The Complexity of Partition Functions

Andrei Bulatov[1] and Martin Grohe[2]

[1] Computing Laboratory, University of Oxford, Oxford, UK
Andrei.Bulatov@comlab.ox.ac.uk
[2] Institut für Informatik, Humboldt-Universität, Berlin, Germany
grohe@informatik.hu-berlin.de

Abstract. We give a complexity theoretic classification of the counting versions of so-called H-colouring problems for graphs H that may have multiple edges between the same pair of vertices. More generally, we study the problem of computing a weighted sum of homomorphisms to a weighted graph H.
The problem has two interesting alternative formulations: First, it is equivalent to computing the partition function of a spin system as studied in statistical physics. And second, it is equivalent to counting the solutions to a constraint satisfaction problem whose constraint language consists of two equivalence relations.
In a nutshell, our result says that the problem is in polynomial time if the adjacency matrix of H has row rank 1, and #P-complete otherwise.

1 Introduction

This paper has two different motivations: The first is concerned with constraint satisfaction problems, the second with "spin-systems" as studied in statistical physics. A known link between the two are so-called H-colouring problems. Our main result is a complete complexity theoretic classification of the problem of counting the number of solutions of an H-colouring problem for an undirected graph H which may have multiple edges, and actually of a natural generalisation of this problem to weighted graphs H. Translated to the world of constraint satisfaction problems, this yields a classification of the problem of counting the solutions to constraint satisfaction problems for two equivalence relations. Translated to the world of statistical physics, it gives a classification of the problem of computing the partition function of a spin system.

Let us describe our result from each of the different perspectives: Let H be a graph, possibly with multiple edges between the same pair of vertices. An *H-colouring* of a graph G is a homomorphism from G to H. Both the decision problem, asking whether a given graph has an H-colouring, and the problem of counting the H-colourings of a given graph, have received considerable attention [6, 7, 10, 12, 13]. Here we are interested in the counting problem. Dyer and Greenhill [6] gave a complete complexity theoretic classification of the counting problem for undirected graphs H without multiple edges; they showed that the problem is in polynomial time if each connected component of H is complete bipartite without any loops or is complete with all loops present, and #P-complete otherwise. Here we are interested in counting H-colourings for graphs H that may have multiple edges. Note that, as opposed to the decision problem, multiple edges do make a difference for the counting problem. Let H be a graph with vertex set $\{1, \ldots, k\}$. H is best described in terms of its adjacency matrix $A = (A_{ij})$, where A_{ij} is the number of edges between vertices i and j. Given a graph $G = (V, E)$, we want to compute the number of homomorphisms from G to H. Observe that this number is

$$Z_A(G) = \sum_{\sigma: V \to \{1, \ldots, k\}} \prod_{e=\{u,v\} \in E} A_{\sigma(u)\sigma(v)}. \tag{1}$$

Borrowing from the physics terminology, we call Z_A the *partition function* of A (or H). We denote the problem of computing $Z_A(G)$ for a given graph G by $\mathrm{EVAL}(A)$. Of course if we define Z_A as in (1), the problem is not only meaningful for matrices A that are adjacency matrices of graphs, but for arbitrary square matrices A. We may view such matrices as adjacency matrices of weighted graphs. We call a symmetric matrix A *connected* (*bipartite*) if the corresponding graph is connected (*bipartite*, respectively).

We prove the following classification result:

Theorem 1. *Let A be a symmetric matrix with non-negative real entries.*

1. *If A is connected and not bipartite, then* $\mathrm{EVAL}(A)$ *is in polynomial time if the row rank of A is at most* 1*; otherwise* $\mathrm{EVAL}(A)$ *is #P-complete.*
2. *If A is connected and bipartite, then* $\mathrm{EVAL}(A)$ *is in polynomial time if the row rank of A is at most* 2*; otherwise* $\mathrm{EVAL}(A)$ *is #P-complete.*
3. *If A is not connected, then* $\mathrm{EVAL}(A)$ *is in polynomial time if each of its connected components satisfies the corresponding condition stated in (1) or (2); otherwise* $\mathrm{EVAL}(A)$ *is #P-complete.*

Note that this generalises Dyer and Greenhill's [6] classification result for graphs without multiple edges, whose adjacency matrices are symmetric 0-1 matrices.

Our proof builds on interpolation techniques similar to those used by Dyer and Greenhill, recent results on counting the number of solutions to constraint satisfaction problems due to Dalmau and the first author [1], and a considerable amount of polynomial arithmetic. Even though we present the proof in the language of constraint satisfaction problems here, in finding the proof it has been very useful to jump back and forth between the H-colouring and constraint satisfaction perspective. The complete proof can be found in [4].

Let us now explain the result for constraint satisfaction problems. A *constraint language* Γ on a finite *domain* D is a set of relations on D. An instance of the problem $\mathrm{CSP}(\Gamma)$ is a triple $(V, D, \mathcal{C})$ consisting of a set V of *variables*, the domain D, and a set $\mathcal{C}$ of *constraints* (s, ρ), where, for some $r \geq 1$, $s \in V^r$ and ρ is an r-ary relation in Γ. A *solution* is a mapping $\sigma : V \to D$ such that for each constraint $((v_1, \ldots, v_r), \rho) \in \mathcal{C}$ we have $(\sigma(v_1), \ldots, \sigma(v_r)) \in \rho$. There has been considerable interest in the complexity of constraint satisfaction problems

[17, 15, 8, 2, 3], which has mainly been driven by Feder and Vardi's [8] *dichotomy question*, asking whether for all languages Γ the problem CSP(Γ) is either solvable in polynomial time or NP-complete. A similar dichotomy question can be asked for the problem #CSP(Γ) of counting the solutions for a given instance [5, 1].

We consider constraint languages Γ consisting of two equivalence relations α, β. Suppose that α has k equivalence classes and β has ℓ equivalence classes. Then Γ can be described by a $(k \times \ell)$-matrix $B = (B_{ij})$, where B_{ij} is the number of elements in the intersection of the ith class of α and the jth class of β. We show that, provided that the matrix is "indecomposable" (in a sense made precise in Section 2.1), the problem #CSP(Γ) is in polynomial time if the row rank of B is 1 and $\#P$-complete otherwise. In [1], it has been shown that if #CSP(Γ) is in polynomial time, then Γ has a so-called *Mal'tsev polymorphism*. The result of this paper provides a further necessary condition for Γ to give to a counting problem solvable in polynomial time.

There is also a straightforward extension to "decomposable" matrices. We can generalise the result to *weighted CSP*, where each domain element d carries a non-negative real weight $\omega(d)$. The weight of a solution $\sigma : V \to D$ is defined to be the product $\prod_{v \in V} \omega(\sigma(v))$, and the goal is to compute the weighted sum over all solutions. As an important intermediate step, we even prove our classification result for weights that are polynomials with integer coefficients.

Let us finally explain the connection with statistical physics. Statistical physics explaines properties of sustances, such as gases, liquids or crystals, using probability distribution of certain particular states of the substance. In one of the standard models, a substance is considered as a conglomeration of particles (atoms) viewed as a graph $G = (V, E)$, called also a *lattice*, in which adjacent vertices represent particles interacting in a non-negligible way. Every particle may have one of the k *spins*; the interaction between neighbouring particles can be described by a *spin system*, which is just a $k \times k$-matrix $K = (K_{ij})$. The entry K_{ij} of K corresponds, in a certain way, the energy that a pair of interacting particles, one of which has spin i, the other one has spin j, contributes into the general energy of G. We always assume K to be symmetric. A *configuration* of the system on a graph $G = (V, E)$ is a mapping $\sigma : V \to \{1, \dots, k\}$. The *energy* of σ is the sum $H(\sigma) = \sum_{e=\{u,v\} \in E} K_{\sigma(u)\sigma(v)}$. Then the probability that G has configuration σ is $\frac{1}{Z} \exp(-H(\sigma)/cT)$, where $Z = \sum_\sigma \exp(-H(\sigma)/cT)$ is the *partition function* and T is a parameter of the system (the *temperature*) and c is a constant. As is easily seen, this probability distribution obeys the law "the lower energy a configuration has, the more likely it is". Observe that $Z = Z_A(G)$ for the matrix A with

$$A_{ij} = \exp(-K_{ij}/cT).$$

Thus EVAL(A) is just the problem of computing the partition function for the system described by A. Dyer and Greenhill in [6] dealt with spin systems in which certain configuration are prohibited and the others are uniformly distributed, while our results are applicable to arbitrary spin systems.

Preliminaries

$\mathbb{R}$, $\mathbb{Q}$ and $\mathbb{Z}$ denote the real numbers, rational numbers and integers, respectively, and $\mathbb{Q}[X]$ and $\mathbb{Z}[X]$ denote the polynomial rings over $\mathbb{Z}$ and $\mathbb{Q}$ in an indeterminate X. Throughout this paper, we let $\mathbb{S}$ denote one of these five rings.

For every set S, $S^{m\times n}$ denotes the set of all $m \times n$-matrices with entries from S. For a matrix A, A_{ij} denotes the entry in row i and column j. The row rank of a matrix $A \in \mathbb{S}^{m\times n}$ is denoted by $\mathsf{rank}(A)$. A matrix $A \in \mathbb{S}^{m\times n}$ is *non-negative* (*positive*), if, for $1 \le i \le m, 1 \le j \le n$, the leading coefficient of A_{ij} is non-negative (positive, respectively).

Graphs are always undirected, unless we explicitly call them *directed graphs*. Graphs and directed graphs may have loops and multiple edges. The *degree*, *in-degree*, and *out-degree* of a vertex in a (directed) graph are defined in the obvious way and denoted by $\deg(v)$, $\mathsf{indeg}(v)$, $\mathsf{outdeg}(v)$, respectively.

Our model of real number computation is a standard model, as it is, for example, underlying the complexity theoretic work on linear programming (cf. [11]). We can either assume that the numbers involved in our computations are polynomial time computable or that they are given by an oracle (see [16] for a detailed description of the model). However, our results do not seem to be very model dependent. All we really need is that the basic arithmetic operations are polynomial time computable. Our situation is fairly simple because all real numbers we encounter are the entries of some matrix A, which is always considered fixed, and numbers computed from the entries of A using a polynomial number of arithmetic operations. Instances of the problem EVAL(A) are just graphs, and we do not have to worry about real numbers as inputs of our computations.

2 The Tractable Cases

2.1 Block Decompositions

Let $B \in \mathbb{S}^{k\times\ell}$. A *submatrix* of B is a matrix obtained from B by deleting some rows and columns. For non-empty sets $I \subseteq \{1,\dots,k\}$, $J \subseteq \{1,\dots,\ell\}$, where $I = \{i_1,\dots,i_p\}$ with $i_1 < \dots < i_p$ and $J = \{j_1,\dots,j_q\}$ with $j_1 < \dots < j_q$, B_{IJ} denotes the $(p \times q)$-submatrix with $(B_{IJ})_{rs} = B_{i_r j_s}$ for $1 \le r \le p, 1 \le s \le q$. A *proper submatrix* of B is a submatrix $B' \ne B$.

Definition 2. *Let $B \in \mathbb{S}^{k\times\ell}$.*

1. *A* decomposition *of B consists of two proper submatrices B_{IJ}, $B_{I'J'}$ such that*
   (a) $I = \{1,\dots,k\} \setminus I'$,
   (b) $J = \{1,\dots,\ell\} \setminus J'$,
   (c) $B_{ij} = 0$ for all $(i,j) \in (I \times J') \cup (I' \times J)$.
   B is indecomposable *if it has no decomposition.*
2. *A* block *of B is an indecomposable submatrix B_{IJ} with at least one non-zero entry such that $B_{IJ}, B_{I^{\complement}J^{\complement}}$ with $I^{\complement} = \{1,\dots,k\} \setminus I$ and $J^{\complement} = \{1,\dots,\ell\} \setminus J$ is a decomposition of B.*

Indecomposability may be viewed as a form of "connectedness" for arbitrary matrices. For square matrices there is also a natural graph based notion of connectedness.

Let $A \in \mathbb{S}^{k\times k}$ be a square matrix. A *principal submatrix* of A is a submatrix of the form A_{II} for some $I \subseteq \{1,\dots,k\}$. Instead of A_{II} we just write A_I. The *underlying graph* of A is the (undirected) graph $G(A)$ with vertex set $\{1,\dots,k\}$ and edge set $\{\{i,j\} \mid 1 \leq i,j \leq n \text{ such that } A_{ij} \neq 0\}$. Note that we define $G(A)$ to be an undirected graph even if A is not symmetric.

Definition 3. *Let $A \in \mathbb{S}^{k\times k}$.*

1. *A is* connected *if the graph $G(A)$ is connected.*
2. *A* connected component *of the matrix A is a principal submatrix A_C, where C is the vertex set of a connected component of $G(A)$.*

Lemma 4. *A connected symmetric matrix is either indecomposable or bipartite. In the latter case, the matrix has two blocks corresponding to the two parts of the bipartition.*

There is another useful connection between indecomposability and connectedness. For a matrix $B \in \mathbb{S}^{k\times \ell}$, let

$$\mathsf{bip}(B) = \begin{pmatrix} 0 & B \\ 0 & 0 \end{pmatrix} \in \mathbb{S}^{(k+\ell)\times(k+\ell)}.$$

Note that $\mathsf{bip}(B)$ is the adjacency matrix of a weighted bipartite directed graph. The following lemma is straightforward.

Lemma 5. *Let $B \in \mathbb{S}^{k\times \ell}$ and $A = \mathsf{bip}(B)$. Then for every block B_{IJ} of B there is a connected component A_C of A such that $A_C = \mathsf{bip}(B_{IJ})$, and conversely for every connected component A_C of A there is a block B_{IJ} of B such that $A_C = \mathsf{bip}(B_{IJ})$.*

In particular, B is indecomposable if, and only if, A is connected.

2.2 Partition functions of graphs

Even though our main result is about symmetric matrices and (undirected) graphs, it is useful to generalise partition functions to directed graphs, which we do in the most straightforward way. Let $A \in \mathbb{S}^{k\times k}$ be a square matrix that is not necessarily symmetric and $G = (V,E)$ a directed graph. For every $\sigma : V \to \{1,\dots,k\}$ we let

$$\omega_A(\sigma) = \prod_{(u,v)\in E} A_{\sigma(u)\sigma(v)},$$

and we let

$$Z_A(G) = \sum_{\sigma:V\to\{1,\dots,k\}} \omega_A(\sigma).$$

Note that if A is symmetric, $G = (V,E)$ a directed graph, and G_U the underlying undirected graph, then $Z_A(G_U) = Z_A(G)$. Thus by EVAL(A) we may denote the problem of computing $Z_A(G)$ for a given directed graph, with the understanding that for symmetric A we can always consider the input graph as undirected.

Theorem 6. *Let $A \in \mathbb{R}^{k\times k}$ be a matrix such that each connected component of A has row rank 1. Then* $\mathrm{EVAL}(A)$ *is in polynomial time.*

Proof. Let $A_1, \dots, A_\ell$ be the connected components of A. Then for every graph G with connected components $G_1, \dots, G_m$ we have

$$Z_A(G) = \prod_{i=1}^{m} \sum_{j=1}^{\ell} Z_{A_j}(G_i).$$

Thus without loss of generality we may assume that A is connected.

Then $\mathsf{rank}(A) \le 1$, and thus there are numbers $a_1, \dots, a_k, b_1, \dots, b_k \in \mathbb{R}$ such that for $1 \le i, j \le k$ we have:

$$A_{ij} = a_i \cdot b_j$$

(the b_j can be chosen to be the A_{1j} and $a_i = A_{i1}/A_{11}$). Let $G = (V, E)$ be a directed graph and $\sigma : V \to \{1, \dots, k\}$. Then

$$\omega_A(\sigma) = \prod_{(v,w)\in E} A_{\sigma(v)\sigma(w)} = \prod_{(v,w)\in E} a_{\sigma(v)} b_{\sigma(w)} = \prod_{v\in V} a_{\sigma(v)}^{\mathsf{outdeg}(v)} b_{\sigma(v)}^{\mathsf{indeg}(v)}.$$

Thus

$$Z_A(G) = \sum_{\sigma:V\to\{1,\dots,k\}} \omega_A(\sigma) = \sum_{\sigma} \prod_{v\in V} a_{\sigma(v)}^{\mathsf{outdeg}(v)} b_{\sigma(v)}^{\mathsf{indeg}(v)} = \prod_{v\in V} \sum_{i=1}^{k} a_i^{\mathsf{outdeg}(v)} b_i^{\mathsf{indeg}(v)}.$$

The last term can easily be evaluated in polynomial time. □

Corollary 7. *Let $A \in \mathbb{R}^{k\times k}$ be a symmetric matrix such that each connected component of A either has row rank at most 1 or is bipartite and has row rank at most 2. Then* $\mathrm{EVAL}(A)$ *is in polynomial time.*

Proof. We may assume that A is connected and bipartite with $\mathsf{rank}(A) = 2$. Then there are $k_1, k_2 \ge 1$ such that $k_1 + k_2 = k$ and a matrix $B \in \mathbb{R}^{k_1\times k_2}$ with $\mathsf{rank}(B) = 1$ and

$$A = \begin{pmatrix} 0 & B \\ B^\top & 0 \end{pmatrix}.$$

Let $G = (V, E)$ be a graph. If G is not bipartitite then $Z_A(G) = 0$, therefore, we may assume that G is connected and bipartite, say, with bipartition V_1, V_2. Let G_{12} be the directed graph obtained from G by directing all edges from V_1 to V_2, and let G_{21} be the directed graph obtained from G by directing all edges from V_2 to V_1. Recall that

$$\mathsf{bip}(B) = \begin{pmatrix} 0 & B \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{k\times k}.$$

We have

$$Z_A(G) = Z_{\mathsf{bip}(B)}(G_{12}) + Z_{\mathsf{bip}(B)}(G_{21}).$$

Since $\mathrm{EVAL}(\mathsf{bip}(B))$ is in polynomial time by Theorem 6, this shows that $Z_A(G)$ can be computed in polynomial time. □

3 Weighted Constraint Satisfaction Problems

It will be convenient for us to view constraint satisfaction problems as homomorphism problems (as first suggested by Feder and Vardi [8]). Recall that a *(relational) vocabulary* τ is a set of *relation symbols*, each with a prescribed arity. A *(relational) structure* $\mathcal{A}$ of vocabulary τ consists of a universe A and, for each r-ary relation symbol $\rho \in \tau$, a relation $\rho^{\mathcal{A}} \subseteq A^r$. Observe that a constraint language Γ on a domain D may be viewed as a relational structure $\mathcal{D}$ with universe D. Feder and Vardi call this structure the *template* of the problem $\mathrm{CSP}(\mathcal{D}) = \mathrm{CSP}(\Gamma)$. An instance $(V, D, \mathcal{C})$ of $\mathrm{CSP}(\mathcal{D})$ may be viewed as a structure $\mathcal{P}$ of the same vocabulary as $\mathcal{D}$. The universe of $\mathcal{P}$ is V, and for each relation symbol ρ in the vocabulary we let $\rho^{\mathcal{P}} = \{s \mid (s, \rho^{\mathcal{D}}) \in \mathcal{C}\}$. Then a *solution* is a homomorphism σ from $\mathcal{P}$ to $\mathcal{D}$. Note that with this notation the H-colouring problem simply becomes $\mathrm{CSP}(H)$.

The objective of the counting problem $\#\mathrm{CSP}(\mathcal{D})$ is to count the solutions for a given instance $\mathcal{P}$. We shall now define a weighted version of this problem. Let $\mathcal{D}$ be a template and $\omega : D \to \mathbb{S}$ be a *weight function.* Slightly abusing notation, we also use ω to denote the weight of a solution $\sigma : V \to D$ of $\mathrm{CSP}(\mathcal{D})$ for $\mathcal{V}$; we let

$$\omega(\sigma) = \prod_{v \in V} \omega(\sigma(v)).$$

As usually, V denotes the set of variables of the CSP, that is, the universe of the structure $\mathcal{P}$. We let

$$\mathcal{Z}_{\mathcal{D},\omega}(\mathcal{P}) := \sum_{\sigma} \omega(\sigma)$$

where the sum ranges over all solutions σ of $\mathrm{CSP}(\mathcal{D})$ for $\mathcal{P}$. We denote the problem of computing $\mathcal{Z}_{\mathcal{D},\omega}$ by $\mathrm{WCSP}(\mathcal{D}, \omega)$.

3.1 CSPs with Two Equivalence Relations

For the rest of this paper, we let α and β be binary relation symbols. Let $\mathcal{D} = (D, \alpha^{\mathcal{D}}, \beta^{\mathcal{D}})$ be a structure in which $\alpha^{\mathcal{D}}$ and $\beta^{\mathcal{D}}$ are equivalence relations on D, and let $\omega : D \to \mathbb{S}$ be a weight function. Suppose that the equivalence classes of $\alpha^{\mathcal{D}}$ are $C_1, \ldots, C_k$ and those of $\beta^{\mathcal{D}}$ are $D_1, \ldots, D_\ell$. Let $B = B(\mathcal{D}, \omega) \in \mathbb{S}^{k \times \ell}$ be defined by

$$B_{ij} = \sum_{d \in C_i \cap D_j} \omega(d).$$

The next lemma shows that the function $\mathcal{Z}_{\mathcal{D},\omega}$ only depends on the matrix B.

Lemma 8. *Let $\mathcal{D}, \mathcal{D}'$ be templates with two equivalence relations and $\omega : D \to \mathbb{S}$, $\omega' : D' \to \mathbb{S}$ weight functions. Suppose that $B(\mathcal{D}, \omega) = B(\mathcal{D}', \omega')$. Then $\mathcal{Z}_{\mathcal{D},\omega} = \mathcal{Z}_{\mathcal{D}',\omega'}$.*

The proof is straightforward. □

Conversely, for every matrix $B \in \mathbb{S}^{k \times \ell}$ we define a *canonical template* $\mathcal{D}_B$ and a *canonical weight function* ω_B as follows: The universe of $\mathcal{D}_B$ is $D_B =$

$\{1,\dots,k\} \times \{1,\dots,\ell\}$, the equivalence relation $\alpha^{\mathcal{D}(B)}$ is equality on the first component, and $\beta^{\mathcal{D}(B)}$ is equality on the second component. The weight function $\omega_B : D_B \to \mathbb{S}$ is defined by $\omega_B((i,j)) = B_{ij}$. Then clearly $B = B(\mathcal{D}, \omega)$.

In the following, we write $\mathcal{Z}_B$ instead of $Z_{\mathcal{D}_B,\omega_B}$ and WCSP(B) instead of WCSP$(\mathcal{D}_B, \omega_B)$.

The following useful lemma is an immediate consequence of the definitions.

Lemma 9. *Let $B, B' \in \mathbb{S}^{k\times\ell}$ be such that B' is obtained from B by permuting rows and/or columns. Then $\mathcal{Z}_B = \mathcal{Z}_{B'}$.*

3.2 Back and Forth between CSP and H-colouring

The next lemma shows that weighted CSP for two equivalence relations are equivalent to evaluation problems for weighted bipartite graphs.

Lemma 10. *Let $B \in \mathbb{S}^{k\times\ell}$. Then the problems* WCSP(B) *and* EVAL$(\mathsf{bip}(B))$ *are polynomial time equivalent.*

Proof. Let

$$A = \mathsf{bip}(B) = \begin{pmatrix} 0 & B \\ 0 & 0 \end{pmatrix} \in \mathbb{S}^{(k+\ell)\times(k+\ell)}.$$

Observe that for every directed graph $G = (V, E)$ we have $Z_A(G) = 0$ unless there is a bipartition V_1, V_2 of V such that $E \subseteq V_1 \times V_2$ (that is, all edges are directed from V_1 to V_2). Assuming that there is such a bipartition V_1, V_2, we let $\mathcal{P} = \mathcal{P}(G)$ be the $\{\alpha, \beta\}$-structure with universe E in which $\alpha^{\mathcal{P}}$ is the relation

$$\{(e, e') \in E^2 \mid e \text{ and } e' \text{ have the same endpoint in } V_1\}$$

and $\beta^{\mathcal{P}}$ is the relation

$$\{(e, e') \in E^2 \mid e \text{ and } e' \text{ have the same endpoint in } V_2\}.$$

Note that for every $\sigma : V \to \{1,\dots,k+\ell\}$ we have $\omega_A(\sigma) = 0$ unless $\sigma(V_1) \subseteq \{1,\dots,k\}$ and $\sigma(V_2) \subseteq \{k+1,\dots,\ell\}$.

Recall the definition of the canonical template $\mathcal{D}_B$ and the canonical weight function ω_B. For a mapping $\sigma : V \to \{1,\dots,k+\ell\}$ with $\omega_A(\sigma) \neq 0$, let $\sigma^* : E \to D_B$ be the mapping that maps $e = (u, v)$ with $u \in V_1$, $v \in V_2$ to $(\sigma(u), \sigma(v) - k)$. Observe that σ^* is a solution of CSP$(\mathcal{D}_B)$ for the instance $\mathcal{P}$. Conversely, every solution of CSP$(\mathcal{D}_B)$ for the instance $\mathcal{P}$ is of the form σ^* for some σ with $\omega_A(\sigma) \neq 0$. Furthermore, we have

$$\omega_A(\sigma) = \prod_{(v,w)\in E} B_{\sigma(v)\sigma(w)} = \omega_B(\sigma^*).$$

Thus

$$Z_A(G) = \sum_{\sigma} \omega_A(\sigma) = \sum_{\sigma^*} \omega_B(\sigma^*) = \mathcal{Z}_B(\mathcal{P}).$$

This yields a reduction from EVAL(A) to WCSP(B).

Let $\mathcal{P} = (P, \alpha^{\mathcal{P}}, \beta^{\mathcal{P}})$ be an instance of WCSP(B). Without loss of generality we may assume that $\alpha^{\mathcal{P}}$ and $\beta^{\mathcal{P}}$ are equivalence relations. To see this, just note that every solution of WCSP(B) for $\mathcal{P}$ is also a solution for the instance $(P, \overline{\alpha}^{\mathcal{P}}, \overline{\beta}^{\mathcal{P}})$, where $\overline{\alpha}^{\mathcal{P}}$ and $\overline{\beta}^{\mathcal{P}}$ are the reflexive symmetric transitive closures of $\alpha^{\mathcal{P}}$ and $\beta^{\mathcal{P}}$, respectively. Let $C_1, \ldots, C_k$ be the equivalence classes of $\alpha^{\mathcal{P}}$ and $D_1, \ldots, D_\ell$ the equivalence classes of $\beta^{\mathcal{P}}$. Let $G = (V, E)$ be the directed graph defined as follows: The vertex set is $V = \{1, \ldots, k + \ell\}$, and for $1 \leq i \leq k$, $1 \leq j \leq \ell$ there are $|C_i \cap D_j|$ edges from i to j. It is easy to see that $\mathcal{Z}_B(\mathcal{P}) = Z_A(G)$. This yields a reduction from WCSP(B) to EVAL(A). □

The following corollary is an immediate consequence of the preceding lemma and Lemma 5:

Corollary 11. *Let $B \in \mathbb{R}^{k \times \ell}$ such that every block of B has row rank at most* 1*. Then* WCSP(B) *is in polynomial time.*

The following lemma is needed to derive the hardness part of Theorem 1 from the hardness results on weighted CSP.

Lemma 12. *Let $A \in \mathbb{S}^{k \times k}$. Then* WCSP($A$) *is polynomial time reducible to* EVAL(A).

Proof. Let $A' = \mathsf{bip}(A)$. By Lemma 10, it suffices to prove that EVAL(A') is reducible to EVAL(A).

Let $G = (V, E)$ be a directed graph. If G is not bipartite with all edges directed from one part to the other, then $Z_{A'}(G) = 0$. Therefore, we assume that there is a partition V_1, V_2 of V such that $E \subseteq V_1 \times V_2$. We claim that

$$Z_{A'}(G) = Z_A(G). \tag{2}$$

Note that for every $\sigma' : V \to \{1, \ldots, 2k\}$ with $\omega_{A'}(\sigma) \neq 0$ we have $\sigma(V_1) \subseteq \{1, \ldots, k\}$ and $\sigma(V_2) \subseteq \{k + 1, \ldots, 2k\}$.

For $\sigma : V \to \{1, \ldots, k\}$, let $f(\sigma) : V \to \{1, \ldots, 2k\}$ be defined by $f(\sigma)(v_1) = \sigma(v_1)$ and $f(\sigma)(v_2) = \sigma(v_2) + k$ for all $v_1 \in V_1$, $v_2 \in V_2$. Then $\omega_A(\sigma) = \omega_{A'}(\sigma')$. Moreover, f is one-to-one, and for every $\sigma' : V \to \{1, \ldots, 2k\}$ with $\omega_{A'}(\sigma') \neq 0$ there exists $\sigma : V \to \{1, \ldots, k\}$ such that $\sigma' = f(\sigma)$. This proves (2). □

4 The Main Hardness Theorem

Theorem 13. *Let $B \in \mathbb{S}^{k \times \ell}$ be non-negative such that at least one block of B has row rank at least 2. Then* WCSP(B) *is #P-complete.*

The full proof of Theorem 13 can be found in [4]. A brief outline of the proof will be given in the next subsection.

Note that, combined with Corollary 11, Theorem 13 yields a complete complexity theoretic classification of problems WCSP(B) for non-negative matrices $B \in \mathbb{R}^{k \times \ell}$.

Furthermore Theorem 1 follows easily from Theorem 6, Corollary 7 (for the tractability results) and Lemma 12, Theorem 13 (for the hardness results). Note that there is no contradiction between Theorem 1(2) and Theorem 13, because if the graph $G(A)$ of a symmetric matrix A is bipartite then A is not indecomposable.

4.1 Outline of the Proof

In this subsection we sketch the proof of Theorem 13. Let $B \in \mathbb{S}^{k \times \ell}$ be a non-negative matrix such that at least one block of B has row rank at least 2.

Step 1: From numbers to polynomials. In this first step of the proof we show that we can assume that all positive (i.e. non-zero) entries of B are powers of some indeterminate X. More precisely, we prove that there is a matrix B^* whose positive entries are powers of X such that B^* also has a block of row rank at least 2 and WCSP(B^*) is polynomial time reducible to WCSP(B). The construction is based on a lemma, which essentially goes back to [6], stating that the problem WCSP(B) is equivalent to the problem counting all solutions of a given weight. For simplicity, let us assume here that all entries of B are non-negative integers; additional tricks are required for real matrices. We can use the lemma to filter out powers of a particular prime p from all entries of B. This way we obtain a matrix B' whose positive entries are powers of a prime p. Using a technique which corresponds to "thickening" in the graph context (cf. [14, 6]), we can replace the entries of this matrix by arbitrary powers, and by interpolation we can then replace p by the indeterminate X. This gives us the desired matrix B^*.

From now on, we assume that all positive entries of B are powers of X.

Step 2: Further preparations. Now we employ two results due to [1]. A consequence of the first is a lemma stating that if there are rows i, i' and columns j, j' such that the entries $B_{ij}, B_{ij'}, B_{i'j}$ are non-zero and $B_{i'j'}$ is zero, then WCSP(B) is #P-complete. This implies that we may assume that in every block of B all entries are positive.

The second result due to [1] is that the more general problem where we only count solutions in which the values of some variables are fixed (i.e., solutions extending some fixed partial solutions) is reducible to WCSP(B). This implies that we may assume that B is indecomposable.

Together, the two assumptions imply that all entries of B are positive.

Another simple reduction shows that either (a) each row of B and each column of B contains a 1, or (b) B has principal submatrix in which all entries are 1, and no 1s appear outside of this principal submatrix. From here we branch into cases (a) and (b).

Step 3(a): Separate 1s. We assume that all entries of B are positive and each row of B and each column of B contains a 1. Since we may permute rows and columns of B, we may assume that all diagonal entries of B are 1. It is not hard to see that then we can reduce the problem EVAL(A) for a symmetric non-singular 2×2-matrices to WCSP(B). For such matrices A the problem EVAL(A) is #P-hard. We believe that this is known (implicitly it is underlying [9]), but in absence of a clear reference we give a proof which boils down to a reduction from the problem of counting MAXCUTs of a graph.

Step 3(b): All 1s together. This part of the proof is the hardest, and it is difficult to describe on a high level. We assume that all entries of B are positive and that a principal submatrix in the upper left corner of B contains all 1s. We define a sequence $B^{[k]}$, for $k \geq 1$, of matrices that are obtained from B by some construction on the instances that is remotely similar to "stretching" and "thickening" (cf. [14, 6]), but more complicated. We show that WCSP($B^{[k]}$) is reducible to WCSP(B) for all k.

The entries of the $B^{[k]}$ are polynomials with integer coefficients (no longer just powers of X as the entries of B). Employing a little bit of complex analysis, we prove that for some k, $B^{[k]}_{11}$ has an irreducible factor $p(X)$ such that the multiplicity of $p(X)$ in $B^{[k]}_{11}$ is higher than in all other entries in the first row and column, and the multiplicity in the corresponding diagonal entries is also sufficiently high. Using similar tricks as in Step 1, we can filter out the powers of this irreducible polynomial $p(X)$. We obtain a matrix whose weighted CSP is #P-complete by the results of Steps 2 and 3(a).

5 Conclusions

We give a complete complexity theoretic classification for the problem of evaluating the partition function of a symmetric non-negative matrix A, which may be viewed as the adjacency matrix of an undirected weighted graph H. Our proofs explore a correspondence between this evaluation problem and weighted constraint satisfaction problems for constraint languages with two equivalence relations.

Peculiarly, our proof does not go through for matrices with negative entries. Indeed, we do not know whether the evaluation problem for the matrix

$$\begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}$$

is #P-complete. (Observe that the evaluation problem for this matrix is equivalent to the problem of counting induced subgraphs with an even number of edges.)

The more important open problem is to obtain a classification result for the evaluation problem for non-symmetric matrices, corresponding to directed graphs. We believe that with our results such a classification may now be within reach, in particular because our main hardness result goes through for directed graphs. The ultimate goal of this line of research is a classification of counting and weighted CSP for arbitrary constraint languages. Towards a solution of this problem, one may try to reduce the weighted CSP to evaluation problems for directed graphs. It is interesting to note that the known reduction between the corresponding decision problems does not give a reduction between the counting problems we are interested in here.

Acknowledgement We wish to thank Mark Jerrum for many useful discussions.

References

1. A. Bulatov and V. Dalmau. Towards a dichotomy theorem for the counting constraint satisfaction problem. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science, FOCS'03*, pages 562–571, 2003.
2. A.A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science, FOCS'02*, pages 649–658, 2002.
3. A.A. Bulatov. Tractable conservative constraint satisfaction problems. In *Proceedings of the 18th Annual IEEE Simposium on Logic in Computer Science*, pages 321–330, 2003.
4. A.A. Bulatov and M. Grohe. The complexity of partition functions. Technical Report PRG-RR-04-04, Computing Laboratory, University of Oxford, Oxford, UK, 2004.
5. N. Creignou and M. Hermann. Complexity of generalized satisfiability counting problems. *Information and Computation*, 125(1):1–12, 1996.
6. M. Dyer and C. Greenhill. The complexity of counting graph homomorphisms. *Random Structures and Algorithms*, 17:260–289, 2000.
7. M.E. Dyer, L.A. Goldberg, and M. Jerrum. Counting and sampling H-colourings. In J.D.P. Rolim and S.P. Vadhan, editors, *Proceedings of the 6th International Workshop on Randomization and Approximation Techniques*, volume 2483 of *Lecture Notes in Computer Science*, pages 51–67. Springer-Verlag, 2002.
8. T. Feder and M.Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM Journal of Computing*, 28:57–104, 1998.
9. L.A. Goldberg, M. Jerrum, and M. Paterson. The computational complexity of two-state spin systems. *Random Structures and Algorithms*, 23:133–154, 2003.
10. L.A. Goldberg, S. Kelk, and M. Paterson. The complexity of choosing an H-colouring (nearly) uniformly at random. In *Proceedings of the 34rd ACM Simposium on Theory of Computing*, pages 53–62, 2002.
11. M. Grötschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimazation*. Springer-Verlag, 1993. 2nd edition.
12. P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Ser.B*, 48:92–110, 1990.
13. P. Hell, J. Nešetřil, and X. Zhu. Duality and polynomial testing of tree homomorphisms. *Trans. of the AMS*, 348(4):1281–1297, 1996.
14. F. Jaeger, D.L. Vertigan, and D.J.A. Welsh. On the computational complexity of the Jones and Tutte polynomials. *Mathematical Proceedings of the Cambridge Philosophical Society*, 108:35–53, 1990.
15. P.G. Jeavons, D.A. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, 1997.
16. K. Ko. *Complexity Theory of Real Functions*. Birkhäuser, 1991.

17. T.J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symposium on Theory of Computing*, pages 216–226, 1978.