action on the closure operations of pps-definability; alternatively, refer to [8].

So to the proof of Theorem 1. We specialise the above definitions to the Boolean domain $\{0, 1\}$. In making use of Post's lattice, we refer to Böhler, Creignou, Reith and Vollmer [1] and Creignou, Kolaitis and Zanuttini [16], and employ their terminology. So suppose Γ is a finite subset of affine relations, that is to say $\Gamma \subset \mathsf{IL}_2$. As we noted earlier, the set of satisfying assignments to an instance of #CSP(Γ) can be expressed as the solution set to a system of linear equations over $\mathbb{F}_2$, and so #CSP(Γ) is in FP.

Now suppose $\Gamma \not\subseteq \mathsf{IL}_2$. For a relation $R \subseteq \{0, 1\}^k$, denote by $\operatorname{fun} R$ the derived function $f : \{0, 1\}^k \to \mathbb{R}_{\geq 0}$ defined by $f(x) = 1$ if $x \in R$ and $f(x) = 0$ otherwise. Extend fun to sets of relations via $\operatorname{fun} \Gamma = \{\operatorname{fun} R : R \in \Gamma\}$. Then the relational clone C generated by Γ is

$$C = \langle \Gamma \rangle = \langle \operatorname{supp} \operatorname{fun} \Gamma \rangle = \operatorname{supp} \langle \operatorname{fun} \Gamma \rangle_{\#}. \tag{3}$$

We know that C is not IL_2, nor any relational clone that lies below IL_2 in Post's lattice of relational clones. Inspecting Figure 2 of [1], we see that this implies that C contains one

of relational clones IS_1^2, IS_0^2, IM or IN. In the first, second and third cases cases (see [16, Table 2]), C contains NAND, OR and IMP, respectively. By (3), this is turn implies that $\langle \operatorname{fun} \Gamma \rangle_{\#}$ contains a function f such that $\operatorname{supp} f$ is one of those three relations. But we know, from routine direct arguments, or from [4], that $\#\mathrm{CSP}(\{f\})$ is #P-hard in any of those three cases. In this context, note that the interaction matrix

$$A = \begin{pmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{pmatrix}$$

associated with f is necessarily of rank 2. The final case, $C \supseteq \mathsf{IN}$ presents a slight fly in the ointment. The relational clone IN contains precisely the relations that are 0-valid, 1-valid and "complementive", i.e., invariant under interchange of 0 and 1. In this case, $\langle \operatorname{fun} \Gamma \rangle_{\#}$ contains a function f such that $\operatorname{supp} f$ is the relation $\{0,1\}^3 \setminus \{(1,1,0),(0,0,1)\}$. Although one or other of the binary functions $f(x,y,z)\,\mathrm{EQ}(y,z)$ or $\sum_{z\in\{0,1\}} f(x,y,z)$ may fail the rank-2 test, it may be verified that at least one will pass. This demonstrates that $\Gamma \not\subseteq \mathsf{IL}_2$ implies $\#\mathrm{CSP}(\Gamma)$ is #P-hard, and completes the proof.

This argument could be carried through without using functional clones, using a theorem of Bulatov and Dalmau's [7, Thm 2], but part of the motivation for writing down the proof in the language of functional clones was to introduce a concept that will be important later. Even at this stage it is possible to appreciate the limitations of the utility of functional clones. We may very well be interested in sets $\mathcal{F}$ of functions, all of which have the complete relation as their supports. In that case, the above line of argument yields nothing. Bulatov's "max-co-clones" [6] may be viewed partly as a response to this failing.

2.4 #CSPs in General

The Feder-Vardi conjecture for decision CSPs [27] is famously open in its original form but, remarkably, has been resolved positively for counting CSPs by Bulatov [5]. The original proof was streamlined by Dyer and Richerby [24], who reduced the dependence on universal algebra and showed that the dichotomy is decidable. In order to state the dichotomy in the form given by Dyer and Richerby, we require some definitions. A matrix is said to be a *rank-one block matrix* if it can be transformed (by row and column permutations) into block diagonal form, such that every block has rank one. A ternary relation $R \subseteq A_1 \times A_2 \times A_3$ is *balanced* if the *balance matrix*

$$M(x,y) = |\{z \in A_3 : (x,y,z) \in R\}|, \quad \text{for all } x \in A_1 \text{ and } y \in A_2$$

is a rank-one block matrix. A set of relations Γ over domain D is *strongly balanced* if every ternary relation that is pp-definable over Γ is balanced.

▶ **Theorem 4.** *Suppose Γ is a finite set of relations over a finite domain D. If Γ is strongly balanced then $\#\mathrm{CSP}(\Gamma)$ is in FP. Otherwise, $\#\mathrm{CSP}(\Gamma)$ is #P-complete. Moreover, the dichotomy is decidable.*

It is possible to offer some hints towards the proof. First, some definitions. A binary relation $B \subseteq A_1 \times A_2$ is *rectangular* if $(a,c),(a,d),(b,c) \in B$ implies $(b,d) \in B$ for all $a,b \in A_1$ and $c,d \in A_2$. Suppose $R \subseteq D^n$ is a relation of arity $n \geq 2$. For each non-trivial partition of $[n]$ into blocks of size k and $n-k$ there is a natural isomorphism between D^n and $D^k \times D^{n-k}$ under which R can be viewed as a binary relation on $D^k \times D^{n-k}$. We say that R is rectangular if every expression of R as a binary relation on $D^k \times D^{n-k}$, for $1 \leq k < n$, is rectangular. A constraint language Γ is *strongly rectangular* if every relation in $\langle \Gamma \rangle$ of

arity at least 2 is rectangular. Finally a relation $R \subseteq D^n$ is strongly rectangular if $\langle\{R\}\rangle$ is strongly rectangular. Dyer and Richerby [24] show that if Γ is strongly balanced then Γ is strongly rectangular, but that the converse does not hold. They also show that if Γ is not strongly balanced then #CSP(Γ) is #P-complete; this strengthens a result of Bulatov and Dalmau [7] that if Γ is not strongly rectangular then #CSP(Γ) is #P-complete.

The really difficult part of the proof is demonstrating tractability in the case that Γ is strongly balanced. For this we need the concept of a frame, which provides is a compact representation for strongly rectangular relations. Say that a set $D' \subseteq D$ is *i-equivalent in a relation R* if R contains tuples which agree on their first $i-1$ elements and whose ith elements are exactly the members of D'. A *frame* for a relation $R \subseteq D^n$ is a relation $F \subseteq R$ satisfying two properties: (i) whenever R contains a tuple whose ith component is a then F also contains such a tuple, and (ii) for $1 < i \leq n$, any set that is i-equivalent in R must also be i-equivalent in F. It can be shown that every strongly rectangular relation $R \subseteq D^n$ has a small frame, specifically, one of cardinality $n\,|D|$.

In the decision world, Bulatov and Dalmau [2] showed (though expressed in different terminology) that CSP(Γ) $\in$ FP for every strongly rectangular constraint language Γ. This result cannot translate to counting CSPs unless #P $\subseteq$ FP. However, it can be reproved with the technology of frames, giving a pointer as to how to proceed. Suppose (X, C) is an instance of CSP(Γ) with $|X| = n$. Note that the n-ary relation R defined by (X, C) is strongly rectangular, since $\langle\{R\}\rangle \subseteq \langle\Gamma\rangle$ and Γ is strongly rectangular. We can construct a small frame for R iteratively, starting with a frame for the complete relation $R_0 = D^n$ Let $R_0 \supseteq R_1 \supseteq \cdots \supseteq R_{|C|} = R$ be a sequence of relations in which R_i is obtained from R_{i-1} by removing the tuples that violate the ith constraint. At each step the frame is updated so that it represents the current relation. The process ends with a frame for R. It can be shown that a frame is empty iff the relation it represents is empty, so this process yields a polynomial-time algorithm for the decision problem CSP(Γ) in the case that Γ is strongly rectangular.

Dyer and Richerby demonstrate that frames can be used to count solutions, under the stronger assumption that Γ is strongly balanced. As before, they construct a frame for the relation R. Their approach is then one of dynamic programming based on a carefully selected set of subproblems. For $1 \leq i < j \leq n$, let $N_{i,j}(a)$ be the number of prefixes $(u_1, \ldots, u_i) \in D^i$ such that there is a tuple $(u_1, \ldots, u_n) \in R$ with $u_j = a$. The key step of the iteration is computing $N_{i,j}(\cdot)$ for each $j > i$, given $N_{i-1,j}(\cdot)$ for each $j \geq i$, and it turns out that this can be achieved using the property of strong balance. At the end of the process, summing $N_{n-1,n}(a)$ over $a \in D$ gives $|R|$.

The universal algebra doesn't go away, but is reduced to an easy to digest and intuitively appealing fragment. A Mal'tsev operation is a ternary operation $\varphi : D^3 \to D$ satisfying $\varphi(a, a, b) = \varphi(b, a, a) = b$ for all $a, b \in D$. An important fact proved in [24] is that a constraint language is strongly rectangular if and only if it has a Mal'tsev polymorphism. This fact has two important consequences. First, it allows an efficient implementation of membership testing in a strongly rectangular relation R given only a frame for R. Second, it allows an efficient (in NP) test for strong rectangularity. (Note that the definition of strong rectangularity in itself does not even imply decidability.) Testing strong rectangularity is the first step in testing strong balance. It transpires that deciding strong balance (and hence the dichotomy itself) is in NP.

The resolution of the counting version of the Feder-Vardi conjecture is a major achievement. One might ask how it is that the counting version has been resolved while the original decision version has not. Of course, this is a vague, possibly nonsensical question. However, it is

difficult to avoid the thought that tractability results are generally harder to prove than intractability results, and #CSP(Γ) simply has fewer tractable cases than CSP(Γ).

Perhaps as remarkable as the dichotomy for relational constraint languages itself is the fact that it has been extended to the weighted case. The first step, to non-negative real weights, was taken by Cai, Chen and Lu [10]. As we have seen already, the extension of dichotomies in counting CSPs to arbitrary real weights adds new possibilities for tractable cases that must be taken into account, and the further extension to complex weights provides further complications. This line of work culminated with Cai and Chen [9], who proved the existence of a dichotomy in the complex weighted case. They provide three rather clean conditions on a set of complex functions $\mathcal{F}$ – block orthogonality, Mal'tsev and type partition – that taken together imply $\#\mathrm{CSP}(\mathcal{F}) \in \mathsf{FP}$. If any of the conditions fail, then $\#\mathrm{CSP}(\mathcal{F})$ is #P-hard. Unfortunately, the conditions are not currently known to be decidable.

Although the conditions of block orthogonality, Mal'tsev and type partition, are really quite clean, it would take too much space to define them here. Nor is it feasible to give a sketch of the proof techniques. For those things the reader should consult the really lucid exposition of the definitions and proof sketch to be found in the conference version of Cai and Chen's paper [9]. Suffice it to say that the main ingredients of Dyer and Richerby's work survive, namely the compact representation of relations and the Mal'tsev polymorphism that allows information to be extracted from it, but completely new ideas need to be added, particularly in the definition and application of the type partition condition.

In summary, there has been massive progress in our understanding of the computational complexity of counting CSPs. In fact, the main questions in the basic setting have been all but answered. That is not to say that there is not much work to do: for example, with the notable exception of the extensive literature on read-twice #CSPs or holants, there is not a great deal of work on restricted instances, e.g., planar [44], and perhaps none at all on infinite domains.

3 Approximate Computation

We saw in Section 2 that certain non-trivial counting CSPs are exactly solvable using interesting algorithmic techniques, such as reduction to a system of linear equations over a finite field or to a quadratic form over $\mathbb{F}_2$. However, the general picture is gloomy, with intractability results dominating. This observation has prompted the search for approximation algorithms. An encouraging sign is that the partition function of a ferromagnetic Ising system (i.e., an instance of the two-spin model specified by the interaction matrix $A^{\lambda}_{\mathrm{Ising}}$, with $\lambda > 1$) can be computed with small relative error in polynomial time [45]. Note that the interaction matrix $A^{\lambda}_{\mathrm{Ising}}$ has rank two, so the partition function is #P-hard to compute exactly.

Before embarking on the study of specific counting CSPs, we need to say a something about the computational complexity of approximate counting problems in general. There is a well-established framework for this. We provide only an informal description here and direct the reader to Dyer, Goldberg, Greenhill and Jerrum [19] for precise definitions.

The standard notion of efficient approximation algorithm is *Fully Polynomial Randomised Approximation Scheme*, or FPRAS. This is a randomised algorithm that is required to produce a solution within relative error specified by a tolerance $\varepsilon > 0$, in time polynomial in the instance size and ε. Under some mild condition, an efficient algorithm that provides only very weak approximations can be boosted to the achieve the quality of approximation demanded by an FPRAS. As a consequence, in the context of counting problems, there is just

one notion of approximation algorithm, namely FPRAS.[1] In this aspect, counting problems provide a contrast with optimisation problems, which exhibit a hierarchy of possible degrees of approximability.

Evidence for the non-existence of an FPRAS for a problem Π can be obtained through *Approximation-Preserving* (or AP-) *reductions*. These are polynomial-time Turing reductions that preserve (closely enough) the error tolerance. The key feature of the definition is that the class of problems admitting an FPRAS is closed under AP-reducibility. Every problem in #P is AP-reducible to #SAT, so #SAT is complete for #P with respect to AP-reductions. In the other direction, we know, using the bisection technique of Valiant and Vazirani [55, Corollary 3.6], that #SAT can be approximated (in the FPRAS sense) by a polynomial-time probabilistic Turing machine equipped with an oracle for the decision problem SAT. Thus, the counting version of any NP-complete decision problem is complete for #P with respect to AP-reductions. Note the contrast with exact computation, where there may exist NP-complete decision problems whose counting analogue is not #P-hard under classical Turing reducibility.

We can summarise the situation as follows. Assuming we restrict attention to counting problems in #P (and this includes all problems of the form #CSP(Γ)), the hardest problems Π are those that are complete for #P with respect to AP-reducibility. Since such problems are AP-interreducible with #SAT, we will use the shorthand "Π is #SAT-equivalent", omitting the qualification "with respect to AP-reducibility" for brevity. We know that these problems do not have an FPRAS unless RP = NP. At the risk of overemphasising the point, in the context of approximate computation, the complexity of a problem that is AP-interreducible with #SAT lies only a little above NP (formally in the class $\mathsf{FP}^{\mathsf{NP}}$) and presumably far below #P.

Sometimes we have to settle for weaker evidence of computational intractability. The problem of counting independent sets in a bipartite graph is denoted by #BIS. The problem #BIS appears to be of intermediate complexity: on the one hand, there is no known FPRAS for #BIS (and it is generally believed that none exists) but, on the other hand, there is no known AP-reduction from #SAT to #BIS. The fact that #BIS is complete for a certain complexity class $\#\mathsf{RH}\Pi_1$ with respect to AP-reducibility [19], can be interpreted as evidence for the special status of #BIS and the problems AP-interreducible with it.

If there is an AP-reduction from #BIS to Π, we say that Π is #BIS-*hard*. We conjecture that no FPRAS for #BIS exists, in which case the same is true for all #BIS-hard problems. If there exists an AP-reduction from Π to #BIS, we say that Π is #BIS-*easy*; if Π is #BIS-hard and #BIS-easy then we say that Π is #BIS-*equivalent*. Many problems are in this last class, including counting downsets in a partial order [19], estimating the partition function of the Widom-Rowlinson model [19] or of the ferromagnetic Ising model with an external field [36]. In the absence of NP-hardness, the claim of #BIS-equivalence is currently almost the strongest one can make for an approximate counting problem Π, in that it locates the complexity of Π quite precisely.

3.1 Boolean #CSPs

As usual, restricting attention to Boolean #CSPs, i.e., those with domain-size two, allows us to make a brisk start. Let us further simplify matters by considering the unweighted

[1] This is not quite accurate. Another sweet spot is occupied by algorithms that provide an additive approximation to the *logarithm* of the solution, within $\pm\varepsilon n$, where n is the instance size.

case. Recall that IL_2 is the clone of affine relations, i.e., relations that can be expressed as the solution set of a system of linear equations over $\mathbb{F}_2$. Define the relational clone IM_2 by $\mathsf{IM}_2 = \langle \mathrm{IMP}, \delta_0, \delta_1 \rangle$, where δ_0 and δ_1 are the unary relations $\delta_0 = \{(0)\}$ and $\delta_1 = \{(1)\}$. Dyer, Goldberg and Jerrum [22] prove the following.

▶ **Theorem 5.** *Let Γ be a Boolean constraint language. If every relation in Γ is in IL_2, then $\#\mathrm{CSP}(\Gamma)$ is in FP. Otherwise, if every relation in Γ is in IM_2 then $\#\mathrm{CSP}(\Gamma)$ is $\#\mathrm{BIS}$-equivalent. Otherwise, $\#\mathrm{CSP}(\Gamma)$ is $\#\mathrm{SAT}$-equivalent.*

Using the language of functional clones and Post's lattice, it is possible to hint at a proof. For example, letting $C = \langle \Gamma \rangle$, any constraint language covered by the final part of the theorem will satisfy $C \not\subseteq \mathsf{IL}_2$ and $C \not\subseteq \mathsf{IM}_2$. Consulting Post's lattice of relational clones [1, Fig. 2], we find that C contains one of IS_1^2, IS_0^2 or IN. In the first two cases we can find an AP-reduction from the problem of counting independent sets in a general graph to $\#\mathrm{CSP}(\Gamma)$, and in the third case an AP-reduction from the problem of evaluating the partition function of the antiferromagnetic Ising model. Both the independent set and antiferromagnetic Ising problems are #SAT-equivalent, showing that $\#\mathrm{CSP}(\Gamma)$ is also. This sketch can be completed to a proof of the final part of the theorem.

Assuming that there is no FPRAS for #BIS, Theorem 5 is a discouraging result, as it says that the only Boolean counting CSPs that are efficiently approximable are the affine ones, which we already know to be exactly solvable. So relaxing the problem specification appears to have gained us nothing. Although we shall not be discussing restricted problem instances extensively in this survey, it is worth pointing out that Dyer, Goldberg, Jalsenius and Richerby [20] have shown that the hardness results in Theorem 5 continue to hold for instances of degree at most six, where the *degree* of a CSP instance is the maximum number of occurrences of any variable in the instance.

The next step is to introduce weights. We restrict attention to non-negative real weights, as this situation seems to give the greatest scope for positive results. (If negative weights are allowed, it is likely that we will be required to compute a small quantity that is the difference of two much larger quantities, and it will be hard to achieve small relative error.) Recall the material on functional clones from Section 2.3. No generally applicable theory of polymorphisms for functional clones exists. However, some interesting functional clones can be defined by operations reminiscent of multimorphisms or fractional polymorphisms in the study of valued CSPs (VCSPs).

Denote by $\mathcal{B}_k$ the set of functions $\{0,1\}^k \to \mathbb{R}_{\geq 0}$, and write $\mathcal{B} = \cup_{k=0}^{\infty} \mathcal{B}_k$. A function $f \in \mathcal{B}_n$ is *log-supermodular* (lsm) if

$$f(\mathbf{x} \vee \mathbf{y}) f(\mathbf{x} \wedge \mathbf{y}) \geq f(\mathbf{x}) f(\mathbf{y}), \quad \text{for all } \mathbf{x}, \mathbf{y} \in \mathcal{B}_n, \tag{4}$$

where $\wedge$ and $\vee$ denote meet and join in the Boolean lattice, which are equivalent to pointwise min and max. The terminology is justified by the observation that $f \in \mathcal{B}_n$ is lsm if and only if $\log f$ is supermodular. Note the similarity to multimorphisms in the study of VCSPs, but with multiplication replacing addition. We denote by $\mathsf{LSM} \subset \mathcal{B}$ the class of all lsm functions.

In the weighted situation we need to work with functional clones that are closed under taking limits of sequences of functions; as our functions are defined on a finite domain we don't need to be specific about the notion of convergence. The clone generated by a set of functions $\mathcal{F}$ with the additional limiting operation is denoted $\langle \mathcal{F} \rangle_{\#,\omega}$ There is no general result to the effect that sets of functions defined by conditions such as (4) are clones; each case must be handled individually. In this instance we do have a clone [8, Lemma 4.2].

▶ **Lemma 6.** $\langle \mathsf{LSM} \rangle_{\#,\omega} = \mathsf{LSM}$.

The non-trivial part of the proof lies in showing that LSM is closed under the operation of summing over a variable. It turns out that this requirement can be viewed as a special case of the Ahlswede-Daykin four-functions theorem.

As usual, we can say more about the conservative case, where all unary functions $\mathcal{B}_1$ are given free. Bulatov, Dyer, Goldberg, Jerrrum and McQuillan show [8].

▶ **Theorem 7.** *Suppose $\mathcal{F} \subseteq \mathcal{B}$.*
- *If $\mathcal{F} \not\subseteq \langle \mathrm{NEQ}, \mathcal{B}_1 \rangle_{\#}$ then $\langle \mathrm{IMP}, \mathcal{B}_1 \rangle_{\#,\omega} \subseteq \langle \mathcal{F}, \mathcal{B}_1 \rangle_{\#,\omega}$.*
- *If, in addition, $\mathcal{F} \not\subseteq$ LSM then $\langle \mathcal{F}, \mathcal{B}_1 \rangle_{\#,\omega} = \mathcal{B}$.*

Informally, every non-trivial functional clone contains $\langle \mathrm{IMP}, \mathcal{B}_1 \rangle_{\#,\omega}$ and any non-trivial clone containing a non-lsm function is in fact $\mathcal{B}$. In other words, all the interesting action takes place between $\langle \mathrm{IMP}, \mathcal{B}_1 \rangle_{\#,\omega}$ and LSM.

Care is needed to obtain computational consequences from Theorem 7. In particular, it is necessary to introduce computationally efficient versions of $\mathcal{B}$ and of the closure operation $\langle \cdot \rangle_{\#,\omega}$. These are needed so that we can compute efficiently with functions in $\mathcal{B}$, and so that we can utilise the limiting operation in the proofs. This programme can in fact be carried out (see [8] for details), resulting in the following classification theorem, in which we assume that the necessary efficient versions of concepts are used.

▶ **Theorem 8.** *Suppose $\mathcal{F}$ is a finite subset of $\mathcal{B}$.*
1. *If $\mathcal{F} \subseteq \langle \mathrm{NEQ}, \mathcal{B}_1 \rangle_{\#}$ then there is an FPRAS for $\#\mathrm{CSP}(\mathcal{F})$.*
2. *Otherwise,*
   - **(a)** *there is a finite subset S of $\mathcal{B}_1$ such that $\#\mathrm{CSP}(\mathcal{F}, S)$ is #BIS-hard, and*
   - **(b)** *if $\mathcal{F} \not\subseteq$ LSM then there is a finite subset S of $\mathcal{B}_1$ such that $\#\mathrm{CSP}(\mathcal{F}, S)$ is #SAT-equivalent.*

The polynomial-time algorithm guaranteed in the first part of the theorem needs to compute sufficiently accurate approximations to functions in $\mathcal{F} \cap \mathcal{B}_1$; it is for this reason only that we specify an FPRAS and not an exact algorithm. The second part of the theorem may be conveniently illustrated with reference to the Ising model. The ferromagnetic Ising model with a field is covered by part (2a) of the theorem, and hence its partition function is #BIS-hard. (In fact the partition function is #BIS-equivalent, as can be seen from [8, Lemma 7.1] or Theorem 11(1b).) The antiferromagnetic Ising model with a field is covered by part (2b), and hence its partition function is #SAT-equivalent. These special cases were known earlier ([36] and [45]), but Theorem 8 places these isolated intractability results in a general setting. The Ising model will be discussed at greater length in §3.3.2.

It is natural to ask if Theorem 7 can be strengthed to a strict trichotomy. Unfortunately the answer is no. Consider the function $g : \{0,1\}^4 \to \mathbb{R}_{\geq 0}$ defined by

$$g(x_1, x_2, x_3, x_4) = \begin{cases} 4, & \text{if } x_1 + x_2 + x_3 + x_4 = 4; \\ 2, & \text{if } x_1 + x_2 + x_3 + x_4 = 3; \\ 1, & \text{otherwise.} \end{cases} \quad \text{and}$$

The function g is in LSM but not in $\langle \mathrm{IMP}, \mathcal{B}_1 \rangle_{\#,\omega}$ [8, Lemma 11.9]. Nevertheless, it is entirely possible that $\#\mathrm{CSP}(\{g\} \cup \mathcal{B}_1)$ is #BIS-easy, since AP-reduction is a more liberal notion than pps-definabilty.

Theorem 8 encapsulates most of what is known about the computational complexity of general conservative Boolean counting CSPs. When we go beyond conservative, we know rather little. We do not even have a complete understanding of $\#\mathrm{CSP}(\mathcal{F})$ when $\mathcal{F} \subset \mathcal{B}_2$.

The problem is that the boundary between tractable and intractable becomes intimately tied up with the existence of phase transition in spin systems. However, much attention has been directed to this issue, and the restriction to the case where $\mathcal{F}$ contains a single *symmetric* binary functions is now well understood, as we shall see in the next section.

3.2 Graph Homomorphisms

We turn to the case of a single binary relation, which can be viewed as an undirected graph H, possibly with loops. As before, we look first at the conservative case, which means that arbitrary unary relations are available in addition to the binary relation H. In the graph theory community, this situation is described as *list H-colouring*. Formally, the problem #List-H-Col is defined as follows. An instance is a graph G and a collection of colour sets $\mathbf{S} = \{S_v \subseteq Q : v \in V(G)\}$, where $Q = V(H)$. The required output is the number of list H-colourings of $(G, \mathbf{S})$, i.e., the number of mappings $\sigma : V(G) \to Q$ such that $\sigma(v) \in S_v$ for all $v \in V(G)$, and $(\sigma(u), \sigma(v)) \in E(H)$ for all $(u, v) \in E(G)$.

A class of graphs is *hereditary* if it is closed under taking induced subgraphs; the class of bipartite graphs is a simple example. A moment's thought reveals that any maximal class of graphs $\mathcal{H}$ for which #List-H-Col is tractable for $H \in \mathcal{H}$ must be hereditary. On the basis of that general consideration, we expect hereditary graph classes to feature in any complexity classification of #List-H-Col. Two graph classes turn out to be important here. There are many equivalent definitions of these, but the matrix characterisation is perhaps easiest to grasp. Say that a 0,1-matrix $A = (a_{i,j} : 0 \leq i < n, 0 \leq j < m)$ has *staircase form* if the 1s in each row are contiguous and the following condition is satisfied: letting $\alpha_i = \min\{j : a_{i,j} = 1\}$ and $\beta_i = \max\{j : a_{i,j} = 1\}$, we require that the sequences (α_i) and (β_i) are non-decreasing. A graph is a *bipartite permutation graph* if the rows and columns of its biadjacency matrix can be (independently) permuted so that the resulting biadjacency matrix has staircase form. A graph is a *proper interval graph* if the rows and columns of its adjacency matrix can be (simultaneously) permuted so that the resulting adjacency matrix has staircase form. The decision version of #List-H-Col was studied by Feder, Hell and Hwang [26], who established a dichotomy between FP and NP-complete. Goldberg and Jerrum prove the following trichotomy for the counting version [30]. Recall that a graph H is reflexive if every vertex has a loop and irreflexive if no vertex has a loop.

▶ **Theorem 9.** *Suppose H is a connected undirected graph, possibly with loops.*

- *If H is an irreflexive complete bipartite graph or a reflexive complete graph then* #List-H-Col *is in* FP.
- *Otherwise, if H is an irreflexive bipartite permutation graph or a reflexive proper interval graph then* #List-H-Col *is* #BIS*-equivalent.*
- *Otherwise,* #List-H-Col *is* #SAT*-equivalent.*

The most interesting part of the proof lies in demonstrating #SAT-hardness in the final case of the theorem. Here, alternative "excluded subgraph" characterisations of the hereditary classes are useful. For example, a graph that is not a bipartite permutation graph must contain either an induced cycle of length other than four, or one of three special graphs. It is enough, then, to verify that each of these possible subgraphs corresponds to a hard list-colouring problem.

In the non-conservative situation, that is to say, the straight graph homomorphism counting problem called #H-Col, the situation becomes more complicated. Formally, #H-Col is defined as follows. An instance is a graph G and the required output is the number of H-colourings of G, i.e., the number of mappings $\sigma : V(G) \to V(H)$ such that

$(\sigma(u), \sigma(v)) \in E(H)$ for all $(u, v) \in E(G)$. For #H-Col we do not have a general complexity classification or even a plausible conjecture. We do, however, have the following complexity lower bound for non-trivial graphs H, due to Galanis, Goldberg and Jerrum [29].

▶ **Theorem 10.** *Let H be a graph, possibly with self-loops but without parallel edges. If every connected component of H is non-trivial (i.e., neither a reflexive complete graph nor an irreflexive complete bipartite graph), then #H-Col is #BIS-hard.*

The proof extends ideas from earlier work of Goldberg, Kelk and Paterson [42] concerning the related problem of *sampling* H-colourings.

There are some quite small graphs, two of them with as few as four vertices, that get in the way of a neat classification in the style of Theorem 9. Take, for example, the reflexive 4-cycle C_4^*. We know from Theorem 10 that #C_4^*-Col is #BIS-hard, but that is all; the problem #C_4^*-Col is not known either to be #BIS-easy or to be #SAT-hard. An extensive exploration of the complexity of #H-Col, undertaken by Kelk [46], suggests a potentially rich classification.

3.3 Partition Functions

By a partition function we mean a counting CSP, #CSP($\mathcal{F}$), where $\mathcal{F}$ is a single binary function, usually, but not necessarily, symmetric. We will assume in this section that the function is symmetric unless explicitly stated otherwise. Note that in the symmetric case, the problem instance can be viewed as an undirected graph. Despite being very restrictive, this special case is important because it covers partition function of spin models in statistical physics. In view of this, we'll use the term *spin model* as a shorthand for a counting CSP of the above form. Recall that a spin model with q spins can be represented by a $q \times q$ interaction matrix A. We say that A is *irreducible* if, for every pair $i, j \in [q]$, there exists an integer t such that $(A^t)_{ij} > 0$. If A is not irreducible then the domain $[q]$ can be partitioned into equivalence classes of interacting spins, and the partition function (1) decomposed into a sum of component partition functions, one for each equivalence class. (This assumes that the instance graph G is connected; the modification for disconnected G is easy.)

3.3.1 The Conservative Case

We look first at the conservative case, which translates to unary functions being freely available. In terms of spin models in physics, "conservative" corresponds to the existence of an applied field.

Let $A = (a_{ij} : 0 \le i, j < q)$ be an $q \times q$ matrix of non-negative reals. Given a graph G and an assignment $\mathbf{h} = (h_v : v \in V(G))$ of unary non-negative real functions to the vertices of G, we are interested in computing the extended partition function

$$Z_A(G, \mathbf{h}) = \sum_{\sigma : V(G) \to [q]} \prod_{\{u,v\} \in E(G)} a_{\sigma(u),\sigma(v)} \prod_{v \in V(G)} h_v(\sigma(v)). \tag{5}$$

Specifically, we would like to know the computational complexity of the following problem.

Name. EvalZ$_c(A)$.
Instance. A graph G and an assignment of unary functions $\mathbf{h} = (h_v : v \in V(G))$ to the vertices of G.
Output. $Z_A(G, \mathbf{h})$, where Z_A is the extended partition function (5).

The subscript "c" in the problem name is intended to indicate "conservative". In the conservative situation, we can restrict attention to irreducible interaction matrices A, since the complexity of computing the partition function Z_A is determined by the maximum complexity of computing $Z_{A'}$ for any block A' of A.

A certain class of spins models have to be treated separately. These are ones in which the spins can be partitioned into two blocks such that two spins can only be adjacent if they occur in different blocks. Such a spin model is called *imprimitive*, while the others are *primitive*. The interaction matrix of an imprimitive model can be written in block form:

$$A = \begin{pmatrix} 0 & B \\ B^\intercal & 0 \end{pmatrix}. \tag{6}$$

The following result is due to by Goldberg and Jerrum [39]. Although we'll be encountering a more general result later, this one has the advantage of providing an explicit and clearly effective characterisation. We say that matrix A is *log-supermodular* if every 2×2 submatrix has non-negative determinant.

▶ **Theorem 11.**

1. *Suppose A is primitive.*
   - **(a)** *If A has rank 1, then* $\text{EVALZ}_c(A) \in \mathsf{FP}$
   - **(b)** *Otherwise, if there is a simultaneous permutation of the rows and columns of A that renders A log-supermodular, then* $\text{EVALZ}_c(A)$ *is* #BIS*-equivalent.*
   - **(c)** *Otherwise,* $\text{EVALZ}_c(A)$ *is* #SAT*-equivalent.*
4. *Now suppose A is imprimitive. Write A in the form (6).*
   - **(a)** *If B has rank 1, then* $\text{EVALZ}_c(A) \in \mathsf{FP}$.
   - **(b)** *Otherwise, if there are independent permutations of the rows and columns of B that render B log-supermodular, then* $\text{EVALZ}_c(A)$ *is* #BIS*-equivalent.*
   - **(c)** *Otherwise,* $\text{EVALZ}_c(A)$ *is* #SAT*-equivalent.*

The log-supermodularity conditions in Theorem 11 are natural generalisations to the weighted situation of the graph-theoretic conditions in Theorem 9. However, it is not the case that one theorem is a generalisation of the other. It is true that Theorem 11 covers a wider range of interaction matrices, but at the same time it permits a wider range of unary functions $\mathbf{h}$ in the problem instance. In fact, Theorem 11 no longer holds if the functions in $\mathbf{h}$ are resticted to take values in $\{0, 1\}$, which is the situation in Theorem 9 [30].

3.3.2 Boolean Domain

As usual, we can say more about domain size two. As we are considering symmetric interactions, the interaction matrix can, after suitable normalisation, be written as $A = \left(\begin{smallmatrix} \beta & 1 \\ 1 & \gamma \end{smallmatrix}\right)$ with $\beta, \gamma \in \mathbb{R}_{\geq 0}$. Also, the problem instance is just an undirected graph G. As well as the weights for pairs of spins given by A, it is quite common to introduce weights for individual spins: 1 for spin 0 and λ for spin 1. The quantity we wish to study is the extended partition function (5) with $A = \left(\begin{smallmatrix} \beta & 1 \\ 1 & \gamma \end{smallmatrix}\right)$, and with h_v given by $h_v(0) = 1$ and $h_v(1) = \lambda$, for all $v \in V(G)$. For future convenience, we define the problem of interest in the general q setting. So letting the domain or set of spins be $Q = [q]$, we model the external field as a function $h : Q \to \mathbb{R}_{\geq 0}$. As usual, A is a $q \times q$ matrix of non-negative reals.

Name. $\text{EVALZ}(A, h)$.
Instance. A graph G.
Output. $Z_A(G, \mathbf{h})$, where Z_A is the extended partition function (5), and $h_v = h$ for all $v \in V(G)$.

We employ the following conventions: the function $h : Q \to \mathbb{R}_{\geq 0}$ will be specified as a column vector whose ith entry is $h(i)$; also, if h is the all-1 vector, then we omit h from the problem name. With this notation, the problem of immediate interest is $\textsc{EvalZ}\big(\left(\begin{smallmatrix}\beta & 1\\ 1 & \gamma\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\ \lambda\end{smallmatrix}\right)\big)$. Although this problem formulation does not fit the CSP framework exactly, it is natural when viewed from the perspective of spin models with an external field.

Up to this point in our study of the complexity of approximating counting CSPs, the only tractable examples have been trivial. The situation now changes. Jerrum and Sinclair [45] presented an FPRAS for the partition function of the ferromagnetic Ising model, i.e., for $\textsc{EvalZ}\big(\left(\begin{smallmatrix}\beta & 1\\ 1 & \gamma\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\ \lambda\end{smallmatrix}\right)\big)$ in the case $\beta = \gamma \geq 1$ and $\lambda = 1$ In fact, the algorithm they presented works also in the presence of an external field, i.e., for $\textsc{EvalZ}\big(\left(\begin{smallmatrix}\beta & 1\\ 1 & \gamma\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\ \lambda\end{smallmatrix}\right)\big)$, with $\lambda \neq 1$. The reader may wonder how this result may be squared with Theorem 11. The matrix A, after all, has rank 2 and is log-supermodular, so Theorem 11 classifies the partition function as #BIS-equivalent. To resolve the paradox, note that the #BIS-equivalence result relates to a setting in which different functions h_v can be assigned to different vertices of the instance G. A varying field can be accommodated by the algorithm of [45] provided either spin 0 is always favoured, or spin 1 always favoured. Intractability apparently arises when 0- and 1-favouring fields are mixed. This phenomenon had been investigated earlier: see [36].

For the rest of the section we concentrate on the complexity of $\textsc{EvalZ}\big(\left(\begin{smallmatrix}\beta & 1\\ 1 & \gamma\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\ \lambda\end{smallmatrix}\right)\big)$. An early investigation was carried out by Goldberg, Jerrum and Paterson [41], who mapped out some easy and hard regions in "phase space" $(\beta, \gamma, \lambda) \in \mathbb{R}_{\geq 0}$, but left quite a bit unclassified. To describe the more refined results that followed, we need to introduce a further parameter Δ, which is a uniform upper bound on the degrees of vertices of the instance graph G. We start our survey with the independent set or "hard-core" model, whose interaction matrix is $A_{\text{IS}} = \left(\begin{smallmatrix}1 & 1\\ 1 & 0\end{smallmatrix}\right)$. After the Ising model, it is perhaps the most intensively studied spin model. Note that the partition function we are required to evaluate is $Z^{\lambda}_{\text{IS}}(G) = \sum_{\sigma} \lambda^{|\sigma|}$, where the sum ranges over all independent sets σ of G, and $|\sigma| = |\sigma^{-1}(1)|$ denotes the size of the independent set σ.

Weitz [57] proved the following surprising and very influential result.

▶ **Theorem 12.** *Let $\lambda_c = (\Delta-1)^{\Delta-1}/(\Delta-2)^{\Delta}$. There is an FPRAS for $\textsc{EvalZ}\big(\left(\begin{smallmatrix}1 & 1\\ 1 & 0\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\ \lambda\end{smallmatrix}\right)\big)$ restricted to graphs of maximum degree Δ, when $\lambda < \lambda_c$.*

To appreciate the result, it is important to understand the significance of the critical value λ_c. Given a finite graph G, there is a natural probability distribution on independent sets on G that assigns probability $\lambda^{|\sigma|}/Z^{\lambda}_{\text{IS}}(G)$ to each independent set σ. Let $\mathbb{T}_{\Delta,\ell}$ denote the Δ-regular tree with root r and depth ℓ. For each ℓ fix some boundary configuration $\tau_\ell : \partial\,\mathbb{T}_{\Delta,\ell} \to \{0,1\}$ on the the leaves $\partial\,\mathbb{T}_{\Delta,\ell}$ of $\mathbb{T}_{\Delta,\ell}$. If $\lambda < \lambda_c$ then $\Pr(\sigma(r) = 1)$ (i.e., the probability that the root r of the tree is in the independent set σ) tends to a limit, as $\ell \to \infty$, independently of the sequence of boundary conditions $(\tau_\ell : \ell \in \mathbb{N})$. If $\lambda > \lambda_c$, then the limit does not exist.

Since the ideas used to prove Theorem 12 have been influential, we provide a sketch of Weitz's approach here. Unlike previous approaches via Markov chain simulation, his approach leads to a *deterministic* approximation algorithm, technically a *Fully Polynomial-Time Approximation Scheme* or FPTAS. The formal definition of FPRAS is similar to that of FPRAS, except that the algorithm is deterministic, and the result is always within relative error $1 \pm \varepsilon$, rather than merely with high probability. Weitz's FPTAS for estimating the partition function $Z^{\lambda}_{\text{IS}}(G)$ is based on an ingenious recursive algorithm for computing the probability that vertex v is occupied in a randomly chosen independent set in G. If this probability p_v can be estimated to sufficient accuracy then the partition function $Z^{\lambda}_{\text{IS}}(G)$ can

be estimated recursively, by estimating the partition function $Z^{\lambda}_{\mathrm{IS}}(G-v)$ of the graph G with vertex v and incident edges removed, and multiplying that quantity by $(1-p_v)^{-1}$. Note that $p_v \leq \lambda/(\lambda+1)$, so this multiplicative factor is not too sensitive to errors in the evaluation of p_v.

Now we look at the same computation in a different way. We define a self-avoiding walk tree $T_{\mathrm{saw}}(G,v)$ whose vertices correspond to self-avoiding walks in G starting at vertex v. The root r of the tree corresponds to the self-avoiding walk of length 0, and the edges of the tree to extensions of a walk of length ℓ by one edge to a walk of length $\ell+1$. Since G is finite, so is the tree $T_{\mathrm{saw}}(G,v)$. Also, the degrees of vertices in the tree are bounded by Δ. Another ingenious ingredient in this approach is the rule for setting the boundary condition at the leaves of $T_{\mathrm{saw}}(G,v)$. A leaf arises when a self-avoiding walk loops back on itself, and the boundary condition in some sense encodes the cycle structure of G.

The probability that the root r is occupied in a randomly chosen independent set in $T_{\mathrm{saw}}(G,v)$ is easily computed using a simple recursive algorithm based on the inductive structure of the tree. The crucial observation is that, provided the boundary condition for $T_{\mathrm{saw}}(G,v)$ is set correctly, this recursive algorithm on the tree goes through exactly the same sequence of operations as the more complex recursive algorithm on the graph G alluded to earlier. The upshot is that we can compute the occupation probability p_v for vertex v in the graph G by computing the occupation probability of the root r of the tree $T_{\mathrm{saw}}(G,v)$.

We are not done, because the number of self-avoiding walks in G starting from v is exponential in $n = |V(G)|$. So although the self-avoiding walk tree is finite, it is nevertheless exponentially large in n. At this point we use the fact that $\lambda < \lambda_c$. When this condition holds, correlations in the tree decay exponentially fast, and the influence of vertices at depth greater than $c \ln n$ becomes small enough to be ignored, without altering the computed occupation probability of the root by too much. As a consequence, the recursive procedure for evaluating the occupation probability can be truncated at depth $O(\log n)$, while retaining adequate accuracy. This description necessarily skates over all the details, and even omits completely some critical issues.

One of those issues is the distinction between weak and strong spatial mixing. It is sufficiently important that we need to give some brief notes here. Earlier, we informally described a property that the sequence of trees $\mathbb{T}_{\Delta,\ell}$ might possess, namely the occupation probability of the root tends to a limit as $\ell \to \infty$, independently of the sequence of boundary conditions τ_ℓ. This property is *weak spatial mixing*. Roughly speaking, the property of *strong spatial mixing* obtains if the limit continues to exist even if the configuration σ (in this case an independent set) is fixed on some of the internal vertices of the trees.

Weitz's technique was extended by other authors. Sinclair, Srivastava and Thurley [52] considered the antiferromagnetic Ising model with a constant field on a graph of maximum degree Δ. Formally, they were interested in approximating $\textsc{EvalZ}\left(\left(\begin{smallmatrix}\beta & 1\\ 1 & \beta\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\ \lambda\end{smallmatrix}\right)\right)$ when $\beta < 1$ and $\lambda > 0$, and the instance graph G has maximum degree Δ. For some critical value $\lambda_c(\beta,\Delta)$, we say that of β and λ are in the uniqueness region of the regular tree of degree Δ if either $\beta \geq \frac{\Delta-2}{\Delta}$, or $\beta < \frac{\Delta-2}{\Delta}$ and $\max\{\lambda,\lambda^{-1}\} > \lambda_c(\beta,\Delta)$. The critical value λ_c is determined by the existence of a fixed point to a certain recursion. (Determining λ_c is a contribution on the paper.) In the interior of the uniqueness region, the trees $(\mathbb{T}_{\Delta,\ell} : \ell \in \mathbb{N})$ with degree Δ exhibit the decay of correlations phenomenon known as weak spatial mixing, which we saw earlier in the case of the independent set model. (Outside of the uniqueness region, decay of correlations does not occur.) An important step in the argument is showing that weak spatial mixing implies strong spatial mixing. Then Weitz's self avoiding tree leads to:

▶ **Theorem 13.** *If $\beta < 1$ and $\lambda > 0$ are in the interior of the uniqueness region of the infinite regular tree of degree Δ, then there is a FPTAS for* $\textsc{EvalZ}\left(\left(\begin{smallmatrix}\beta & 1\\ 1 & \beta\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\ \lambda\end{smallmatrix}\right)\right)$ *restricted to graphs of degree at most Δ.*

The algorithms in Theorems 12 and 13 have rather natural limits of validity, and it is reasonable to ask whether matching intractability results can be found. Outside of the uniqueness region, we do not have decay of correlations, which leaves open the possibility that we can construct gadgets of maximum degree Δ in which the spins are correlated at the global (or "macroscopic") level. Consider a regular bipartite graph B_Δ of degree Δ that is locally tree-like, a natural choice being a uniform random such graph. If $\Delta = 6$ then $\lambda_c < 1$ and we are outside the tree uniqueness region when $\lambda = 1$. This observation suggests that B_Δ may exhibit correlation at a global level. What we expect to happen is that a typical independent set will be asymmetric: a definite majority of the vertices in the independent set will accumulate on the left or right side of the bipartition of B_Δ. We can then plausibly use B_Δ as a bistable gadget in a reduction from an NP-hard decision or optimisation problem, to the problem $\textsc{EvalZ}\left(\left(\begin{smallmatrix}1 & 1\\ 1 & 0\end{smallmatrix}\right)\right)$ (evaluating the partition function of the independent set model at $\lambda = 1$). In a rather basic form, this programme was carried through by Dyer, Frieze and Jerrum [18], to show that approximating $\textsc{EvalZ}\left(\left(\begin{smallmatrix}1 & 1\\ 1 & 0\end{smallmatrix}\right)\right)$ is #SAT-equivalent when $\Delta \geq 25$. In other words, there is no FPRAS for counting independent sets in a graph of maximum degree 25, unless $\mathsf{RP} = \mathsf{NP}$.

Of course, 25 is a long way from 6. Using much more delicate arguments, Mossel, Weitz and Wormald [50] proved a negative result for λ just above the critical value λ_c of Theorem 12; specifically they showed that local Markov chain Monte Carlo algorithms for evaluating $\textsc{EvalZ}\left(\left(\begin{smallmatrix}1 & 1\\ 1 & 0\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\ \lambda\end{smallmatrix}\right)\right)$ have exponential mixing time. Developing this theme, Sly and Sun [53] (see also Galanis, Štefankovič, Vigoda [28]), proved a general intractability result (i.e., one not restricted to a particular algorithmic technique, but conditional on standard complexity theoretic assumptions).

▶ **Theorem 14.** *The problem* $\textsc{EvalZ}\left(\left(\begin{smallmatrix}\beta & 1\\ 1 & \gamma\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\ \lambda\end{smallmatrix}\right)\right)$ *restricted to graphs of maximum degree Δ is* #SAT*-equivalent in either of the following cases:*

- [The independent set model.] *$\beta = 1$, $\gamma = 0$ and $\lambda > \lambda_c = (\Delta - 1)^{\Delta-1}/(\Delta - 2)^{\Delta}$.*
- [The antiferromagnetic Ising model.] *$\beta = \gamma < 1$, and β and λ are outside of the uniqueness region of the Δ-regular tree.*

The complexity classification of antiferromagnetic two-spin systems, i.e., satisfying $\beta\gamma < 1$, culminates with the work of Li, Lu and Yin [47]. They show the following result, where, by convention, $\Delta = \infty$ indicates that there is no upper bound on vertex degree.

▶ **Theorem 15.** *Suppose $\beta\gamma < 1$ and $\Delta \geq 3$ or $\Delta = \infty$. Suppose also that, for all $\Delta' \leq \Delta$, the parameters (β, γ, λ) lie in the interior of the uniqueness region of the infinite Δ'-regular tree. Then there exists an FPTAS for the problem* $\textsc{EvalZ}\left(\left(\begin{smallmatrix}\beta & 1\\ 1 & \gamma\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\ \lambda\end{smallmatrix}\right)\right)$ *restricted to graphs of maximum degree at most Δ.*

Combined with the negative results of Sly and Sun [53], this essentially completes the analysis of antiferromagnetic two-spin models, except at the boundary of the uniqueness region. We have a dichotomy between models that admit a FPTAS and those which are #SAT-equivalent, and everything is down to the uniqueness condition on regular trees of the appropriate degrees. It should be remembered, however, that we have restricted our attention to symmetric models, i.e., ones where the instance is an *undirected* graph, and the interaction matrix is symmetric. The non-symmetric situation is currently too complex to analyse completely.

In the absence of an external field (i.e., when $\lambda = 1$), the complexity of ferromagnetic models (i.e., those with $\beta\gamma \geq 1$), is easy to describe: they all admit an FPRAS by reduction to the ferromagnetic Ising model with a consistent field [41, 45]. However, if $\beta > \gamma$ and $\lambda > 1$ (or $\beta < \gamma$ and $\lambda < 1$) then a tension arises between the interactions between sites, which tend to pull in one direction, and the action of the field, which tends to pull in the other. How this tension resolves itself is not completely understood, but Liu, Lu and Zhang [49] and Guo and Lu [43] have extracted a great deal of information. It is reasonable to conjecture that there is a dichotomy, with all spin models either admitting an FPRAS or being #BIS-equivalent.

Finally, there in another way in which essentially ferromagnetic models can arise which exhibit the tension alluded to above, namely by restricting an antiferromagnetic model to a bipartite graph. Although we could in principle treat these by inverting the role 0 and 1 in one side of the bipartition, we would then lose symmetry, which, as we observed, is currently fatal. In fact, there is a dichotomy for bipartite antiferromagnetic models between spin models that admit an FPRAS and those that are #BIS-equivalent, as was shown by Cai, Galanis, Goldberg, Guo, Jerrum, Štefankovič and Vigoda [12].

3.3.3 Domain Size Greater Than Two

We have covered the conservative situation. So now suppose a symmetric $q \times q$ interaction matrix A is given, and we want to know the complexity of approximating $\textsc{EvalZ}(A)$, i.e, the complexity of computing an approximation to partition function $Z_A(G)$ defined in (1). In the Boolean case, there is a natural distinction between ferromagnetic ($\beta\gamma > 1$) and antiferromagnetic ($\beta\gamma < 1$) models. When $q > 2$ it is less clear what these terms should mean. Since A is symmetric, we know its eigenvalues are real. Suppose further that A is irreducible. By the Perron-Frobenius theorem, A has at least one positive eigenvalue. Galanis, Štefankovič and Vigoda say that a model is antiferromagnetic if all the other eigenvalues are negative.

The q state Potts model with interaction matrix

$$A^{q,B}_{\text{Potts}} = \begin{pmatrix} B & 1 & \cdots & 1 \\ 1 & B & & 1 \\ \vdots & & \ddots & \\ 1 & 1 & & B \end{pmatrix} \in \mathbb{R}^{q\times q}_{\geq 0},$$

is antiferromagnetic under this, or any other reasonable definition of the term, when $B < 1$. $\textsc{EvalZ}(A^{q,B}_{\text{Potts}})$ is #SAT-hard by a rather direct reduction from maximum q-way cut in a graph, which is an NP-hard optimisation problem. However, we can discuss, as we did in the case $q = 2$, the computational complexity of approximating $\textsc{EvalZ}(A^{q,B}_{\text{Potts}})$, for restricted instances of degree at most Δ. Galanis, Štefankovič and Vigoda [32] prove the following.

▶ **Theorem 16.** *Suppose $q \geq 4$ is even, $\Delta > q$ and $0 \leq B < (\Delta - q)/\Delta$. Then $\textsc{EvalZ}(A^{q,B}_{\text{Potts}})$, restricted to graphs of degree at most Δ, is #SAT-equivalent.*

The reduction employed in proving this result again employs random Δ-regular bipartite graphs as bistable gadgets. The condition for these gadgets to have distinguishable "phases" relates to a certain threshold in an infinite regular tree of degree Δ. However the picture is more complicated than in the case $q = 2$, and there is more than one critical value of B. The specific threshold that is relevant to Theorem 16 is the uniqueness threshold for "semi-translation-invariant measures". These are invariant measures on an infinite regular tree of degree Δ that are invariant under automorphisms of the tree that move the root a

distance of two. Proving that the gadgets have the appropriate bistability property below the threshold is challenging. Some ingenious devices are introduced to simplify the technical details of the proof, but the paper still runs to 60 pages.

Theorem 16 provides a natural boundary beyond which the partition function of the antiferromagnetic Potts model is hard to approximate. Unlike the $q = 2$ case, we don't know whether we can approach the boundary arbitrary closely from the other side. This is because Weitz's approach has not so far been generalised to $q > 2$. As an illustration of the gap, in the special case $B = 0$, we have intractability when $q < \Delta$, but the best general positive result requires $q > \frac{11}{6}\Delta$ [56].

Galanis, Štefankovič, Vigoda and Yang [31] say that a model is ferromagnetic if the interaction matrix A is positive definite. An example is, of course, the ferromagnetic Potts model defined by the interaction matrix $A^{q,B}_{\text{Potts}}$ with $q \geq 2$ and $B > 1$. When $q = 2$, we know that an FPRAS exists [45]. In contrast, Goldberg and Jerrum [38] provide evidence of computational intractability when $q > 2$.

▶ **Theorem 17.** $\textsc{EvalZ}(A^{q,B}_{\text{Potts}})$ *is* #BIS*-hard, for all* $q \geq 3$ *and* $B > 1$.

What is the essential difference between the $q = 2$ and $q > 2$ situations that explains apparent switch from tractability to intractability? In both situations, there is a phase transition from a disordered to an ordered phase as B increases. However, the nature of that transition is different when $q > 2$ than when $q \leq 2$. This difference can be appreciated by looking at typical configurations of the Potts model on a complete graph when B is a little below and a little above the critical value, which we'll call B_o. Configurations are assignments $\sigma : V(G) \to [q]$ of spins to the vertices of G, and they occur with probability implicitly given by (1). Suppose we observe the fraction of vertices that are assigned the majority spin. For $B < B_o$, this fraction is roughly q^{-1} (the "disordered phase") but when $B > B_o$ it is strictly greater (the "ordered phase").

If we plot the fraction of majority spins as a function of B, we find a discontinuity at B_o: a discontinuity of the derivative when $q = 2$ and of the function itself when $q > 2$. A phase transition of the latter kind is called "first-order". At a first-order phase transition, the disordered and ordered phases coexist, and it is this that allows us to construct a bistable gadget, the two phases coding true and false. It appears that we cannot use such a gadget to code an NP-hard problem, but we can code the problem #BIS [38]. When $q = 2$, the phase transition is "second-order", and does not permit gadget construction.

Actually, using the random cluster formulation of the Potts model, we can make sense of the Potts partition function for non-integer q; with this interpretation, Theorem 17 holds for all $q > 2$. Note that this is best possible, as we noted earlier.

Galanis, Štefankovič, Vigoda and Yang [31] greatly strengthen this result so that it applies to bounded degree graphs. Suppose $q \geq 3$ and $\Delta \geq 3$. Define $B_o = B_o(q, \Delta) = (q-2)/[(q-1)^{1-2/\Delta} - 1]$; the significance of B_o is that it is the point of coexistence of ordered and disordered phases in the infinite regular tree of degree Δ.

▶ **Theorem 18.** $\textsc{EvalZ}(A^{q,B}_{\text{Potts}})$ *is* #BIS*-hard, for all* $q \geq 3$, $\Delta \geq 3$ *and* $B > B_o(q, \Delta)$.

The gadgets used in this proof are again random regular graphs. There are substantial technical hurdles to overcome, particularly in describing the phase transition in a very precise way, and proving rigorously that the description is correct.

The majority of spin models are neither ferromagnetic nor antiferromagnetic in the the sense described above, i.e., the number of negative eigenvalues is in the range $[1, q-2]$. What then? As a test case, we can take the interaction matrix associated with the Widom-Rowlinson

model, namely

$$A_{\mathrm{WR}} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

The eigenvalues of this matrix are 1 and $1 \pm \sqrt{2}$, so the model is neither ferromagnetic nor antiferromagnetic in the technical sense. This matrix A_{WR} fits the second part of Theorem 11, so computing $\textsc{EvalZ}_c(A_{\mathrm{WR}})$ is #BIS-equivalent. Evidently, the model does not allow us to encode a hard partitioning problem, such as maximum cut in a graph, and so does not feel "antiferromagnetic".

On the other hand, if we replace the off-diagonal 1s by 2s, to get the modified matrix A'_{WR} then the eigenvalues are 1 and $1 \pm 2\sqrt{2}$ which is still indeterminate. (Replacing each 1 on the diagonal by $1 + \varepsilon$ would also work.) We are now in situation of the third part of Theorem 11, so that $\textsc{EvalZ}_c(A'_{\mathrm{WR}})$ is #SAT-equivalent. Indeed it is not too difficult to see that the external fields are not really required, so that $\textsc{EvalZ}(A'_{\mathrm{WR}})$ is also #SAT-equivalent. (We just need to extract the latent antiferromagnetic Ising model embedded in the top-left 2×2 submatrix of A'_{WR}, which can be done with standard gadgetry.) This model feels genuinely antiferromagnetic. In summary, if the number of negative eigenvalues of A is in the range $[1, q-2]$ then the spin model with interaction matrix A may exhibit either ferromagnetic or antiferromagnetic characteristics.

3.4 #CSPs in General

Progress has been made towards classifying the complexity of approximating general counting CSPs, but only in the conservative case. Fix a finite domain D, and recall that $\mathcal{U}_k$, for all $k \in \mathbb{N}$, is the class of all functions $D^k \to \mathbb{R}_{\geq 0}$, and that $\mathcal{U} = \cup_{k=0}^{\infty} \mathcal{U}_k$. In particular, $\mathcal{U}_1$ is the set of unary functions, which are given free in the conservative case. Recall also the class of functions LSM that is defined in the case $|D| = 2$ by (4).

To state the main result concerning general counting CSPs, we require some further definitions. Recall the notion of functional clone from §2.3. A set of functions $\mathcal{F}$ is *weakly log-modular* if, for all binary functions $f \in \langle \mathcal{F} \rangle_{\#}$ and elements $a, b \in D$,

$$f(a,a)f(b,b) = f(a,b)f(b,a) \quad \text{or} \quad f(a,a) = f(b,b) = 0 \quad \text{or} \quad f(a,b) = f(b,a) = 0;$$

$\mathcal{F}$ is *weakly log-supermodular* if, for all binary functions $f \in \langle \mathcal{F} \rangle_{\#}$ and elements $a, b \in D$,

$$f(a,a)f(b,b) \geq f(a,b)f(b,a) \quad \text{or} \quad f(a,a) = f(b,b) = 0.$$

Finally, a problem Π is LSM-*easy* if there is a finite set $\mathcal{G} \subset \mathsf{LSM}$ of log-supermodular functions (over the Boolean domain) such that Π is AP-reducible to $\#\mathrm{CSP}(\mathcal{G})$.

Chen, Dyer, Goldberg, Jerrum, Lu, McQuillan and Richerby [14] studied general counting CSPs and found the following classification.

▶ **Theorem 19.** *Let $\mathcal{F} \subseteq \mathcal{U}$ be a set of functions that includes all unary functions $\mathcal{U}_1$.*

- *If $\mathcal{F}$ is weakly log-modular then $\#\mathrm{CSP}(\mathcal{G})$ is in FP for every finite $\mathcal{G} \subset \mathcal{F}$.*
- *If $\mathcal{F}$ is weakly log-supermodular but not weakly log-modular, then $\#\mathrm{CSP}(\mathcal{G})$ is LSM-easy for every finite $\mathcal{G} \subset \mathcal{F}$ and #BIS-hard for some such $\mathcal{G}$.*
- *If $\mathcal{F}$ is weakly log-supermodular but not weakly log-modular and consists of functions of arity at most two, then $\#\mathrm{CSP}(\mathcal{G})$ is #BIS-easy for every finite $\mathcal{G} \subset \mathcal{F}$ and #BIS-equivalent for some such $\mathcal{G}$.*
- *If $\mathcal{F}$ is not weakly log-supermodular, then $\#\mathrm{CSP}(\mathcal{G})$ is #SAT-easy for every finite $\mathcal{G} \subset \mathcal{F}$ and #SAT-equivalent for some such $\mathcal{G}$.*

This theorem is clearly more general than Theorem 11, but the latter provides more insight into the particular counting CSPs (i.e., partition functions) that it covers. Indeed, it is not obvious that the classification provided by Theorem 19 is decidable. However there is a kind of multimorphism underlying weak log-submodularity that can be tested fairly directly, and weak-modularity is essentially equivalent to another condition, known as "balance", that was already known to be decidable.

We saw already (see the comments following Theorem 11) that Theorem 19 does not in general establish a trichotomy. However, it does in the "bijunctive" case where all functions have arity at most 2.

4 Esoterica

Faben and Jerrum [25] considered the complexity of the problem $\oplus H$-Col of computing the parity of the number of H-colourings of a graph. This can be viewed as a counting CSP over $\mathbb{F}_2$, of the form $\#\mathrm{CSP}(\{f\})$, where $f : D^2 \to \mathbb{F}_2$ is a symmetric binary function. Define $\oplus\mathsf{P}$ to be the class of functions $\Sigma^* \to \{0,1\}$ that can be expressed as the number of accepting computations of a polynomial-time nondeterministic Turing machine, reduced modulo 2. It is tempting to conjecture that $\oplus H$-Col exhibits a dichotomy between FP and $\oplus\mathsf{P}$-complete. However, the dichotomy here, if it exists, has a very different flavour to conventional counting CSPs.

In order to understand the possible nature of the dichotomy, we introduce a reduction system on undirected graphs in which a single transition has the following form. Suppose H is an undirected graph, possibly with loops. If π is an involution of H (automorphism of order 2), remove from H all vertices that are moved by π and denote the resulting graph by H^π. Then $H \to H^\pi$ is a possible transition of the system. If H has no involution, then no transition from H is possible; in this case, H is a normal form. This reduction system is confluent, that is to say, for each H there is a unique normal form H_0 such that $H \to^* H_0$. where $\to^*$ is the transitive closure of the reduction relation $\to$. Call a graph trivial if it has zero vertices, one vertex (with or without a loop), or two disconnected vertices, one with a loop and one without. Suppose H is a graph and H_0 is its normal form. It is easy to show that $\oplus H$-Col in FP if H_0 is trivial. Faben and Jerrum conjecture that $\oplus H$-Col is $\oplus\mathsf{P}$-complete if H_0 is not trivial, and confirm the conjecture in the special case that H is a tree.

The conjecture for general graphs is still open. However, Göbel, Goldberg and Richerby confirm the conjecture for cactus graphs [33] and square-free graphs [34]. A graph is a *cactus* if every edge is in at most one (simple) cycle. Note that trees are a special case of cactus graphs. A graph is *square-free* if it contains no (not necessarily induced) 4-cycle.

Finally, one can study variants of $\#\mathrm{CSP}(\Gamma)$ in which only minimal (or maximal) satisfying assignments are to be counted. Durand and Hermann consider the problem of "propositional circumscription" [17]. Fix the domain to be $D = \{0,1\}$. A circumscription problem is defined as usual by a constraint language Γ of relations of various arities over D. An instance (X, C) is specified by a set of variables X and constraints C. Instead of counting all satisfying assignments, we are required to count just the minimal such assignments. A satisfying assignment $\sigma : X \to \{0,1\}$ is *mininal* if there does not exist a satisfying assignment $\sigma' \neq \sigma$ such that $\sigma'(x) \leq \sigma(x)$ for all $x \in X$.

The first thing to note is that we are (apparently) no longer working within the complexity class $\#\mathsf{P}$. A non-deterministic polynomial-time Turing machine can guess an assignment $\sigma : X \to \{0,1\}$ and decide whether it is satisfying, but it cannot in general decide whether a

satisfying assignment is minimal. Indeed, Durand and Hermann show that circumscription in general is $\# \cdot \text{coNP}$-complete and hence, presumably, not in $\#\mathsf{P}$. (Roughly, a problem is in $\# \cdot \text{coNP}$ if it is a witness counting problem for which witness checking is in coNP. In this case, deciding whether a satisfying assignment σ is minimal is clearly in coNP.)

However, Durand and Hermann prove that certain circumscription problems are in fact $\#\mathsf{P}$-complete: examples include ones whose constraint language Γ that are bijunctive (all relations in Γ have arity at most two), or that are affine or dual Horn. In contrast, the circumscription problems deriving from constraint languages that are Horn, or that are both affine and bijunctive, are in FP (trivially, in the former case).

Within a similar framework, Goldberg and Jerrum [40] consider the problem of counting satisfying assignments that are locally maximal. The crucial difference with Durand and Hermann lies in the "locally" and not in the "maximal". A satisfying assignment σ is *locally maximal* if any assignment σ' that can be obtained from σ by flipping a single 0 to a 1 is unsatisfying. Local maximality can easily be tested in polynomial time, so we find ourselves again working within the complexity class $\#\mathsf{P}$.

It turns out that counting locally maximal satisfying assignments can sometimes be easier than counting all satisfying assignments, but never harder. One kind of constraint language Γ that is trivially tractable in this variant is one in which all relations $R \in \Gamma$ are monotone (increasing). A relation R of arity k is *monotone* if for all $(x_1, \ldots, x_k) \in R$ and all $i \in [k]$, it is the case that $(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_k) \in R$. Actually, this definition can be relaxed slightly to essentially monotone, while retaining tractability. Let $Z \subseteq [n]$ be the set of indices for which R forces $x_i = 0$; that is, $i \in Z$ if $(x_1, \ldots, x_k) \in R$ implies $x_i = 0$. Then R is *essentially monotone* if it is locally monotone when restricted to the variables $\{x_i : i \in [k] \setminus Z\}$ (and with the variables in Z set to 0).

Goldberg and Jerrum [40] show that the dichotomy for exact counting (Theorem 1) and the trichotomy for approximate counting (Theorem 5) carry over to locally maximal CSPs provided we add an additional case asserting tractability in the case that every relation in Γ is essentially monotone.

Acknowledgements. My understanding of the topic, such as it is, was acquired through collaborations with many people. I therefore thank my coauthors: Andrei Bulatov, Jin-Yi Cai, Xi Chen, Martin Dyer, John Faben, Alan Frieze, Andreas Galanis, Leslie Ann Goldberg, Catherine Greenhill, Martin Grohe, Heng Guo, Pinyan Lu, Colin McQuillan, Mike Paterson, David Richerby, Alistair Sinclair, Daniel Štefankovič, Mark Thurley and Eric Vigoda. Any misunderstandings are my own.

References

1 Elmar Böhler, Nadia Creignou, Steffen Reith, and Heribert Vollmer. Playing with Boolean blocks, part II: Constraint satisfaction problems. *ACM SIGACT Newsletter*, 35:22–35, 2004.

2 Andrei Bulatov and Víctor Dalmau. A simple algorithm for Mal′tsev constraints. *SIAM J. Comput.*, 36(1):16–27 (electronic), 2006. doi:10.1137/050628957.

3 Andrei Bulatov, Martin Dyer, Leslie Ann Goldberg, Markus Jalsenius, and David Richerby. The complexity of weighted Boolean #CSP with mixed signs. *Theoret. Comput. Sci.*, 410(38-40):3949–3961, 2009. doi:10.1016/j.tcs.2009.06.003.

4 Andrei Bulatov and Martin Grohe. The complexity of partition functions. *Theoret. Comput. Sci.*, 348(2-3):148–186, 2005. doi:10.1016/j.tcs.2005.09.011.

5 Andrei A. Bulatov. The complexity of the counting constraint satisfaction problem. *J. ACM*, 60(5):Art 34, 41, 2013. doi:10.1145/2528400.

6 Andrei A. Bulatov. Boolean max-co-clones. *Algebra Universalis*, 74(1-2):139–162, 2015. doi:10.1007/s00012-015-0336-1.

7 Andrei A. Bulatov and Víctor Dalmau. Towards a dichotomy theorem for the counting constraint satisfaction problem. *Inform. and Comput.*, 205(5):651–678, 2007. doi:10.1016/j.ic.2006.09.005.

8 Andrei A. Bulatov, Martin Dyer, Leslie Ann Goldberg, Mark Jerrum, and Colin McQuillan. The expressibility of functions on the Boolean domain, with applications to counting CSPs. *J. ACM*, 60(5):Art. 32, 36, 2013. doi:10.1145/2528401.

9 Jin-Yi Cai and Xi Chen. Complexity of counting CSP with complex weights. In *STOC'12 – Proceedings of the 2012 ACM Symposium on Theory of Computing*, pages 909–919. ACM, New York, 2012. doi:10.1145/2213977.2214059.

10 Jin-Yi Cai, Xi Chen, and Pinyan Lu. Non-negatively weighted #CSP: an effective complexity dichotomy. In *26th Annual IEEE Conference on Computational Complexity*, pages 45–54. IEEE Computer Soc., Los Alamitos, CA, 2011.

11 Jin-Yi Cai, Xi Chen, and Pinyan Lu. Graph homomorphisms with complex values: a dichotomy theorem. *SIAM J. Comput.*, 42(3):924–1029, 2013. doi:10.1137/110840194.

12 Jin-Yi Cai, Andreas Galanis, Leslie Ann Goldberg, Heng Guo, Mark Jerrum, Daniel Štefankovič, and Eric Vigoda. #BIS-hardness for 2-spin systems on bipartite bounded degree graphs in the tree non-uniqueness region. *Journal of Computer and System Sciences*, 82(5):690–711, 2016. doi:http://dx.doi.org/10.1016/j.jcss.2015.11.009.

13 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. The complexity of complex weighted Boolean #CSP. *J. Comput. System Sci.*, 80(1):217–236, 2014. doi:10.1016/j.jcss.2013.07.003.

14 Xi Chen, Martin Dyer, Leslie Ann Goldberg, Mark Jerrum, Pinyan Lu, Colin McQuillan, and David Richerby. The complexity of approximating conservative counting CSPs. *J. Comput. System Sci.*, 81(1):311–329, 2015. doi:10.1016/j.jcss.2014.06.006.

15 Nadia Creignou and Miki Hermann. Complexity of generalized satisfiability counting problems. *Inform. and Comput.*, 125(1):1–12, 1996. doi:10.1006/inco.1996.0016.

16 Nadia Creignou, Phokion Kolaitis, and Bruno Zanuttini. Structure identification of Boolean relations and plain bases for co-clones. *J. Comput. System Sci.*, 74(7):1103–1115, 2008. doi:10.1016/j.jcss.2008.02.005.

17 Arnaud Durand and Miki Hermann. On the counting complexity of propositional circumscription. *Inform. Process. Lett.*, 106(4):164–170, 2008. doi:10.1016/j.ipl.2007.11.006.

18 Martin Dyer, Alan Frieze, and Mark Jerrum. On counting independent sets in sparse graphs. *SIAM J. Comput.*, 31(5):1527–1541, 2002. doi:10.1137/S0097539701383844.

19 Martin Dyer, Leslie Ann Goldberg, Catherine Greenhill, and Mark Jerrum. The relative complexity of approximate counting problems. *Algorithmica*, 38(3):471–500, 2004. Approximation algorithms. doi:10.1007/s00453-003-1073-y.

20 Martin Dyer, Leslie Ann Goldberg, Markus Jalsenius, and David Richerby. The complexity of approximating bounded-degree Boolean #CSP. *Inform. and Comput.*, 220/221:1–14, 2012. doi:10.1016/j.ic.2011.12.007.

21 Martin Dyer, Leslie Ann Goldberg, and Mark Jerrum. The complexity of weighted Boolean #CSP. *SIAM J. Comput.*, 38(5):1970–1986, 2008/09. doi:10.1137/070690201.

22 Martin Dyer, Leslie Ann Goldberg, and Mark Jerrum. An approximation trichotomy for Boolean #CSP. *J. Comput. System Sci.*, 76(3-4):267–277, 2010. doi:10.1016/j.jcss.2009.08.003.

23 Martin Dyer and Catherine Greenhill. The complexity of counting graph homomorphisms. *Random Structures Algorithms*, 17(3-4):260–289, 2000. doi:10.1002/1098-2418(200010/12)17:3/4<260::AID-RSA5>3.3.CO;2-N.

24 Martin E. Dyer and David Richerby. An effective dichotomy for the counting constraint satisfaction problem. *SIAM J. Comput.*, 42(3):1245–1274, 2013. doi:10.1137/100811258.

25 John Faben and Mark Jerrum. The complexity of parity graph homomorphism: an initial investigation. *Theory Comput.*, 11:35–57, 2015. doi:10.4086/toc.2015.v011a002.

26 Tomas Feder, Pavol Hell, and Jing Huang. Bi-arc graphs and the complexity of list homomorphisms. *J. Graph Theory*, 42(1):61–80, 2003. doi:10.1002/jgt.10073.

27 Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory. *SIAM J. Comput.*, 28(1):57–104 (electronic), 1999. doi:10.1137/S0097539794266766.

28 Andreas Galanis, Qi Ge, Daniel Štefankovič, Eric Vigoda, and Linji Yang. Improved inapproximability results for counting independent sets in the hard-core model. *Random Structures Algorithms*, 45(1):78–110, 2014. doi:10.1002/rsa.20479.

29 Andreas Galanis, Leslie Ann Goldberg, and Mark Jerrum. Approximately Counting H-Colorings is #BIS-Hard. *SIAM J. Comput.*, 45(3):680–711, 2016. doi:10.1137/15M1020551.

30 Andreas Galanis, Leslie Ann Goldberg, and Mark Jerrum. A complexity trichotomy for approximately counting list H-colourings. *CoRR*, abs/1602.03985, 2016. Extended abstract to appear in Proc. International Colloquium for Automata, Languages and Programming (ICALP), 2016. URL: http://arxiv.org/abs/1602.03985.

31 Andreas Galanis, Daniel Štefankovič, Eric Vigoda, and Linji Yang. Ferromagnetic Potts Model: Refined #BIS-hardness and Related Results. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, volume 28 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 677–691. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2014. doi:10.4230/LIPIcs.APPROX-RANDOM.2014.677.

32 Andreas Galanis, Daniel Štefankovič, and Eric Vigoda. Inapproximability for antiferromagnetic spin systems in the tree nonuniqueness region. *J. ACM*, 62(6):50:1–50:60, December 2015. doi:10.1145/2785964.

33 Andreas Göbel, Leslie Ann Goldberg, and David Richerby. The complexity of counting homomorphisms to cactus graphs modulo 2. *ACM Trans. Comput. Theory*, 6(4):Art. 17, 29, 2014. doi:10.1145/2635825.

34 Andreas Göbel, Leslie Ann Goldberg, and David Richerby. Counting homomorphisms to square-free graphs, modulo 2. *ACM Trans. Comput. Theory*, 8(3):12:1–12:29, May 2016. doi:10.1145/2898441.

35 Leslie Ann Goldberg, Martin Grohe, Mark Jerrum, and Marc Thurley. A complexity dichotomy for partition functions with mixed signs. *SIAM J. Comput.*, 39(7):3336–3402, 2010. doi:10.1137/090757496.

36 Leslie Ann Goldberg and Mark Jerrum. The complexity of ferromagnetic Ising with local fields. *Combin. Probab. Comput.*, 16(1):43–61, 2007. doi:10.1017/S096354830600767X.

37 Leslie Ann Goldberg and Mark Jerrum. Inapproximability of the Tutte polynomial. *Inform. and Comput.*, 206(7):908–929, 2008. doi:10.1016/j.ic.2008.04.003.

38 Leslie Ann Goldberg and Mark Jerrum. Approximating the partition function of the ferromagnetic Potts model. *J. ACM*, 59(5):Art. 25, 31, 2012. doi:10.1145/2371656.2371660.

39 Leslie Ann Goldberg and Mark Jerrum. A complexity classification of spin systems with an external field. *Proceedings of the National Academy of Sciences*, 112(43):13161–13166, 2015. doi:10.1073/pnas.1505664112.

40 Leslie Ann Goldberg and Mark Jerrum. The complexity of counting locally maximal satisfying assignments of Boolean CSPs. *Theoret. Comput. Sci.*, 634:35–46, 2016. doi:10.1016/j.tcs.2016.04.008.

41 Leslie Ann Goldberg, Mark Jerrum, and Mike Paterson. The computational complexity of two-state spin systems. *Random Structures Algorithms*, 23(2):133–154, 2003. doi:10.1002/rsa.10090.

42 Leslie Ann Goldberg, Steven Kelk, and Mike Paterson. The complexity of choosing an H-coloring (nearly) uniformly at random. *SIAM J. Comput.*, 33(2):416–432 (electronic), 2004. doi:10.1137/S0097539702408363.

43 Heng Guo and Pinyan Lu. Uniqueness, spatial mixing, and approximation for ferromagnetic 2-spin systems. *CoRR*, abs/1511.00493, 2015. URL: http://arxiv.org/abs/1511.00493.

44 Heng Guo and Tyson Williams. The complexity of planar Boolean #CSP with complex weights. In *Automata, languages, and programming. Part I*, volume 7965 of *Lecture Notes in Comput. Sci.*, pages 516–527. Springer, Heidelberg, 2013. doi:10.1007/978-3-642-39206-1_44.

45 Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM J. Comput.*, 22(5):1087–1116, 1993. doi:10.1137/0222066.

46 Steven Kelk. *On the relative complexity of approximately counting H-colourings.* PhD thesis, Warwick University, 2003.

47 Liang Li, Pinyan Lu, and Yitong Yin. Correlation decay up to uniqueness in spin systems. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 67–84. SIAM, Philadelphia, PA, 2012. Full version available at arXiv:1111.7064.

48 Rudolf Lidl and Harald Niederreiter. *Finite fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, second edition, 1997. With a foreword by P. M. Cohn.

49 Jingcheng Liu, Pinyan Lu, and Chihao Zhang. The complexity of ferromagnetic two-spin systems with external fields. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, volume 28 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 843–856. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2014. doi:10.4230/LIPIcs.APPROX-RANDOM.2014.843.

50 Elchanan Mossel, Dror Weitz, and Nicholas Wormald. On the hardness of sampling independent sets beyond the tree threshold. *Probab. Theory Related Fields*, 143(3-4):401–439, 2009. doi:10.1007/s00440-007-0131-9.

51 J. Scott Provan and Michael O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Comput.*, 12(4):777–788, 1983. doi:10.1137/0212053.

52 Alistair Sinclair, Piyush Srivastava, and Marc Thurley. Approximation algorithms for two-state anti-ferromagnetic spin systems on bounded degree graphs. *J. Stat. Phys.*, 155(4):666–686, 2014. doi:10.1007/s10955-014-0947-5.

53 Allan Sly and Nike Sun. Counting in two-spin models on d-regular graphs. *Ann. Probab.*, 42(6):2383–2416, 2014. doi:10.1214/13-AOP888.

54 Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979. doi:10.1137/0208032.

55 Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theoret. Comput. Sci.*, 47(1):85–93, 1986. doi:10.1016/0304-3975(86)90135-0.

56 Eric Vigoda. Improved bounds for sampling colorings. *J. Math. Phys.*, 41(3):1555–1569, 2000. doi:10.1063/1.533196.

57 Dror Weitz. Counting independent sets up to the tree threshold. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 140–149. ACM, New York, 2006. doi:10.1145/1132516.1132538.