

A Decidable Dichotomy Theorem on Directed Graph Homomorphisms with Non-negative Weights*

Jin-Yi Cai[†]

Xi Chen[‡]

March 8, 2018

Abstract

The complexity of graph homomorphism problems has been the subject of intense study for some years. In this paper, we prove a *decidable* complexity dichotomy theorem for the partition function of directed graph homomorphisms. Our theorem applies to all non-negative weighted forms of the problem: given any fixed matrix $\mathbf{A}$ with non-negative algebraic entries, the partition function $Z_{\mathbf{A}}(G)$ of directed graph homomorphisms from any directed graph G is *either* tractable in polynomial time *or* $\#P$ -hard, depending on the matrix $\mathbf{A}$. The proof of the dichotomy theorem is combinatorial, but involves the definition of an infinite family of graph homomorphism problems. The proof of its decidability on the other hand is algebraic and based on properties of polynomials.

*We would like to thank the following colleagues for their interest and helpful comments: Martin Dyer, Alan Frieze, Leslie Ann Goldberg, Richard J. Lipton, Pinyan Lu, Mike Paterson, Marc Thurley, and Leslie Valiant. A preliminary version of the paper appeared in Proceedings of the 51st IEEE Annual Symposium on Foundations of Computer Science [6] in 2010.

[†]Computer Science Department, University of Wisconsin-Madison and Beijing University. Research supported by NSF grants CCF-0830488 and CCF-0914969.

[‡]Department of Computer Science, Columbia University. Part of the research done while the author was a postdoc at Princeton University and University of Southern California. Supported by NSF grants CCF-0832797 and DMS-0635607.

1 Introduction

The complexity of counting graph homomorphisms has received much attention [13, 4, 2, 1, 12, 18, 9]. The problem can be defined over both *directed* and *undirected* graphs, and the *directed* version of the problem turns out to be significantly more challenging. In particular, Feder and Vardi showed that the decision problems defined by directed graph homomorphisms are as general as the Constraint Satisfaction Problems (CSPs), and it is known that a complexity dichotomy for the former implies the full dichotomy conjecture for all decision CSPs [16]. While no such implication is known for counting problems, our understanding of the problem over directed graphs is more limited compared to that over undirected graphs.

Let G and H be two graphs. We follow the standard definition of graph homomorphisms, where G is allowed to have multiple edges but no self loops; and H can have both multiple edges and self loops.¹ We say $\xi : V(G) \rightarrow V(H)$ is a graph homomorphism from G to H if $\xi(u)\xi(v)$ is an edge in $E(H)$ for all $uv \in E(G)$. Here if H is an *undirected* graph, then G is also an undirected graph; if H is *directed*, then G is also directed. The undirected problem is a special case of the directed one.

For a fixed H , we are interested in the complexity of the following integer function $Z_H(G)$: The input is a graph G , and the output is the number of graph homomorphisms from G to H . More generally, we can define $Z_{\mathbf{A}}(\cdot)$ for any fixed $m \times m$ matrix $\mathbf{A} = (A_{i,j})$:

$$Z_{\mathbf{A}}(G) = \sum_{\xi: V \rightarrow [m]} \prod_{uv \in E} A_{\xi(u), \xi(v)}, \quad \text{for any directed graph } G = (V, E).$$

Note that the input G is a directed graph in general. However, if $\mathbf{A}$ is a symmetric matrix, then one can always view G as an undirected graph. Moreover, if $\mathbf{A}$ is a $\{0, 1\}$ -matrix, then $Z_{\mathbf{A}}(\cdot)$ is exactly $Z_H(\cdot)$, where H is the graph whose adjacency matrix is $\mathbf{A}$.

Graph homomorphisms can express many interesting counting problems over graphs. For example, if we take H to be an undirected graph over two vertices $\{0, 1\}$ with an edge $(0, 1)$ and a loop $(1, 1)$ at 1, then a graph homomorphism from G to H corresponds to a VERTEX COVER of G , and $Z_H(G)$ is simply the number of vertex covers of G . As another example, if H is the complete graph on k vertices without self loops, then $Z_H(G)$ is the number of k -COLORINGS of G . In [17], Freedman, Lovász, and Schrijver characterized what graph functions can be expressed as $Z_{\mathbf{A}}(\cdot)$.

For increasingly more general families $\mathcal{C}$ of matrices $\mathbf{A}$, the complexity of $Z_{\mathbf{A}}(\cdot)$ has been studied and *dichotomy* theorems have been proved. A *dichotomy* theorem for a given family $\mathcal{C}$ of matrices $\mathbf{A}$ states that for any $\mathbf{A} \in \mathcal{C}$, the problem of computing $Z_{\mathbf{A}}(\cdot)$ is either *in polynomial time* or *#P-hard* (note that all such problems belong to $P^{\#P}$, or more precisely $FP^{\#P}$, the class of functions² computable by a polynomial-time Turing machine with access to a #P oracle). A *decidable* dichotomy theorem further requires that the dichotomy criterion is computably decidable: There is a finite-time classification algorithm that, given any $\mathbf{A}$ in $\mathcal{C}$, decides whether $Z_{\mathbf{A}}(\cdot)$ is in polynomial time or #P-hard. Most results have been obtained for undirected graphs.

Symmetric matrices $\mathbf{A}$, and $Z_{\mathbf{A}}(G)$ over undirected graphs G :

In [20, 21], Hell and Nešetřil showed that given any symmetric $\{0, 1\}$ matrix $\mathbf{A}$, *deciding* whether $Z_{\mathbf{A}}(G) > 0$ is either in P or NP-complete. Then Dyer and Greenhill [13] showed that given any symmetric $\{0, 1\}$ matrix $\mathbf{A}$, the problem of computing $Z_{\mathbf{A}}(\cdot)$ is either in P or #P-complete. Bulatov and Grohe generalized their result to all non-negative symmetric matrices $\mathbf{A}$ [4].³ They obtained an elegant dichotomy theorem

¹Our results are actually stronger in that our tractability result allows for loops in G , while our hardness result holds for G without loops.

²We will abuse the notation slightly and use P to denote polynomial-time computable predicates as well as functions.

³More exactly, they proved a dichotomy theorem for symmetric matrices $\mathbf{A}$ in which every entry $A_{i,j}$ is a non-negative algebraic number. Our result in this paper applies similarly to all non-negative algebraic numbers. Throughout the paper we use $\mathbb{R}$ to denote the set of real *algebraic* numbers and refer to them as real numbers when it is clear from the context.

which basically says that $Z_{\mathbf{A}}(\cdot)$ is in P if every *block* of $\mathbf{A}$ has rank at most one, and is $\#P$ -hard otherwise. In [18] Goldberg, Grohe, Jerrum and Thurley proved a beautiful dichotomy for all symmetric real matrices. Finally a dichotomy theorem for all symmetric complex matrices was proved by Cai, Chen and Lu [9]. We remark that all these dichotomy theorems for symmetric matrices above are *polynomial-time* decidable, meaning that given any matrix $\mathbf{A}$, one can decide in polynomial time (in the input size of $\mathbf{A}$) whether $Z_{\mathbf{A}}(\cdot)$ is in P or $\#P$ -hard.

General matrices $\mathbf{A}$, and $Z_{\mathbf{A}}(G)$ over directed graphs G :

In a paper that received the best paper award at ICALP in 2006, Dyer, Goldberg and Paterson [12] proved a dichotomy theorem for directed graph homomorphism problems $Z_H(\cdot)$ that are restricted to directed *acyclic* graphs H . They introduced the notion of *Lovász-goodness* and proved that $Z_H(\cdot)$ is in P if the graph H is *layered*⁴ and *Lovász-good*, and is $\#P$ -hard otherwise. The property of Lovász-goodness turns out to be polynomial-time decidable.

In 2008, Bulatov [5, 1] obtained a sweeping dichotomy theorem for all unweighted (i.e., $\{0, 1\}$ -valued) counting Constraint Satisfaction Problems ($\#CSP$ for short). Later Dyer and Richerby [14] presented an alternative proof at STOC in 2010. The dichotomy theorem of Bulatov implies a dichotomy for $Z_H(\cdot)$ over directed graphs H but its decidability was unclear since its dichotomy criterion⁵ requires one to check a condition on an infinitary object. The decidability of the dichotomy theorem of Bulatov was left as an open problem in [5]. This was resolved by Dyer and Richerby in their journal version [15] of [14] after the preliminary version of the present paper appeared in [6] (see discussion on “Recent Developments on the Complexity of $\#CSP$ ”). The techniques of Dyer and Richerby generalize to rational weights, but in a more complicated way, via the translation of Bulatov et al. [3]. In contrast, the present paper gives a direct proof that applies to all non-negative algebraic weights; the techniques of Bulatov et al. [3] depend heavily on the weights being rational.

In this paper we prove a dichotomy theorem for the family of all non-negative algebraic matrices. We show that for every fixed $m \times m$ matrix $\mathbf{A}$ with non-negative algebraic entries, the problem of computing $Z_{\mathbf{A}}(\cdot)$ is either in P or $\#P$ -hard. Furthermore, our dichotomy criterion is *decidable*: We present a finite-time algorithm which, given a non-negative and algebraic matrix $\mathbf{A}$, determines whether $Z_{\mathbf{A}}(\cdot)$ is in P or $\#P$ -hard. In particular, for the special family of matrices with $\{0, 1\}$ entries, our result gives an alternative dichotomy criterion⁶ to that of Bulatov [5] and Dyer–Richerby [14], which we show is decidable.

The main obstacle we encountered in obtaining the dichotomy theorem is due to the abundance of new intricate but tractable cases, when moving from acyclic graphs to general directed graphs. For example, H does not have to be *layered* for the problem $Z_H(\cdot)$ to be tractable (see Figure 1 for an example). Because of the generality of directed graphs, it seems impossible to have a simply stated criterion (e.g., Lovász-goodness, as was used in the acyclic case [12]) which is both powerful enough to completely characterize all the tractable cases and also easy to check. However, we manage to find a dichotomy criterion as well as a finite-time algorithm to decide whether $\mathbf{A}$ satisfies it or not.

In particular, the dichotomy theorem of Dyer, Goldberg and Paterson [12] for the acyclic case fits into our framework as follows. In our dichotomy, we start from an $m \times m$ matrix $\mathbf{A}$ and then define, in each round, a (possibly infinite) set of new matrices. The size of the matrices defined in round $i + 1$ is strictly

⁴A directed acyclic graph is *layered* if one can partition its vertices into k sets $V_1, \dots, V_k$, for some $k \geq 1$, such that every edge goes from V_i to V_{i+1} for some $i : 1 \leq i < k$.

⁵A dichotomy criterion is a well-defined mathematical property over the family of matrices $\mathbf{A}$ being considered such that $Z_{\mathbf{A}}(\cdot)$ is in P if $\mathbf{A}$ has this property; and is $\#P$ -hard otherwise.

⁶Both our dichotomy criterion (when specialized to $\{0, 1\}$ matrices) and the one of Bulatov characterize $\{0, 1\}$ matrices $\mathbf{A}$ with $Z_{\mathbf{A}}(\cdot)$ in P and thus, they must be equivalent assuming $P \neq P^{\#P}$, i.e., $\mathbf{A}$ satisfies our criterion if and only if it satisfies the one of Bulatov. As a corollary, our result implies a finite-time algorithm for checking the dichotomy criterion of Bulatov [1] (and the version of Dyer and Richerby [14]) for the case of $\{0, 1\}$ matrices, assuming that $P \neq P^{\#P}$. However, we are not able to prove *unconditionally* that these dichotomy criteria for $\{0, 1\}$ matrices are equivalent.

smaller than that of round i (so there can be at most m rounds). The dichotomy then states that $Z_{\mathbf{A}}(\cdot)$ is in P if and only if every *block* of any matrix defined in the process above is of rank 1 (see Section 1.1 and 1.2 for details). For the special acyclic case treated by Dyer, Goldberg and Paterson in [12], let $\mathbf{A}$ be the adjacency matrix of H which is acyclic and has k layers, then at most k rounds are necessary to reach a conclusion about whether $Z_{\mathbf{A}}(\cdot) = Z_H(\cdot)$ is in P or $\#P$ -hard. The general case is more difficult. For example, let H be a directed graph obtained from a k -layered graph by adding edges from V_k back to V_1 . Then deciding whether $Z_{\mathbf{A}}(\cdot)$ is in P or $\#P$ -hard becomes much harder compared to the original k -layered graph in the sense that we may need many more than k rounds to reach a conclusion.

Recent Developments on the Complexity of $\#CSP$:

A number of new developments have been made on the complexity of $\#CSP$ after a preliminary version of this paper appeared in [6]. First of all, in their journal version of [14], Dyer and Richerby [15] showed that their dichotomy criterion for unweighted $\#CSP$ is indeed decidable in NP. This was extended to $\#CSP$ with non-negative and rational weights by Bulatov, Dyer, Goldberg, Jalsenius, Jerrum and Richerby in [3], and then to $\#CSP$ with non-negative weights by Cai, Chen and Lu in [8], both decidable in NP. In [7], Cai and Chen obtained a complexity dichotomy theorem for $\#CSP$ with complex weights, though the decidability of its dichotomy criterion remains open.

Compared to [8], our dichotomy theorem is weaker: 1) While [8] covers all $\#CSP$ with non-negative weights, ours only covers counting directed graph homomorphisms with non-negative weights, which can be equivalently viewed as a special case of non-negative $\#CSP$ for which the language consists of a single binary function; 2) The criterion of [8] is shown to be not only decidable but in NP. However, we believe that the approach of our dichotomy in this paper is still of interest because of the following reasons: 1) Compared to the recent series of dichotomy theorems for $\#CSP$ [1, 15, 3, 8, 7], the current paper has a more natural and combinatorial approach that is designed specifically for directed graph homomorphisms. This goes for the tractability algorithm and the decidability algorithm, both of which are more germane to the problem of counting directed graph homomorphisms. 2) Our method does not use the machinery of Universal Algebra, which played a critical role in all the $\#CSP$ papers, and thus may find applications when Universal Algebra is not known to be applicable such as Holant problems [10, 11]. 3) Assuming that $P \neq P^{\#P}$, our dichotomy criterion is equivalent to that of Bulatov [1] (and that of Dyer and Richerby [15]) over $Z_{\mathbf{A}}(\cdot)$ with $\{0, 1\}$ -matrices $\mathbf{A}$. However, it remains an open problem to show this equivalence unconditionally. Such a proof may improve our understanding of these criteria and shed new light on the decidability of the dichotomy for $\#CSP$ with complex values [7].

1.1 Intuition of the Dichotomy: Domain Reduction

For an integer $m \geq 1$ we denote $[m] = \{1, \dots, m\}$; for $m = 0$ we denote $[0] = \emptyset$.

Let $\mathbf{A}$ be the $m \times m$ non-negative matrix being considered, and let $G = (V, E)$ be an input directed graph. Before giving a more formal sketch of the dichotomy theorem, we use a simple example to illustrate one of the most important ideas of this work: *domain reduction*.

For this purpose we need to introduce the concept of *labeled directed graphs*. A labeled directed graph $\mathcal{G}$ over domain $[m]$ is a directed graph in which each directed edge e is labeled with an $m \times m$ matrix $\mathbf{A}^{[e]}$ and each vertex v is labeled with an m -dimensional vector $\mathbf{w}^{[v]}$. The partition function of $\mathcal{G}$ is defined as

$$Z(\mathcal{G}) = \sum_{\xi: V \rightarrow [m]} \prod_{v \in V} \mathbf{w}_{\xi(v)}^{[v]} \prod_{uv \in E} \mathbf{A}_{\xi(u), \xi(v)}^{[uv]}.$$

In particular, we have $Z_{\mathbf{A}}(G) = Z(\mathcal{G}_0)$ where $\mathcal{G}_0$ has the same graph structure as G ; every edge of $\mathcal{G}_0$ is labeled with the same $\mathbf{A}$; and every vertex of $\mathcal{G}_0$ is labeled with $\mathbf{1}$, the m -dimensional all-1 vector.

Roughly speaking, starting from the input G , we build (in polynomial time) a *finite* sequence of new *labeled* directed graphs $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_h$ one by one. $\mathcal{G}_{k+1}$ is constructed from $\mathcal{G}_k$ by using the domain

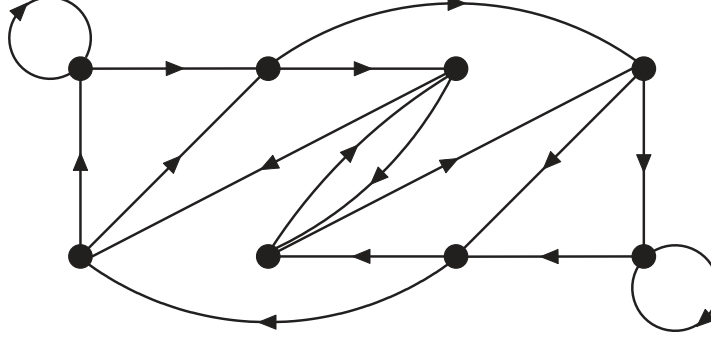


Figure 1: A directed graph H such that $Z_H(\cdot)$ is tractable

reduction method which we are going to describe next. On the one hand, the *domains* of these labeled graphs shrink along with k . This means, the size of the edge weight matrices associated with the edges of $\mathcal{G}_k$ (or equivalently, the dimension of the vectors associated with the vertices of $\mathcal{G}_k$) strictly decreases along with k . On the other hand, we have $Z(\mathcal{G}_{k+1}) = Z(\mathcal{G}_k)$ for all $k \geq 0$ and thus,

$$Z_{\mathbf{A}}(G) = Z(\mathcal{G}_0) = \dots = Z(\mathcal{G}_h).$$

Since the domain size decreases monotonically, the number of graphs $\mathcal{G}_k$ in this sequence is at most m . To prove our dichotomy theorem, we show that, either something bad happens which forces us to stop the domain reduction process, in which case we show that $Z_{\mathbf{A}}(\cdot)$ is $\#P$ -hard; or we can keep reducing the domain size until the computation becomes trivial, in which case we show that $Z_{\mathbf{A}}(\cdot)$ is in P .

$$\mathbf{A} = \begin{pmatrix} A_{1,1} & A_{1,3} & & & & & & \\ A_{2,1} & A_{2,3} & & & & & & \\ & & A_{3,5} & A_{3,7} & & & & \\ & & A_{4,5} & A_{4,7} & & & & \\ & A_{5,2} & A_{5,4} & & & & & \\ & A_{6,2} & A_{6,4} & & & & & \\ & & & & A_{7,6} & A_{7,8} & & \\ & & & & A_{8,6} & A_{8,8} & & \end{pmatrix}$$

Figure 2: The 8×8 block-rank-1 matrix $\mathbf{A}$

We say a non-negative matrix $\mathbf{A}$ is *block-rank-1* if one can separately permute its rows and columns to get a block diagonal matrix in which every block is of rank at most 1. Bulatov and Grohe [4] (see Theorem 2) showed that $Z_{\mathbf{A}}(\cdot)$ is $\#P$ -hard when $\mathbf{A}$ is not block-rank-1. So we assume below that $\mathbf{A}$ is block-rank-1; otherwise the problem is already known to be $\#P$ -hard. As an example, let $\mathbf{A}$ be the 8×8 block-rank-1 non-negative matrix with 16 positive entries as shown in Figure 2, and let

$$\mathcal{T} = \{(A_1, B_1), (A_2, B_2), (A_3, B_3), (A_4, B_4)\}$$

denote its *block structure*, where

$$\forall s \in [4], A_s = \{2s-1, 2s\}, B_1 = \{1, 3\}, B_2 = \{5, 7\}, B_3 = \{2, 4\} \text{ and } B_4 = \{6, 8\},$$

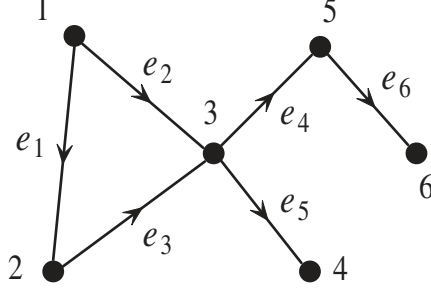


Figure 3: The input directed graph G

so that $A_{i,j} > 0$ if and only if $i \in A_s$ and $j \in B_s$, for some $s \in [4]$. Because $\mathbf{A}$ is block-rank-1, there also exist two 8-dimensional *positive* vectors α and β such that

$$A_{i,j} = \alpha_i \cdot \beta_j, \quad \text{for all } (i,j) \text{ such that } i \in A_s \text{ and } j \in B_s \text{ for some } s \in [4].$$

Now let $G = (V, E)$ be the directed graph in Figure 3, where $|V| = 6$ and $|E| = 6$. We illustrate the *domain reduction* process by constructing the first labeled directed graph $\mathcal{G}_1$ in the sequence as follows. To simplify the presentation, we let $\mathbf{y} \in [8]^6$ (instead of $\xi : V \rightarrow [8]$) denote an assignment, where $y_i \in [8]$ denotes the value of vertex i in Figure 3 for every $i \in [6]$.

First, let $\mathbf{y} \in [8]^6$ be any assignment with a nonzero weight: $A_{y_i, y_j} > 0$ for every edge $ij \in E$. Since $\mathbf{A}$ has the block structure $\mathcal{T}$, for every $ij \in E$, there exists a unique index $s \in [4]$ such that $y_i \in A_s$ and $y_j \in B_s$. This inspires us to introduce a new variable $x_\ell \in [4]$ for each edge $e_\ell \in E$, $\ell \in [6]$ (as shown in Figure 3). For every possible assignment of $\mathbf{x} = (x_1, x_2, \dots, x_6) \in [4]^6$, we use $Y[\mathbf{x}]$ to denote the set of all possible assignments $\mathbf{y} \in [8]^6$ such that for every $e_\ell = ij$, $y_i \in A_{x_\ell}$ and $y_j \in B_{x_\ell}$. Now we have

$$Z_{\mathbf{A}}(G) = \sum_{\mathbf{x} \in [4]^6} \sum_{\mathbf{y} \in Y[\mathbf{x}]} \text{wt}(\mathbf{y}), \quad \text{where } \text{wt}(\mathbf{y}) = \prod_{ij \in E} A_{y_i, y_j}.$$

Second, we further simplify the sum above by noticing that if $x_2 \neq x_3$ in $\mathbf{x}$, then $Y[\mathbf{x}]$ must be empty because the two edges e_2 and e_3 share the same head in G . In general, we only need to sum over the case when $x_1 = x_2 = x_3$ and $x_4 = x_5$, since otherwise the set $Y[\mathbf{x}]$ is empty. As a result,

$$Z_{\mathbf{A}}(G) = \sum_{\substack{x_1=x_2=x_3 \\ x_4=x_5 \\ x_6}} \sum_{\mathbf{y} \in Y[\mathbf{x}]} \text{wt}(\mathbf{y}).$$

The advantage of introducing x_ℓ , $\ell \in [6]$, is that, once $\mathbf{x}$ is fixed, one can always decompose A_{y_i, y_j} as a product $\alpha_{y_i} \cdot \beta_{y_j}$, for all $\mathbf{y} \in Y[\mathbf{x}]$ and all $ij \in E$, since $\mathbf{y}$ belonging to $Y[\mathbf{x}]$ guarantees that (y_i, y_j) falls inside one of the four blocks of $\mathbf{A}$. This allows us to greatly simplify $\text{wt}(\mathbf{y})$: If $\mathbf{y} \in Y[\mathbf{x}]$, then

$$\text{wt}(\mathbf{y}) = A_{y_1, y_3} \cdot A_{y_1, y_2} \cdot A_{y_2, y_3} \cdot A_{y_3, y_4} \cdot A_{y_3, y_5} \cdot A_{y_5, y_6} = \alpha_{y_1} \beta_{y_3} \alpha_{y_1} \beta_{y_2} \alpha_{y_2} \beta_{y_3} \alpha_{y_3} \beta_{y_4} \alpha_{y_3} \beta_{y_5} \alpha_{y_5} \beta_{y_6}.$$

Also notice that $Y[\mathbf{x}]$, for any $\mathbf{x}$, is a direct product of subsets of $[8]$: $\mathbf{y} \in Y[\mathbf{x}]$ if and only if

$$\begin{aligned} y_1 \in L_1 = A_{x_1}, \quad y_2 \in L_2 = A_{x_3} \cap B_{x_1} = A_{x_1} \cap B_{x_1}, \quad y_3 \in L_3 = A_{x_4} \cap A_{x_5} \cap B_{x_2} \cap B_{x_3} = A_{x_4} \cap B_{x_1} \\ y_4 \in L_4 = B_{x_4}, \quad y_5 \in L_5 = A_{x_6} \cap B_{x_4}, \quad y_6 \in L_6 = B_{x_6}. \end{aligned}$$

As a result, $Z_{\mathbf{A}}(G)$ becomes

$$Z_{\mathbf{A}}(G) = \sum_{x_1, x_4, x_6} \sum_{y_i \in L_i, i \in [6]} ((\alpha_{y_1})^2 \alpha_{y_2} \beta_{y_2}) \cdot ((\alpha_{y_3})^2 (\beta_{y_3})^2) \cdot \beta_{y_4} \cdot (\alpha_{y_5} \beta_{y_5}) \cdot \beta_{y_6}. \quad (1)$$

Finally we construct the following *labeled* directed graph $\mathcal{G}_1$ over domain $[4]$. There are three vertices a, b and c , which correspond to x_1, x_4 and x_6 , respectively; and there are two directed edges ab and bc . We construct the weights as follows. The vertex weight vector of a is

$$w_\ell^{[a]} = \sum_{y_1 \in A_\ell, y_2 \in A_\ell \cap B_\ell} (\alpha_{y_1})^2 \cdot (\alpha_{y_2} \beta_{y_2}), \quad \text{for every } \ell \in [4];$$

the vertex weights of b and c are the same:

$$w_\ell^{[b]} = w_\ell^{[c]} = \sum_{y \in B_\ell} \beta_y, \quad \text{for every } \ell \in [4].$$

The edge weight matrix $\mathbf{C}^{[ab]}$ of ab is

$$C_{k,\ell}^{[ab]} = \sum_{y_3 \in B_k \cap A_\ell} (\alpha_{y_3})^2 (\beta_{y_3})^2, \quad \text{for all } k, \ell \in [4];$$

and the edge weight matrix $\mathbf{C}^{[bc]}$ of bc is

$$C_{k,\ell}^{[bc]} = \sum_{y_5 \in B_k \cap A_\ell} \alpha_{y_5} \beta_{y_5}, \quad \text{for all } k, \ell \in [4].$$

Using (1) and the definition of $Z(\mathcal{G}_1)$, one can verify that $Z_{\mathbf{A}}(G) = Z(\mathcal{G}_1)$ and thus, we have reduced the domain size from 8 (which is the number of rows and columns in $\mathbf{A}$), to 4 (which is the number of blocks in $\mathbf{A}$). However, we also seem to have paid a high price. Two issues are worth pointing out here:

1. Unlike in $Z_{\mathbf{A}}(G)$, different edges in $\mathcal{G}_1$ have *different* edge weight matrices in general. For example, the matrices associated with ab and bc are clearly different, for general α and β . Actually, the set of matrices that may appear as an edge weight of $\mathcal{G}_1$, constructed from *all possible* directed graphs G after one round of domain reduction, is *infinite* in general.
2. Unlike in $Z_{\mathbf{A}}(G)$, we have to introduce vertex weights in $\mathcal{G}_1$. Similarly, vertices may have different vertex weight vectors, and the set of vectors that may appear as a vertex weight of $\mathcal{G}_1$, constructed from *all possible* G after one round of domain reduction, is *infinite* in general.

It is also worth noticing that $\{0, 1\}$ -matrices are not that special under this framework. Even if the $\mathbf{A}$ we start with is $\{0, 1\}$, the edge and vertex weights of $\mathcal{G}_1$ immediately become general non-negative integers right after the first round of domain reduction, and we have to deal with integer weights afterwards.

These two issues pose a difficulty because we need to carry out the domain reduction process several times, until the computation becomes trivial. However, the reduction process described above crucially used the assumption that $\mathbf{A}$ is block-rank-1 (otherwise one cannot replace $A_{i,j}$ with $\alpha_i \cdot \beta_j$). Thus, there is no way to continue this process if some edge weight matrix in $\mathcal{G}_1$ is *not* block-rank-1. To deal with this case, we show that if this happens for some G , then $Z_{\mathbf{A}}(\cdot)$ is $\#P$ -hard. Informally, we have

Theorem 1 (Informal). *For any G , if one of the edge matrices in $\mathcal{G}_k$ (constructed from G after k rounds of domain reductions), for some $k \geq 1$, is not block-rank-1, then $Z_{\mathbf{A}}(\cdot)$ is $\#P$ -hard.*

The proof of Theorem 1 for $k = 1$ follows directly from Bulatov and Grohe [4]. However, due to the two issues discussed earlier, edge weights and vertex weights of $\mathcal{G}_1$ are drawn from infinite sets in general and thus, even proving it for $k = 2$ is highly non-trivial.

After obtaining Theorem 1, which essentially gives us a dichotomy theorem for non-negative matrices, it remains unclear whether the dichotomy is *decidable*. The difficulty is that, to decide whether $Z_{\mathbf{A}}(\cdot)$ is

in P or #P-hard, one needs to check infinitely many matrices (all the edge weight matrices that appear in the domain reduction process, from *all possible* directed graphs G) and to see whether all of them are block-rank-1. To do this, we give an algebraic proof using properties of polynomials. We manage to show that it is not necessary to check these matrices one by one, but only need to check whether or not the entries of $\mathbf{A}$ satisfy finitely many polynomial constraints.

1.2 Proof Sketch

We assume below that $\mathbf{A}$ is non-negative and block-rank-1 since the case when $\mathbf{A}$ is not block-rank-1 has already been dealt with by Bulatov and Grohe [4]. To show that $Z_{\mathbf{A}}(\cdot)$ is either in P or #P-hard, we use the following two steps.

In the first step, we *define* from $\mathbf{A}$ a finite sequence of pairs:

$$(\mathfrak{X}_0, \mathfrak{Y}_0), (\mathfrak{X}_1, \mathfrak{Y}_1), \dots, (\mathfrak{X}_h, \mathfrak{Y}_h), \quad \text{for some } h : 0 \leq h < m,$$

where $\mathfrak{X}_0 = \{\mathbf{1}\}$, $\mathfrak{Y}_0 = \{\mathbf{A}\}$ and $\mathbf{1}$ denotes the m -dimensional all-1 vector. Each pair $(\mathfrak{X}_k, \mathfrak{Y}_k)$, $k \in [h]$, is defined from $(\mathfrak{X}_{k-1}, \mathfrak{Y}_{k-1})$. Roughly speaking, $\mathfrak{Y}_k$ (resp. $\mathfrak{X}_k$) is the set of all edge matrices (resp. vertex vectors) that may appear in $\mathcal{G}_k$, after k rounds of domain reductions. There also exist positive integers

$$m = m_0 > m_1 > \dots > m_h \geq 1$$

such that every $\mathfrak{Y}_k$, $k \in [h]$, is a set of $m_k \times m_k$ non-negative matrices; and every $\mathfrak{X}_k$, $k \in [h]$, is a set of m_k -dimensional non-negative vectors. Although the sets $\mathfrak{X}_k$ and $\mathfrak{Y}_k$ are *infinite* in general (which is the reason why we used the word “*define*” instead of “*construct*”), the definition of $(\mathfrak{X}_k, \mathfrak{Y}_k)$ guarantees the following two properties:

1. For each $k \in [h]$, matrices in $\mathfrak{Y}_k$ share the same *support structure*:

$$\text{for all } \mathbf{B}, \mathbf{B}' \in \mathfrak{Y}_k, \quad B_{i,j} > 0 \iff B'_{i,j} > 0;$$

2. Every matrix $\mathbf{B}$ in $\mathfrak{Y}_h$ is a *permutation* matrix.

The definition of $(\mathfrak{X}_k, \mathfrak{Y}_k)$ from $(\mathfrak{X}_{k-1}, \mathfrak{Y}_{k-1})$ can be found in Section 4. In Section 7 we prove that, for all $k \in [h]$ and $\mathbf{B} \in \mathfrak{Y}_k$, the problem of computing $Z_{\mathbf{B}}(\cdot)$ is polynomial-time reducible to that of $Z_{\mathbf{A}}(\cdot)$. From this we obtain the hardness part of our dichotomy theorem using Bulatov and Grohe [4]: If there exists a matrix $\mathbf{B} \in \mathfrak{Y}_k$ for some $k \in [h]$ such that $\mathbf{B}$ is not block-rank-1, then $Z_{\mathbf{A}}(\cdot)$ is #P-hard.

Now we assume that all matrices in $\mathfrak{Y}_k$, $k \in [h]$, are block-rank-1. To finish the proof we only need to show that if this is true, then $Z_{\mathbf{A}}(\cdot)$ is indeed in P. To this end, we use the *domain reduction* process to construct from the input graph G a sequence of *labeled* directed graphs $\mathcal{G}_1, \dots, \mathcal{G}_h$ such that

1. $Z(\mathcal{G}_1) = Z_{\mathbf{A}}(G)$ and $Z(\mathcal{G}_{k+1}) = Z(\mathcal{G}_k)$ for all $k : 1 \leq k < h$; and
2. For every $k \in [h]$, we have $\mathbf{A}^{[e]} \in \mathfrak{Y}_k$ for all edges e in $\mathcal{G}_k$ and $\mathbf{w}^{[v]} \in \mathfrak{X}_k$ for all vertices v in $\mathcal{G}_k$.

This sequence can be constructed in polynomial time, because the construction of $\mathcal{G}_{k+1}$ from $\mathcal{G}_k$ can be done very efficiently as described in Section 1.1, and also because the number of graphs in the sequence is at most m . By the two properties above, we have $Z_{\mathbf{A}}(G) = Z(\mathcal{G}_h)$; and every edge weight matrix $\mathbf{A}^{[e]}$ in $\mathcal{G}_h$ is a *permutation* matrix. As a result, we can compute $Z_{\mathbf{A}}(G)$ in polynomial time since $Z(\mathcal{G}_h)$ can be computed efficiently.

This finishes the proof of our dichotomy theorem: Given any non-negative matrix $\mathbf{A}$, the problem of computing $Z_{\mathbf{A}}(\cdot)$ is either in polynomial time or #P-hard. Moreover, to decide which case it is, we only need to check whether the matrices in $\mathfrak{Y}_k$, $k \in [h]$, satisfy the following condition:

The Block-Rank-1 Condition: Every matrix $\mathbf{B} \in \mathfrak{Y}_k$, $k \in [h]$, is block-rank-1.

However, as mentioned earlier, each of the sets $\mathfrak{Y}_k$, $k \in [h]$, is infinite in general, so one cannot check the matrices one by one. Instead, we express the block-rank-1 condition as a finite collection of polynomial constraints over $\mathfrak{Y}_k$. The way $(\mathfrak{X}_k, \mathfrak{Y}_k)$ is defined from $(\mathfrak{X}_{k-1}, \mathfrak{Y}_{k-1})$ allows us to prove that, to check whether every matrix in $\mathfrak{Y}_k$ (or every vector in $\mathfrak{X}_k$) satisfies a certain polynomial constraint, one only needs to check a finitely many polynomial constraints for $(\mathfrak{X}_{k-1}, \mathfrak{Y}_{k-1})$. Therefore, to check whether $\mathfrak{Y}_k$, $k \in [h]$, satisfies the block-rank-1 condition we only need to check a finitely many polynomial constraints for $(\mathfrak{X}_0, \mathfrak{Y}_0)$. Since $\mathfrak{X}_0 = \{\mathbf{1}\}$ and $\mathfrak{Y}_0 = \{\mathbf{A}\}$ are both finite, this can be done in a finite number of steps.

2 Preliminaries

We write $\mathbb{R}_+$ to denote the set of non-negative *algebraic* numbers. Throughout the rest of the paper, we deal with non-negative algebraic numbers (or vectors/matrices with non-negative algebraic entries) only and will refer to them simply as non-negative (real) numbers for convenience. We can also work with any reasonable model of computation for algebraic numbers, e.g., the one used in [22, 23, 9]. This issue does not seem central to this paper because when the complexity of $Z_{\mathbf{A}}(\cdot)$ is concerned, the matrix $\mathbf{A}$ is fixed and its entries are considered as constants. The input size only depends on the size of the input graph.

We say $\mathcal{G} = (G, \mathcal{V}, \mathcal{E})$ is a *labeled directed graph* over $[m] = \{1, \dots, m\}$ for some positive integer m , if

1. $G = (V, E)$ is a directed graph (which may have parallel edges but no self-loops);
2. Every vertex $v \in V$ is labeled with an m -dimensional non-negative vector $\mathcal{V}(v) \in \mathbb{R}_+^m$ as its vertex weight; and
3. Every edge $uv \in E$ is labeled with an $m \times m$ (not necessarily symmetric) non-negative matrix $\mathcal{E}(uv) \in \mathbb{R}_+^{m \times m}$ as its edge weight.

Let $\mathcal{G} = (G, \mathcal{V}, \mathcal{E})$ be a labeled directed graph, where $G = (V, E)$. For each $v \in V$, we use $\mathbf{w}^{[v]} = \mathcal{V}(v)$ to denote its vertex weight vector; and for each $uv \in E$, we use $\mathbf{C}^{[uv]} = \mathcal{E}(uv)$ to denote its edge weight matrix. Then we define $Z(\mathcal{G})$ as follows:

$$Z(\mathcal{G}) = \sum_{\xi: V \rightarrow [m]} \text{wt}(\mathcal{G}, \xi), \quad \text{where } \text{wt}(\mathcal{G}, \xi) = \prod_{v \in V} w_{\xi(v)}^{[v]} \prod_{uv \in E} C_{\xi(u), \xi(v)}^{[uv]}$$

denotes the *weight* of the assignment ξ .

Let $\mathbf{C}$ be an $m \times m$ non-negative matrix. We are interested in the complexity of $Z_{\mathbf{C}}(\cdot)$:

$$Z_{\mathbf{C}}(G) = Z(\mathcal{G}), \quad \text{for any directed graph } G = (V, E),$$

where $\mathcal{G} = (G, \mathcal{V}, \mathcal{E})$ is the labeled directed graph with $\mathcal{V}(v) = \mathbf{1} \in \mathbb{R}_+^m$ for all $v \in V$ and $\mathcal{E}(uv) = \mathbf{C}$ for all edges $uv \in E$.

Definition 1 (Pattern and block pattern). *We say $\mathcal{P}$ is an $m \times m$ pattern if $\mathcal{P} \subseteq [m] \times [m]$. $\mathcal{P}$ is said to be trivial if $\mathcal{P} = \emptyset$. A non-negative $m \times m$ matrix $\mathbf{C}$ is of pattern $\mathcal{P}$, if for all $i, j \in [m]$, we have $C_{i,j} > 0$ if and only if $(i, j) \in \mathcal{P}$. $\mathbf{C}$ is also called a $\mathcal{P}$ -matrix. We say $\mathcal{T}$ is an $m \times m$ block pattern if*

1. $\mathcal{T} = \{(A_1, B_1), \dots, (A_r, B_r)\}$ for some $r \geq 0$;
2. $A_i \subseteq [m]$, $A_i \neq \emptyset$, $B_i \subseteq [m]$ and $B_i \neq \emptyset$ for all $i \in [r]$; and
3. $A_i \cap A_j = B_i \cap B_j = \emptyset$, for all $i \neq j \in [r]$.

$\mathcal{T}$ is said to be trivial if $\mathcal{T} = \emptyset$. A block pattern $\mathcal{T}$ naturally defines a pattern $\mathcal{P}$, where

$$\mathcal{P} = \{(i, j) \mid \exists k \in [r] \text{ such that } i \in A_k \text{ and } j \in B_k\}.$$

We also say $\mathcal{P}$ is consistent with $\mathcal{T}$. Finally, we say a non-negative $m \times m$ matrix $\mathbf{C}$ is of block pattern $\mathcal{T}$, if $\mathbf{C}$ is of pattern $\mathcal{P}$ defined by $\mathcal{T}$. $\mathbf{C}$ is also called a $\mathcal{T}$ -matrix.

Definition 2. We say an $m \times m$ non-negative matrix $\mathbf{C}$ is block-rank-1 if

1. Either $\mathbf{C} = \mathbf{0}$ is the zero matrix (and is of block pattern $\mathcal{T} = \emptyset$); or
2. $\mathbf{C}$ is of block pattern $\mathcal{T}$, for some $m \times m$ block pattern $\mathcal{T} = \{(A_1, B_1), \dots, (A_r, B_r)\}$ with $r \geq 1$; and for every $k \in [r]$, the sub-matrix of $\mathbf{C}$ indexed by A_k and B_k is (exactly) rank 1.

Let $\mathbf{C}$ be a non-negative block-rank-1 matrix of block pattern $\mathcal{T}$. Then there exists a unique pair (α, β) of non-negative m -dimensional vectors such that

1. For every $i \in [m]$, $\alpha_i > 0 \iff i \in \bigcup_{k \in [r]} A_k$; and $\beta_i > 0 \iff i \in \bigcup_{k \in [r]} B_k$;
2. $C_{i,j} = \alpha_i \cdot \beta_j$ for all $i, j \in [m]$ such that $C_{i,j} > 0$; and
3. If $r \geq 1$, then $\sum_{j \in A_i} \alpha_j = 1$ for every $i \in [r]$.

The pair (α, β) is called the (vector) representation of $\mathbf{C}$. Note that we have $\alpha = \beta = \mathbf{0}$ when $\mathbf{C} = \mathbf{0}$.

It is clear that $\mathcal{T}$ and (α, β) together uniquely determine a non-negative block-rank-1 matrix.

The following theorem concerning the complexity of $Z_{\mathbf{C}}(\cdot)$ is proved by Bulatov and Grohe [4] (also see [19]).

Theorem 2 (Bulatov and Grohe [4]). Let $\mathbf{C}$ be a non-negative $m \times m$ matrix. If $\mathbf{C}$ is not block-rank-1, then the problem of computing $Z_{\mathbf{C}}(\cdot)$ is $\#P$ -hard.

Let $\mathcal{T}$ be an $m \times m$ non-trivial block pattern where $\mathcal{T} = \{(A_1, B_1), \dots, (A_r, B_r)\}$ for some $r \geq 1$. It defines the following $r \times r$ pattern $\mathcal{P} = \text{gen}(\mathcal{T})$: For all $i, j \in [r]$, $(i, j) \in \mathcal{P}$ if and only if $B_i \cap A_j \neq \emptyset$. (To avoid ambiguity, we always assume that A_j 's in $\mathcal{T}$ are indexed using a canonical ordering when deriving $\mathcal{P} = \text{gen}(\mathcal{T})$ from $\mathcal{T}$, e.g., sorted by the smallest elements in A_j 's.) Note that $\mathcal{P} = \text{gen}(\mathcal{T})$ can be trivial, i.e., $\mathcal{P} = \emptyset$, even if $\mathcal{T}$ is non-trivial.

Next, we introduce a generalized version of $Z_{\mathbf{C}}(\cdot)$. Let $m \geq 1$ and $(\mathfrak{P}, \Omega)$ be a pair in which

1. $\mathfrak{P}$ is a finite and nonempty set of non-negative m -dimensional vectors with $\mathbf{1} \in \mathfrak{P}$; and
2. Ω is a finite and nonempty set of $m \times m$ non-negative matrices.

We then use $Z(\cdot)$ to define the function $Z_{\mathfrak{P}, \Omega}(\cdot)$ as follows:

$$Z_{\mathfrak{P}, \Omega}(\mathcal{G}) = Z(\mathcal{G}),$$

where $\mathcal{G} = (G, \mathcal{V}, \mathcal{E})$ is a labeled directed graph with $\mathcal{V}(v) \in \mathfrak{P}$ for any vertex $v \in V(G)$; and $\mathcal{E}(uv) \in \Omega$ for any edge $uv \in E(G)$. Note that $Z_{\mathfrak{P}, \Omega}(\cdot)$ captures exactly $\#CSPs$ with non-negative constraint functions of arity at most two. As a special case, $Z_{\mathbf{C}}(\cdot)$ is exactly $Z_{\mathfrak{P}, \Omega}(\cdot)$ with $\mathfrak{P} = \{\mathbf{1}\}$ and $\Omega = \{\mathbf{C}\}$.

Finally, let $m \geq 1$ and $(\mathfrak{X}, \mathfrak{Y})$ and $(\mathfrak{X}', \mathfrak{Y}')$ be two pairs such that:

1. $\mathfrak{X}$ and $\mathfrak{X}'$ are two nonempty (and possibly infinite) sets of non-negative m -dimensional vectors with $\mathbf{1} \in \mathfrak{X}$ and $\mathbf{1} \in \mathfrak{X}'$; and
2. $\mathfrak{Y}$ and $\mathfrak{Y}'$ are two nonempty (and possibly infinite) sets of non-negative $m \times m$ matrices.

Definition 3 (Reduction). We say $(\mathfrak{X}', \mathfrak{Y}')$ is polynomial-time reducible to $(\mathfrak{X}, \mathfrak{Y})$ if for every finite and nonempty subset $\mathfrak{P}' \subseteq \mathfrak{X}'$ with $\mathbf{1} \in \mathfrak{P}'$ and every finite and nonempty subset $\Omega' \subseteq \mathfrak{Y}'$, there exist a finite and nonempty subset $\mathfrak{P} \subseteq \mathfrak{X}$ with $\mathbf{1} \in \mathfrak{P}$ and a finite and nonempty subset $\Omega \subseteq \mathfrak{Y}$, such that $Z_{\mathfrak{P}', \Omega'}(\cdot)$ is polynomial-time reducible to $Z_{\mathfrak{P}, \Omega}(\cdot)$.

3 Main Theorems

We prove a complexity dichotomy theorem for all counting problems $Z_{\mathbf{C}}(\cdot)$ where $\mathbf{C}$ is any non-negative matrix. Actually, our main theorem is more general and applies to $Z_{\mathfrak{P},\Omega}(\cdot)$ over all *finite* $\mathcal{P}$ -pairs $(\mathfrak{P}, \Omega)$ as defined below.

Definition 4. Let $\mathcal{P}$ be an $m \times m$ pattern. An m -dimensional non-negative vector $\mathbf{w}$ is said to be

- positive: $w_i > 0$ for all $i \in [m]$; and
- $\mathcal{P}$ -weakly positive: for all $i \in [m]$, $w_i > 0$ if and only if $(i, i) \in \mathcal{P}$.

We call $(\mathfrak{X}, \mathfrak{Y})$ a $\mathcal{P}$ -pair if

1. $\mathfrak{X}$ is a nonempty (and possibly infinite) set of positive or $\mathcal{P}$ -weakly positive vectors with $\mathbf{1} \in \mathfrak{X}$;
2. $\mathfrak{Y}$ is a nonempty (and possibly infinite) set of $m \times m$ (non-negative) $\mathcal{P}$ -matrices.

We say it is a *finite* $\mathcal{P}$ -pair if both sets are finite. We normally use $(\mathfrak{P}, \Omega)$ to denote a finite $\mathcal{P}$ -pair.

Similarly, for any $m \times m$ block pattern $\mathcal{T}$, we can define $\mathcal{T}$ -weakly positive vectors as well as $\mathcal{T}$ -pairs by replacing the $\mathcal{P}$ above with the pattern defined by $\mathcal{T}$.

We prove the following complexity dichotomy theorem:

Theorem 3 (Complexity Dichotomy). Let $\mathcal{P}$ be an $m \times m$ pattern for some $m \geq 1$. Then for any finite $\mathcal{P}$ -pair $(\mathfrak{P}, \Omega)$, the problem of computing $Z_{\mathfrak{P},\Omega}(\cdot)$ is either in polynomial time or $\#P$ -hard.

It gives us a dichotomy for the special case of $Z_{\mathbf{C}}(\cdot)$ when $\mathfrak{P} = \{\mathbf{1}\}$ and $\Omega = \{\mathbf{C}\}$. Moreover, we show that for the special case when $\mathfrak{P} = \{\mathbf{1}\}$, we can decide in a finite number of steps whether $Z_{\mathfrak{P},\Omega}$ is in polynomial time or $\#P$ -hard. In particular, it implies that the dichotomy for $Z_{\mathbf{C}}(\cdot)$ is decidable.

Theorem 4 (Decidability). Given any positive integer $m \geq 1$, an $m \times m$ pattern $\mathcal{P}$, and a finite $\mathcal{P}$ -pair $(\mathfrak{P}, \Omega)$ with $\mathfrak{P} = \{\mathbf{1}\}$, the problem of whether $Z_{\mathfrak{P},\Omega}(\cdot)$ is in polynomial time or $\#P$ -hard is decidable.

Remark 1. Compared to the dichotomy theorem for $\#CSP$ with non-negative weights [8], the classification of Theorems 3 and 4 are limited since they only apply to $\#CSP$ with constraint functions that have arity at most two and satisfy certain conditions.

We prove Theorem 3 and 4 in the rest of the section. The lemmas (Lemmas 1, 2, and 3) used in the proof will be proved in the rest of the paper.

3.1 Defining New Pairs: $\text{gen-pair}(\mathfrak{X}, \mathfrak{Y})$

We first state a key lemma which will be proved in Sections 4 and 7. We need the following definition.

Definition 5. A set S of non-negative m -dimensional vectors, for some $m \geq 1$, is closed if $\mathbf{w}_1 \circ \mathbf{w}_2 \in S$ for all vectors $\mathbf{w}_1, \mathbf{w}_2 \in S$, where we use $\circ$ to denote the Hadamard product of two vectors: $\mathbf{w}_1 \circ \mathbf{w}_2$ is the m -dimensional vector whose i th entry is $w_{1,i} \cdot w_{2,i}$ for all $i \in [m]$.

Let $(\mathfrak{X}, \mathfrak{Y})$ be a (possibly infinite) $\mathcal{T}$ -pair for some non-trivial $m \times m$ block pattern $\mathcal{T}$. We also assume that every matrix in $\mathfrak{Y}$ is block-rank-1. Let $\mathcal{P}' = \text{gen}(\mathcal{T})$. Then in Section 4, we introduce an operation gen-pair over $(\mathfrak{X}, \mathfrak{Y})$, which defines a new (and possibly infinite) $\mathcal{P}'$ -pair $(\mathfrak{X}', \mathfrak{Y}') = \text{gen-pair}(\mathfrak{X}, \mathfrak{Y})$ in which $\mathfrak{X}'$ is closed. In Section 7, we further show that $(\mathfrak{X}', \mathfrak{Y}')$ is *polynomial-time reducible* to $(\mathfrak{X}, \mathfrak{Y})$.

We summarize properties of gen-pair needed in the following lemma:

Lemma 1. Let $(\mathfrak{X}, \mathfrak{Y})$ be a $\mathcal{T}$ -pair for some non-trivial block pattern $\mathcal{T}$. Suppose that every matrix in $\mathfrak{Y}$ is block-rank-1, then $(\mathfrak{X}', \mathfrak{Y}') = \text{gen-pair}(\mathfrak{X}, \mathfrak{Y})$ (as defined in Section 4) is a $\mathcal{P}'$ -pair, where $\mathcal{P}' = \text{gen}(\mathcal{T})$. Moreover, the new vector set $\mathfrak{X}'$ is closed and $(\mathfrak{X}', \mathfrak{Y}')$ is polynomial-time reducible to $(\mathfrak{X}, \mathfrak{Y})$.

3.2 Proof of Theorem 3

Assuming Lemma 1, we are now ready to prove Theorem 3.

Let $(\mathfrak{P}, \Omega)$ be a *finite* $\mathcal{P}$ -pair for some $m \times m$ pattern $\mathcal{P}$. We may assume that there is a block pattern $\mathcal{T}$ that is consistent with $\mathcal{P}$ and all matrices in Ω are block-rank-1; otherwise it follows from Theorem 2 that $Z_{\mathfrak{P}, \Omega}(\cdot)$ is $\#P$ -hard and we are done with it. We summarize this as the following property:

R₀: $(\mathfrak{P}, \Omega)$ is a finite $\mathcal{T}$ -pair for some $m \times m$ block pattern $\mathcal{T}$; and
Every matrix in Ω is block-rank-1.

For convenience, we rename $(\mathfrak{P}, \Omega)$ to be $(\mathfrak{X}_0, \mathfrak{Y}_0)$ and rename m and $\mathcal{T}$ to be m_0 and $\mathcal{T}_0$, respectively. Next we define a finite sequence of pairs using the **gen-pair** operation, starting with $(\mathfrak{X}_0, \mathfrak{Y}_0)$.

First, if $|A_i| = |B_i| = 1$ for all i , i.e., every set A_i and B_i in $\mathcal{T}_0$ is a singleton, then the sequence has only one pair $(\mathfrak{X}_0, \mathfrak{Y}_0)$, and the definition of this sequence is complete. Note that this includes the special case when $\mathcal{T}_0 = \emptyset$ and $\mathfrak{Y}_0 = \{\mathbf{0}\}$, where $\mathbf{0}$ denotes the all-0 matrix of dimension m_0 .

Otherwise, in Step 1, we define a new $\mathcal{P}_1$ -pair $(\mathfrak{X}_1, \mathfrak{Y}_1)$ using **gen-pair**:

$$(\mathfrak{X}_1, \mathfrak{Y}_1) = \text{gen-pair}(\mathfrak{X}_0, \mathfrak{Y}_0), \quad \text{where } \mathcal{P}_1 = \text{gen}(\mathcal{T}_0).$$

By Lemma 1, $(\mathfrak{X}_1, \mathfrak{Y}_1)$ is polynomial-time reducible to $(\mathfrak{X}_0, \mathfrak{Y}_0)$ (recall Definition 3). This leads to one of the following two cases: (1) either we have that $\mathcal{P}_1$ is consistent with a block pattern, denoted by $\mathcal{T}_1$ (hence $(\mathfrak{X}_1, \mathfrak{Y}_1)$ is also a $\mathcal{T}_1$ -pair), and every matrix in $\mathfrak{Y}_1$ is block-rank-1, or (2) it follows from Theorem 2 and the polynomial-time reduction that $Z_{\mathfrak{P}, \Omega}(\cdot)$ is $\#P$ -hard and we are done with this case. (To see the latter, assuming that $\mathbf{D} \in \mathfrak{Y}_1$ is not block-rank-1, it follows from Theorem 2 that $Z_{\mathfrak{P}_1, \Omega_1}(\cdot)$ is $\#P$ -hard where we let $\mathfrak{P}_1 = \{\mathbf{1}\}$ and $\Omega_1 = \{\mathbf{D}\}$. It follows from Lemma 1 (and the fact that $\mathbf{1} \in \mathfrak{X}_1$) that there exists a *finite*⁷ pair $(\mathfrak{P}_0, \Omega_0)$ with $\mathfrak{P}_0 \subseteq \mathfrak{X}_0$ and $\Omega_0 \subseteq \mathfrak{Y}_0$ such that $Z_{\mathfrak{P}_1, \Omega_1}(\cdot)$ is polynomial-time reducible to $Z_{\mathfrak{P}_0, \Omega_0}(\cdot)$ which is trivially reducible to $Z_{\mathfrak{P}, \Omega}(\cdot)$ since $\mathfrak{P}_0 \subseteq \mathfrak{X}_0 = \mathfrak{P}$ and $\Omega_0 \subseteq \mathfrak{Y}_0 = \Omega$.)

As a result, we assume below that $\mathcal{T}_1$ and $(\mathfrak{X}_1, \mathfrak{Y}_1)$ satisfy the following property:

R*: $\mathcal{T}_1$ is an $m_1 \times m_1$ block pattern that is consistent with $\mathcal{P}_1 = \text{gen}(\mathcal{T}_0)$, where m_1 is the number of pairs in $\mathcal{T}_0$; $(\mathfrak{X}_1, \mathfrak{Y}_1) = \text{gen-pair}(\mathfrak{X}_0, \mathfrak{Y}_0)$ is a $\mathcal{T}_1$ -pair, and every matrix in $\mathfrak{Y}_1$ is block-rank-1.

We also have $m_0 > m_1$ since at least one of the sets in $\mathcal{T}_0$ is not a singleton.

We remark that both sets $\mathfrak{X}_1$ and $\mathfrak{Y}_1$ are infinite in general, so one cannot check the matrices in $\mathfrak{Y}_1$ for the block-rank-1 property one by one. It does not matter right now because we are only proving the dichotomy theorem. However, it will become a serious problem later when we show that the dichotomy is decidable. We have to show that the block-rank-1 property can be verified in a finite number of steps.

We repeat the process above. After $\ell \geq 1$ steps, either we are already done with $(\mathfrak{P}, \Omega)$ by showing that $Z_{\mathfrak{P}, \Omega}(\cdot)$ is $\#P$ -hard, or we have defined a sequence of $\ell + 1$ pairs:

$$(\mathfrak{X}_0, \mathfrak{Y}_0), (\mathfrak{X}_1, \mathfrak{Y}_1), \dots, (\mathfrak{X}_\ell, \mathfrak{Y}_\ell),$$

and $\ell + 1$ block patterns $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_\ell$ that satisfy the following property:

R_ℓ: For every $i \in [0 : \ell]$, $\mathcal{T}_i$ is a block pattern;
For every $i \in [\ell]$, $\mathcal{T}_i$ is consistent with $\mathcal{P}_i = \text{gen}(\mathcal{T}_{i-1})$;
For every $i \in [\ell]$, $(\mathfrak{X}_i, \mathfrak{Y}_i) = \text{gen-pair}(\mathfrak{X}_{i-1}, \mathfrak{Y}_{i-1})$ is a $\mathcal{T}_i$ -pair; and
For every $i \in [0 : \ell]$, all the matrices in $\mathfrak{Y}_i$ are block-rank-1.

⁷Here this is trivial since $(\mathfrak{X}_0, \mathfrak{Y}_0)$ is itself a finite pair.

We have two cases. If every set in $\mathcal{T}_\ell$ is a singleton (including the case when $\mathcal{T}_\ell = \emptyset$ and $\mathfrak{Y}_\ell = \{\mathbf{0}\}$), then the sequence has only $\ell + 1$ pairs and the definition of the sequence is complete. Otherwise in Step $\ell + 1$ we apply **gen** and **gen-pair** again to define $\mathcal{P}_{\ell+1}$ and $(\mathfrak{X}_{\ell+1}, \mathfrak{Y}_{\ell+1})$ from $\mathcal{T}_\ell$ and $(\mathfrak{X}_\ell, \mathfrak{Y}_\ell)$. It follows from Theorem 2, Lemma 1, and a similar argument (note that the definition of reductions in Definition 3 is transitive so $(\mathfrak{X}_{\ell+1}, \mathfrak{Y}_{\ell+1})$ is polynomial-time reducible to $(\mathfrak{X}_0, \mathfrak{Y}_0)$) that either $\mathcal{P}_{\ell+1}$ is consistent with a block pattern, denoted by $\mathcal{T}_{\ell+1}$, and

$$(\mathfrak{X}_0, \mathfrak{Y}_0), (\mathfrak{X}_1, \mathfrak{Y}_1), \dots, (\mathfrak{X}_{\ell+1}, \mathfrak{Y}_{\ell+1}),$$

together with $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_{\ell+1}$ satisfy **(R)**, or we have that $Z_{\mathfrak{P}, \Omega}(\cdot)$ is $\#P$ -hard.

We conclude that either we are already done with $(\mathfrak{P}, \Omega)$ by showing that $Z_{\mathfrak{P}, \Omega}(\cdot)$ is $\#P$ -hard, or this process must end with a sequence of $h + 1$ pairs

$$(\mathfrak{X}_0, \mathfrak{Y}_0), (\mathfrak{X}_1, \mathfrak{Y}_1), \dots, (\mathfrak{X}_h, \mathfrak{Y}_h), \quad \text{for some } h \geq 0,$$

together with $h + 1$ positive integers $m_0 > \dots > m_h \geq 1$ and $h + 1$ block patterns $\mathcal{T}_0, \dots, \mathcal{T}_h$ such that

- R:** For every $i \in [0 : h]$, $\mathcal{T}_i$ is an $m_i \times m_i$ block pattern;
- For every $i \in [h]$, $\mathcal{T}_i$ is consistent with $\mathcal{P}_i = \mathbf{gen}(\mathcal{T}_{i-1})$;
- Either $\mathcal{T}_h = \emptyset$ is trivial or every set in $\mathcal{T}_h$ is a singleton;
- For every $i \in [h]$, $(\mathfrak{X}_i, \mathfrak{Y}_i) = \mathbf{gen-pair}(\mathfrak{X}_{i-1}, \mathfrak{Y}_{i-1})$ is a $\mathcal{T}_i$ -pair; and
- For every $i \in [0 : h]$, all the matrices in $\mathfrak{Y}_i$ are block-rank-1.

Because $m_0 > \dots > m_h \geq 1$, we also have $h < m_0 = m$.

To complete the proof of the dichotomy theorem, we show in Section 5 that

Lemma 2 (Tractability). *Given any block pattern $\mathcal{T}$ and a finite $\mathcal{T}$ -pair $(\mathfrak{P}, \Omega)$, let $(\mathfrak{X}_0, \mathfrak{Y}_0), \dots, (\mathfrak{X}_h, \mathfrak{Y}_h)$ be a sequence of pairs that satisfies condition **(R)** for some $h < m$, with $(\mathfrak{X}_0, \mathfrak{Y}_0) = (\mathfrak{P}, \Omega)$. Then we have that $Z_{\mathfrak{P}, \Omega}(\cdot)$ is computable in polynomial time.*

This finishes the proof of Theorem 3.

3.3 Proof of Theorem 4

Let $(\mathfrak{P}, \Omega)$ be a finite $\mathcal{P}$ -pair, where $\mathcal{P}$ is an $m \times m$ pattern. We now show that for the special case when $\mathfrak{X}_0 = \mathfrak{P} = \{\mathbf{1}\}$ the dichotomy (Theorem 3) is indeed decidable. For convenience we rename m and $(\mathfrak{P}, \Omega)$ to be m_0 and $(\mathfrak{X}_0, \mathfrak{Y}_0)$, and without loss of generality write $\mathcal{T}_0$ as the block pattern that is consistent with $\mathcal{P}$ (if no such $\mathcal{T}_0$ exists we know that $Z_{\mathfrak{P}, \Omega}(\cdot)$ is $\#P$ -hard). We describe our decidability algorithm below.

The algorithm first checks **(R)** on $\mathfrak{Y}_0$, i.e., every matrix in $\mathfrak{Y}_0$ is block-rank-1. This is easy since $\mathfrak{Y}_0$ is finite. It then computes from $\mathcal{T}_0$ a sequence of no more than m block patterns $\mathcal{T}_0, \dots, \mathcal{T}_h$ using **gen** repeatedly, and checks if they satisfy **(R)**. If **(R)** is violated (i.e., there is an i such that $\mathcal{P}_i = \mathbf{gen}(\mathcal{T}_{i-1})$ is not consistent with any block pattern) we know from the proof of Theorem 3 that $Z_{\mathfrak{P}, \Omega}(\cdot)$ is $\#P$ -hard and the algorithm terminates. If $h = 0$, i.e., either $\mathcal{T}_0 = \emptyset$ is trivial or every set in $\mathcal{T}_h$ is a singleton, $Z_{\mathfrak{P}, \Omega}(\cdot)$ is in polynomial time by Lemma 2 so the algorithm also terminates. Without loss of generality, we assume below that both $(\mathfrak{X}_0, \mathfrak{Y}_0)$ and $\mathcal{T}_0, \dots, \mathcal{T}_h$ satisfy condition **(R)**, and $h \geq 1$. The rest of the algorithm consists of $h < m$ steps.

At the beginning of the ℓ th step, $\ell \in [h]$, we have defined from $(\mathfrak{X}_0, \mathfrak{Y}_0)$ a sequence of ℓ pairs:

$$(\mathfrak{X}_0, \mathfrak{Y}_0), (\mathfrak{X}_1, \mathfrak{Y}_1), \dots, (\mathfrak{X}_{\ell-1}, \mathfrak{Y}_{\ell-1}), \quad \text{where } (\mathfrak{X}_{i+1}, \mathfrak{Y}_{i+1}) = \mathbf{gen-pair}(\mathfrak{X}_i, \mathfrak{Y}_i) \text{ is a } \mathcal{T}_{i+1}\text{-pair,}$$

and the algorithm has verified that they all satisfy condition **(R)**, i.e., every matrix in $\mathfrak{Y}_i$ with $i < \ell$ is block-rank-1; otherwise we know from the proof of Theorem 3 that $Z_{\mathfrak{P}, \Omega}(\cdot)$ is $\#P$ -hard and the algorithm

terminates. As a result, $(\mathfrak{X}_\ell, \mathfrak{Y}_\ell) = \text{gen-pair}(\mathfrak{X}_{\ell-1}, \mathfrak{Y}_{\ell-1})$ is a new and well-defined $\mathcal{T}_\ell$ -pair, and the goal of the ℓ th step of the algorithm is to check whether every matrix in $\mathfrak{Y}_\ell$ is block-rank-1. We refer to this property as the *rank property* for $\mathfrak{Y}_\ell$. We prove the following lemma in Section 8, showing that the rank property for $\mathfrak{Y}_\ell$ can be checked in a finite number of steps.

Lemma 3. *Let $(\mathfrak{X}_0, \mathfrak{Y}_0)$ be a finite $\mathcal{T}_0$ -pair with $\mathfrak{X}_0 = \{\mathbf{1}\}$. Assuming that $\mathcal{T}_0, \dots, \mathcal{T}_{\ell-1}, \mathcal{T}_\ell$ and $(\mathfrak{X}_0, \mathfrak{Y}_0), \dots, (\mathfrak{X}_{\ell-1}, \mathfrak{Y}_{\ell-1})$ all satisfy their conditions in **(R)** for some $\ell \in [h]$, the rank property for $\mathfrak{Y}_\ell$ can be checked in a finite number of steps.*

This finishes the description of our decidability algorithm, and Theorem 4 follows.

4 Definition of the gen-pair Operation

In this section, we define the operation **gen-pair**.

Let $\mathcal{T} = \{(A_1, B_1), \dots, (A_r, B_r)\}$ be a non-trivial $m \times m$ block pattern with $r \geq 1$. We use $\text{diag}(\mathcal{T})$ to denote the set of all $i \in [m]$ such that $i \in A_k$ and $i \in B_k$ for some $k \in [r]$. In this section, we always assume that $(\mathfrak{X}, \mathfrak{Y})$ is a $\mathcal{T}$ -pair such that every matrix in $\mathfrak{Y}$ is block-rank-1. This means that

1. All matrices in $\mathfrak{Y}$ are block-rank-1 and are of the same block pattern $\mathcal{T}$;
2. $\mathbf{1} \in \mathfrak{X}$ and every vector $\mathbf{w} \in \mathfrak{X}$ is either

positive: $w_i > 0$ for all $i \in [m]$; or

$\mathcal{T}$ -*weakly positive*: $w_i > 0$ if and only if $i \in \text{diag}(\mathcal{T})$.

Given such a pair $(\mathfrak{X}, \mathfrak{Y})$, **gen-pair** defines a new $\mathcal{P}$ -pair

$$(\mathfrak{X}', \mathfrak{Y}') = \text{gen-pair}(\mathfrak{X}, \mathfrak{Y}), \quad \text{where } \mathcal{P} = \text{gen}(\mathcal{T}).$$

To this end we first define a pair $(\mathfrak{X}^*, \mathfrak{Y}^*)$ from $(\mathfrak{X}, \mathfrak{Y})$, which is a *generalized $\mathcal{P}$ -pair* defined as follows.

Definition 6. *Let $\mathcal{P}$ be an $r \times r$ pattern with $r \geq 1$. An $r \times r$ non-negative matrix is called a $\mathcal{P}$ -diagonal matrix if it is a diagonal matrix and for all $i \in [r]$, its (i, i) th entry is positive if and only if $(i, i) \in \mathcal{P}$.*

We call $(\mathfrak{X}^, \mathfrak{Y}^*)$ a generalized $\mathcal{P}$ -pair if*

1. $\mathfrak{X}^*$ is a nonempty (and possibly infinite) set of positive or $\mathcal{P}$ -weakly positive vectors with $\mathbf{1} \in \mathfrak{X}^*$;
2. $\mathfrak{Y}^*$ is a nonempty (and possibly infinite) set of $\mathcal{P}$ -matrices or $\mathcal{P}$ -diagonal matrices.

For any block pattern $\mathcal{T}$, one can define $\mathcal{T}$ -diagonal matrices and generalized $\mathcal{T}$ -pairs similarly, by replacing the pattern $\mathcal{P}$ above with the one defined by $\mathcal{T}$.

We then use $(\mathfrak{X}^*, \mathfrak{Y}^*)$ to define $(\mathfrak{X}', \mathfrak{Y}')$. In this section we only show that $(\mathfrak{X}', \mathfrak{Y}')$ is a $\mathcal{P}$ -pair and $\mathfrak{X}'$ is *closed*. We will give the polynomial-time reduction from $(\mathfrak{X}', \mathfrak{Y}')$ to $(\mathfrak{X}, \mathfrak{Y})$ in Section 7.

4.1 Definition of $\mathfrak{Y}^*$

We define $\mathfrak{Y}^*$ which contains both $\mathcal{P}$ -matrices and $\mathcal{P}$ -diagonal matrices, where $\mathcal{P} = \text{gen}(\mathcal{T})$. As it becomes clear later in Section 4.3, $\mathcal{P}$ -diagonal matrices are really just $\mathcal{P}$ -weakly positive vectors in disguise, and will be absorbed into $\mathfrak{X}'$ when we define the final $\mathcal{P}$ -pair $(\mathfrak{X}', \mathfrak{Y}')$ in which $\mathfrak{Y}'$ contains $\mathcal{P}$ -matrices only.

There are two types of matrices in $\mathfrak{Y}^*$. First, an $r \times r$ matrix $\mathbf{D}$ is in $\mathfrak{Y}^*$ if there exist

1. a finite subset of matrices $\{\mathbf{C}^{[1]}, \dots, \mathbf{C}^{[g]}\} \subseteq \mathfrak{Y}$ with $g \geq 1$, and positive integers $s_1, \dots, s_g$;

2. a finite subset of matrices $\{\mathbf{D}^{[1]}, \dots, \mathbf{D}^{[h]}\} \subseteq \mathfrak{Y}$ with $h \geq 1$, and positive integers $t_1, \dots, t_h$;
3. a *positive* vector $\mathbf{w} \in \mathfrak{X}$,

such that: Let $(\boldsymbol{\alpha}^{[i]}, \boldsymbol{\beta}^{[i]})$ and $(\boldsymbol{\gamma}^{[i]}, \boldsymbol{\delta}^{[i]})$ be the representations of $\mathbf{C}^{[i]}$ and $\mathbf{D}^{[i]}$, respectively, then

$$D_{i,j} = \sum_{x \in B_i \cap A_j} \left(\beta_x^{[1]} \right)^{s_1} \cdots \left(\beta_x^{[g]} \right)^{s_g} \cdot \left(\gamma_x^{[1]} \right)^{t_1} \cdots \left(\gamma_x^{[h]} \right)^{t_h} \cdot w_x, \quad \text{for all } i, j \in [r].$$

The following lemma is easy to prove.

Lemma 4. *If $\mathbf{w} \in \mathfrak{X}$ is positive, then the matrix $\mathbf{D}$ defined above is a $\mathcal{P}$ -matrix, where $\mathcal{P} = \text{gen}(\mathcal{T})$.*

Proof. Because $(\mathfrak{X}, \mathfrak{Y})$ is a $\mathcal{T}$ -pair, all the matrices $\mathbf{C}^{[i]}$ and $\mathbf{D}^{[j]}$, $i \in [g]$ and $j \in [h]$, are $\mathcal{T}$ -matrices and thus, $\boldsymbol{\beta}^{[i]}$ is positive over $B_1 \cup \dots \cup B_r$ and $\boldsymbol{\gamma}^{[j]}$ is positive over $A_1 \cup \dots \cup A_r$. Since $\mathbf{w}$ is positive, we have that $D_{i,j} > 0$ if and only if $B_i \cap A_j \neq \emptyset$. $\square$

Second, an $r \times r$ matrix $\mathbf{D}$ is in $\mathfrak{Y}^*$ if there exist

1. a finite subset of matrices $\{\mathbf{C}^{[1]}, \dots, \mathbf{C}^{[g]}\} \subseteq \mathfrak{Y}$ with $g \geq 1$, and positive integers $s_1, \dots, s_g$;
2. a finite subset of matrices $\{\mathbf{D}^{[1]}, \dots, \mathbf{D}^{[h]}\} \subseteq \mathfrak{Y}$ with $h \geq 1$, and positive integers $t_1, \dots, t_h$;
3. a $\mathcal{T}$ -weakly positive vector $\mathbf{w} \in \mathfrak{X}$,

such that: Let $(\boldsymbol{\alpha}^{[i]}, \boldsymbol{\beta}^{[i]})$ and $(\boldsymbol{\gamma}^{[i]}, \boldsymbol{\delta}^{[i]})$ be the representations of $\mathbf{C}^{[i]}$ and $\mathbf{D}^{[i]}$, respectively, then

$$D_{i,j} = \sum_{x \in B_i \cap A_j} \left(\beta_x^{[1]} \right)^{s_1} \cdots \left(\beta_x^{[g]} \right)^{s_g} \cdot \left(\gamma_x^{[1]} \right)^{t_1} \cdots \left(\gamma_x^{[h]} \right)^{t_h} \cdot w_x, \quad \text{for all } i, j \in [r].$$

Similarly one can show that

Lemma 5. *If $\mathbf{w}$ is $\mathcal{T}$ -weakly positive, then the matrix $\mathbf{D}$ defined above is $\mathcal{P}$ -diagonal where $\mathcal{P} = \text{gen}(\mathcal{T})$.*

Proof. First, we show that $\mathbf{D}$ is diagonal. Let $i \neq j$ be two distinct indices in $[r]$. If $B_i \cap A_j = \emptyset$, then $D_{i,j}$ is trivially 0. Otherwise, for every $k \in B_i \cap A_j$, we know that (k, k) is not in the pattern defined by $\mathcal{T}$ because $k \in B_i$, $k \in A_j$ but $i \neq j$. As a result, we have $w_k = 0$ which implies $D_{i,j} = 0$ for all $i \neq j \in [r]$.

Second, if $A_i \cap B_i \neq \emptyset$ then (k, k) is in the pattern defined by $\mathcal{T}$ for every $k \in A_i \cap B_i$. This implies that $w_k > 0$. As a result, we have $D_{i,i} > 0$ if and only if $A_i \cap B_i \neq \emptyset$. $\square$

It follows that $\mathfrak{Y}^*$ contains $\mathcal{P}$ -matrices and $\mathcal{P}$ -diagonal matrices only, where $\mathcal{P} = \text{gen}(\mathcal{T})$.

4.2 Definition of $\mathfrak{X}^*$

Now we define $\mathfrak{X}^*$. To this end, we first define $\mathfrak{X}^\#$ which is a set of r -dimensional positive and $\mathcal{P}$ -weakly positive vectors. We have $\mathbf{w}^\# \in \mathfrak{X}^\#$ if and only if one of the following four cases is true:

1. $\mathbf{w}^\# = \mathbf{1}$;
2. There exist a finite subset $\{\mathbf{C}^{[1]}, \dots, \mathbf{C}^{[g]}\} \subseteq \mathfrak{Y}$ with $g \geq 1$, positive integers $s_1, \dots, s_g$ and a vector $\mathbf{w} \in \mathfrak{X}$ (positive or $\mathcal{T}$ -weakly positive) such that: Let $(\boldsymbol{\alpha}^{[i]}, \boldsymbol{\beta}^{[i]})$ be the representation of $\mathbf{C}^{[i]}$, then

$$w_i^\# = \sum_{x \in A_i} \left(\alpha_x^{[1]} \right)^{s_1} \cdots \left(\alpha_x^{[g]} \right)^{s_g} \cdot w_x, \quad \text{for all } i \in [r].$$

We have that $\mathbf{w}^\#$ is positive if $\mathbf{w}$ is positive and $\mathbf{w}^\#$ is $\mathcal{P}$ -weakly positive if $\mathbf{w}$ is $\mathcal{T}$ -weakly positive.

3. There exist a finite subset $\{\mathbf{D}^{[1]}, \dots, \mathbf{D}^{[h]}\} \subseteq \mathfrak{Y}$ with $h \geq 1$, positive integers $t_1, \dots, t_h$ and a vector $\mathbf{w} \in \mathfrak{X}$ (positive or $\mathcal{T}$ -weakly positive) such that: Let $(\gamma^{[i]}, \delta^{[i]})$ be the representation of $\mathbf{D}^{[i]}$, then

$$w_i^\# = \sum_{x \in B_i} \left(\delta_x^{[1]} \right)^{t_1} \cdots \left(\delta_x^{[h]} \right)^{t_h} \cdot w_x, \quad \text{for all } i \in [r].$$

Similarly, it can be checked that $\mathbf{w}^\#$ is positive if $\mathbf{w}$ is positive and $\mathbf{w}^\#$ is $\mathcal{P}$ -weakly positive if $\mathbf{w}$ is $\mathcal{T}$ -weakly positive.

4. There exist two finite subsets $\{\mathbf{C}^{[1]}, \dots, \mathbf{C}^{[g]}\} \subseteq \mathfrak{Y}$ and $\{\mathbf{D}^{[1]}, \dots, \mathbf{D}^{[h]}\} \subseteq \mathfrak{Y}$ with $g \geq 1$ and $h \geq 1$, positive integers $s_1, \dots, s_g, t_1, \dots, t_h$ and a vector $\mathbf{w} \in \mathfrak{X}$ (positive or $\mathcal{T}$ -weakly positive) such that: Let $(\alpha^{[i]}, \beta^{[i]})$ and $(\gamma^{[i]}, \delta^{[i]})$ be the representations of $\mathbf{C}^{[i]}$ and $\mathbf{D}^{[i]}$, respectively, then

$$w_i^\# = \sum_{x \in B_i \cap A_i} \left(\beta_x^{[1]} \right)^{s_1} \cdots \left(\beta_x^{[g]} \right)^{s_g} \cdot \left(\gamma_x^{[1]} \right)^{t_1} \cdots \left(\gamma_x^{[h]} \right)^{t_h} \cdot w_x, \quad \text{for all } i \in [r].$$

It can be checked that $\mathbf{w}^\#$ is always a $\mathcal{P}$ -weakly positive vector.

This finishes the definition of $\mathfrak{X}^\#$.

Set $\mathfrak{X}^*$ is the *closure* of $\mathfrak{X}^\#$: $\mathbf{w} \in \mathfrak{X}^*$ if and only if there exist a finite subset $\{\mathbf{w}_1, \dots, \mathbf{w}_g\} \subseteq \mathfrak{X}^\#$ and positive integers $s_1, \dots, s_g$ such that

$$\mathbf{w} = (\mathbf{w}_1)^{s_1} \circ \cdots \circ (\mathbf{w}_g)^{s_g},$$

where $\circ$ denotes the Hadamard product and $(\mathbf{w}_i)^{s_i}$ denote the vector in which the j th entry is the s_i -th power of the j th entry of $\mathbf{w}_i$. It follows that $\mathfrak{X}^*$ is closed and any vector in it is either positive or $\mathcal{P}$ -weakly positive. It is also easy to check that $(\mathfrak{X}^*, \mathfrak{Y}^*)$ is a *generalized* $\mathcal{P}$ -pair.

4.3 Definition of $(\mathfrak{X}', \mathfrak{Y}')$

We use $(\mathfrak{X}^*, \mathfrak{Y}^*)$ to define $(\mathfrak{X}', \mathfrak{Y}')$ as follows.

First, $\mathfrak{Y}'$ contains exactly all the $\mathcal{P}$ -matrices in $\mathfrak{Y}^*$.

The definition of $\mathfrak{X}'$ is more complicated. We have $\mathbf{w}' \in \mathfrak{X}'$ if and only if

1. $\mathbf{w}' \in \mathfrak{X}^*$; or
2. There exist
 - (a) a finite subset of $\mathcal{P}$ -matrices $\{\mathbf{C}^{[1]}, \dots, \mathbf{C}^{[g]}\} \subseteq \mathfrak{Y}^*$ with $g \geq 0$ (so this set could be empty) and g positive integers $s_1, \dots, s_g$;
 - (b) a finite subset of $\mathcal{P}$ -diagonal matrices $\{\mathbf{D}^{[1]}, \dots, \mathbf{D}^{[h]}\} \subseteq \mathfrak{Y}^*$ with $h \geq 1$, and h positive integers $t_1, \dots, t_h$;
 - (c) and a vector $\mathbf{w} \in \mathfrak{X}^*$ (which is either positive or $\mathcal{P}$ -weakly positive),

such that $\mathbf{w}'$ satisfies

$$w'_i = w_i \cdot \left(C_{i,i}^{[1]} \right)^{s_1} \cdots \left(C_{i,i}^{[g]} \right)^{s_g} \cdot \left(D_{i,i}^{[1]} \right)^{t_1} \cdots \left(D_{i,i}^{[h]} \right)^{t_h}, \quad \text{for any } i \in [r].$$

It can be checked that every $\mathbf{w}' \in \mathfrak{X}'$ is either positive or $\mathcal{P}$ -weakly positive.

This finishes the definition of $(\mathfrak{X}', \mathfrak{Y}')$ and the **gen-pair** operation. It is easy to verify that the new pair $(\mathfrak{X}', \mathfrak{Y}')$ is a $\mathcal{P}$ -pair. Moreover, since $\mathfrak{X}^*$ is closed, one can show that $\mathfrak{X}'$ is also closed. This proves the first part of Lemma 1:

Lemma 6. *Let $(\mathfrak{X}, \mathfrak{Y})$ be a $\mathcal{T}$ -pair for some non-trivial block pattern $\mathcal{T}$. Suppose every matrix in $\mathfrak{Y}$ is block-rank-1, then $(\mathfrak{X}', \mathfrak{Y}') = \text{gen-pair}(\mathfrak{X}, \mathfrak{Y})$ is a $\mathcal{P}$ -pair, where $\mathcal{P} = \text{gen}(\mathcal{T})$, and $\mathfrak{X}'$ is closed. Moreover, the pair $(\mathfrak{X}^*, \mathfrak{Y}^*)$ defined from $(\mathfrak{X}, \mathfrak{Y})$ is a generalized $\mathcal{P}$ -pair and $\mathfrak{X}^*$ is also closed.*

5 Dichotomy: Tractability

In this section, we prove Lemma 2 (restated below), the tractability part of the dichotomy theorem.

Lemma 2 (Tractability). *Given any block pattern $\mathcal{T}$ and a finite $\mathcal{T}$ -pair $(\mathfrak{P}, \mathfrak{Q})$, let $(\mathfrak{X}_0, \mathfrak{Y}_0), \dots, (\mathfrak{X}_h, \mathfrak{Y}_h)$ be a sequence of pairs that satisfies condition **(R)** for some $h < m$, with $(\mathfrak{X}_0, \mathfrak{Y}_0) = (\mathfrak{P}, \mathfrak{Q})$. Then we have that $Z_{\mathfrak{P}, \mathfrak{Q}}(\cdot)$ is computable in polynomial time.*

Let $(\mathfrak{X}_0, \mathfrak{Y}_0) = (\mathfrak{P}, \mathfrak{Q})$ be a finite $\mathcal{T}_0$ -pair, for some block pattern $\mathcal{T}_0$. Let $(\mathfrak{X}_0, \mathfrak{Y}_0), \dots, (\mathfrak{X}_h, \mathfrak{Y}_h)$ be a sequence of $h + 1$ pairs for some $h \geq 0$, $m_0 > m_1 > \dots > m_h \geq 1$ be $h + 1$ positive integers, and $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_h$ be $h + 1$ block patterns such that

- R:** For every $i \in [0 : h]$, $\mathcal{T}_i$ is an $m_i \times m_i$ block pattern;
- For every $i \in [h]$, $\mathcal{T}_i$ is consistent with $\mathcal{P}_i = \text{gen}(\mathcal{T}_{i-1})$;
- Either $\mathcal{T}_h = \emptyset$ is trivial or every set in $\mathcal{T}_h$ is a singleton;
- For every $i \in [h]$, $(\mathfrak{X}_i, \mathfrak{Y}_i) = \text{gen-pair}(\mathfrak{X}_{i-1}, \mathfrak{Y}_{i-1})$ is a $\mathcal{T}_i$ -pair; and
- For every $i \in [0 : h]$, all the matrices in $\mathfrak{Y}_i$ are block-rank-1.

We need to show that $Z_{\mathfrak{P}, \mathfrak{Q}}(\cdot) = Z_{\mathfrak{X}_0, \mathfrak{Y}_0}(\cdot)$ can be computed in polynomial time.

Let $\mathcal{G}_0 = (G_0, \mathcal{V}_0, \mathcal{E}_0)$ be an input labeled directed graph of $Z_{\mathfrak{X}_0, \mathfrak{Y}_0}(\cdot)$. By definition we have $\mathcal{V}_0(v) \in \mathfrak{X}_0$ for all vertices $v \in V(G_0)$, and $\mathcal{E}_0(uv) \in \mathfrak{Y}_0$ for all edges $uv \in E(G_0)$. We further assume that the underlying undirected graph of G_0 is connected. (If G_0 is not connected, then we only need to compute $Z_{\mathfrak{X}_0, \mathfrak{Y}_0}(\cdot)$ for each undirected connected component of G_0 and multiply them to obtain $Z_{\mathfrak{X}_0, \mathfrak{Y}_0}(\mathcal{G}_0)$.)

To compute $Z_{\mathfrak{X}_0, \mathfrak{Y}_0}(\mathcal{G}_0)$, we will construct in polynomial-time a sequence of $h + 1$ labeled directed graphs $\mathcal{G}_0, \dots, \mathcal{G}_h$. We will show that these graphs have the following two properties:

- P₁:** For every $\ell \in [0 : h]$, $\mathcal{G}_\ell = (G_\ell, \mathcal{V}_\ell, \mathcal{E}_\ell)$ is a labeled directed graph such that $\mathcal{V}_\ell(v) \in \mathfrak{X}_\ell$ for all $v \in V(G_\ell)$; $\mathcal{E}_\ell(uv) \in \mathfrak{Y}_\ell$ for all $uv \in E(G_\ell)$; and the underlying undirected graph of G_ℓ is connected.
- P₂:** $Z(\mathcal{G}_0) = Z(\mathcal{G}_1) = \dots = Z(\mathcal{G}_h)$.

As a result, to compute $Z(\mathcal{G}_0)$, one only needs to compute $Z(\mathcal{G}_h)$. On the other hand, we do know how to compute $Z(\mathcal{G}_h)$ in polynomial time. If $\mathcal{T}_h$ is trivial, then computing $Z(\mathcal{G}_h)$ is trivial. Otherwise, if every set in $\mathcal{T}_h$ is a singleton, then one can efficiently enumerate all assignments of $\mathcal{G}_h$ with a positive weight. (To see this we note that for any edge uv in G_h and any assignment of u , there is at most one assignment of v such that the edge weight of uv is positive. As G_h is connected, each assignment of u can be extended to at most one assignment of vertices of G_h with a positive weight, and this extension can be computed efficiently.) This allows us to compute $Z(\mathcal{G}_0) = Z(\mathcal{G}_h)$ in polynomial time.

5.1 Construction of $\mathcal{G}'$ from $\mathcal{G}$

Let $(\mathfrak{X}, \mathfrak{Y})$ be a $\mathcal{T}$ -pair for some $m \times m$ non-trivial block pattern $\mathcal{T}$ such that all the matrices in $\mathfrak{Y}$ are block-rank-1. Then by Lemma 6, $(\mathfrak{X}', \mathfrak{Y}') = \text{gen-pair}(\mathfrak{X}, \mathfrak{Y})$ is a $\mathcal{P}$ -pair where $\mathcal{P} = \text{gen}(\mathcal{T})$.

Let $\mathcal{G} = (G, \mathcal{V}, \mathcal{E})$ be a labeled directed graph such that $\mathcal{V}(v) \in \mathfrak{X}$ for all $v \in V(G)$; $\mathcal{E}(uv) \in \mathfrak{Y}$ for all $uv \in E(G)$; and the underlying undirected graph of G is connected. We further assume that G is not trivial: V is not a singleton (since for this special case, $Z(\mathcal{G})$ can be computed trivially). In this section,

we show how to construct a new graph $\mathcal{G}' = (G', \mathcal{V}', \mathcal{E}')$ in polynomial time such that $\mathcal{V}'(v) \in \mathfrak{X}'$ for all $v \in V(G')$; $\mathcal{E}'(uv) \in \mathfrak{Y}'$ for all $uv \in E(G')$; the underlying undirected graph of G' is connected; and

$$Z(\mathcal{G}) = Z(\mathcal{G}'). \quad (2)$$

Then we can repeatedly apply this construction, starting from $\mathcal{G}_0$, to obtain a sequence of $h + 1$ labeled directed graphs $\mathcal{G}_0, \dots, \mathcal{G}_h$ that satisfy both $\mathbf{P}_1$ and $\mathbf{P}_2$. Lemma 2 then follows.

Now we describe the construction of $\mathcal{G}'$. Let $G = (V, E)$ and $\mathcal{T} = \{(A_1, B_1), \dots, (A_n, B_n)\}$ for some $n \geq 1$, then $\mathcal{P} = \text{gen}(\mathcal{T})$ is an $n \times n$ pattern. The construction of $\mathcal{G}'$ is divided into two steps, just like the definition of $(\mathfrak{X}', \mathfrak{Y}') = \text{gen-pair}(\mathfrak{X}, \mathfrak{Y})$ in Section 4. In the first step, we construct a labeled graph $\mathcal{G}^* = (G^*, \mathcal{V}^*, \mathcal{E}^*)$ from $\mathcal{G}$ such that

1. $\mathcal{V}^*(v) \in \mathfrak{X}^*$ for all $v \in V(G^*)$; $\mathcal{E}^*(uv) \in \mathfrak{Y}^*$ for all $uv \in E(G^*)$; and the underlying undirected graph of G^* is connected, where $(\mathfrak{X}^*, \mathfrak{Y}^*)$ denotes the *generalized* $\mathcal{P}$ -pair defined in Section 4.
2. $Z(\mathcal{G}^*) = Z(\mathcal{G})$.

In the second step, we construct $\mathcal{G}'$ from $\mathcal{G}^*$ and show that $Z(\mathcal{G}') = Z(\mathcal{G}^*)$.

5.1.1 Construction of $\mathcal{G}^*$ from $\mathcal{G}$

Let $\mathcal{G} = (G, \mathcal{V}, \mathcal{E})$ and $G = (V, E)$. We decompose the edge set using the following equivalence relation:

Definition 7. Let e, e' be two directed edges in E . We say $e \sim e'$ if $e = e'$ or there is a sequence of edges

$$e = e_0, e_1, \dots, e_k = e'$$

in E such that for all $i \in [0 : k - 1]$, e_i and e_{i+1} share either the same head or the same tail.

We divide E into equivalence classes $R_1, \dots, R_f$ using $\sim$:

$$E = R_1 \cup \dots \cup R_f, \quad \text{for some } f \geq 1.$$

Because the underlying undirected graph of G is connected, there is no isolated vertex v in G and thus every vertex $v \in V$ appears as an incident vertex of some edge in at least one of the equivalence classes. This equivalence relation is useful because of the following observation.

Observation 1. For any $i \in [f]$, the subgraph spanned by R_i is connected if we view it as an undirected graph. There are three types of vertices in it:

1. Type-L: vertices which only have outgoing edges in R_i ;
2. Type-R: vertices which only have incoming edges in R_i ; and
3. Type-M: vertices which have both incoming and outgoing edges in R_i .

Let $\xi : V \rightarrow [m]$ be any assignment with $\text{wt}(\mathcal{G}, \xi) \neq 0$, then for any $i \in [f]$ there exists a unique $k_i \in [n]$ such that the value of every edge $uv \in R_i$ is derived from the k_i -th block of $\mathcal{T}$:

$$\xi(u) \in A_{k_i} \quad \text{and} \quad \xi(v) \in B_{k_i}.$$

Therefore, for every $i \in [f]$, there exists a unique $k_i \in [n]$ such that

1. For every Type-L vertex v in the graph spanned by R_i , $\xi(v) \in A_{k_i}$;

2. For every Type-R vertex v in the graph spanned by R_i , $\xi(v) \in B_{k_i}$; and
3. For every Type-M vertex v in the graph spanned by R_i , $\xi(v) \in A_{k_i} \cap B_{k_i}$.

Now we build $\mathcal{G}^* = (G^*, \mathcal{V}^*, \mathcal{E}^*)$, where $G^* = (V^*, E^*)$. The next observation is important.

Observation 2. Each vertex $v \in V$ can appear in no more than two subgraphs spanned by R_i 's. To see this, assume for a contradiction that v appears in three subgraphs spanned by three distinct subsets of edges R_i , R_j and R_k . Then there exist three distinct edges $e \in R_i$, $e' \in R_j$ and $e'' \in R_k$ such that v is incident to all of them, either as head or as tail. So v must be the head of at least two of them, or the tail of at least two of them. Then at least two of e , e' and e'' are equivalent under $\sim$, and R_i , R_j and R_k are not three distinct equivalence classes. This is a contradiction.

We start with the construction of G^* . V^* is exactly $[f]$ in which the vertex $i \in [f]$ corresponds to R_i of G . For each vertex $v \in V$, if it appears in two subgraph spanned by R_i and R_j for some $i \neq j \in [f]$ and if the incoming edges of v are from R_i and the outgoing edges of v are from R_j , then we add a directed edge ij in E^* . Note that E^* may have parallel edges. This finishes the construction of G^* . It is easy to verify that the underlying undirected graph of G^* is also *connected*.

The only thing left is to label the graph G^* with vertex and edge weights. For every edge in E^* we assign it the following $n \times n$ matrix. Assume that an edge ij is created because of a vertex $v \in V$ which appears in both R_i and R_j . Let the incoming edges of v be $u_1v, \dots, u_sv$ in R_i and the outgoing edges of v be $vw_1, \dots, vw_t$ in R_j , where $s, t \geq 1$. We use $\mathbf{C}^{[k]} \in \mathfrak{Y}$ to denote the edge weight of u_kv , $\mathbf{D}^{[k]} \in \mathfrak{Y}$ to denote the edge weight of vw_k , and $\mathbf{w} \in \mathfrak{X}$ to denote the vertex weight of v in $\mathcal{G}$. We use $(\alpha^{[k]}, \beta^{[k]})$ and $(\gamma^{[k]}, \delta^{[k]})$ to denote the representations of $\mathbf{C}^{[k]}$ and $\mathbf{D}^{[k]}$, respectively. Then the (k, ℓ) th entry of $\mathbf{D}$ is

$$D_{k,\ell} = \sum_{x \in B_k \cap A_\ell} \beta_x^{[1]} \dots \beta_x^{[s]} \cdot \gamma_x^{[1]} \dots \gamma_x^{[t]} \cdot w_x, \quad \text{for all } k, \ell \in [n].$$

By the definition of **gen-pair**, it is easy to check that $\mathbf{D} \in \mathfrak{Y}^*$.

Finally, we define the vertex weight of $i \in [f]$. To this end, we first define an n -dimensional vector $\mathbf{w}^{[v]}$ for each vertex $v \in V$ that *only appears* in R_i . We then multiply (using Hadamard product) all such vectors to get the vertex weight vector of $i \in [f]$.

Let $v \in V$ be a vertex which only appears in R_i , then we have the following three cases:

1. If v is Type-L, then we use $vw_1, \dots, vw_s$ to denote its outgoing edges. We let $\mathbf{w}$ denote the vertex weight of v in $\mathcal{G}$ and $\mathbf{C}^{[j]}$ denote the edge weight of vw_j with representation $(\alpha^{[j]}, \beta^{[j]})$. Then

$$w_k^{[v]} = \sum_{x \in A_k} \alpha_x^{[1]} \dots \alpha_x^{[s]} \cdot w_x, \quad \text{for all } k \in [n].$$

2. If v is Type-R, then we use $u_1v, \dots, u_sv$ to denote its incoming edges. We let $\mathbf{w}$ denote the vertex weight of v in $\mathcal{G}$ and $\mathbf{C}^{[j]}$ denote the edge weight of u_jv with representation $(\alpha^{[j]}, \beta^{[j]})$. Then

$$w_k^{[v]} = \sum_{x \in B_k} \beta_x^{[1]} \dots \beta_x^{[s]} \cdot w_x, \quad \text{for all } k \in [n].$$

3. If v is Type-M, then we use $u_1v, \dots, u_sv, vw_1, \dots, vw_t$ to denote its edges where $s, t \geq 1$. We let $\mathbf{w}$ be the vertex weight of v in $\mathcal{G}$, $\mathbf{C}^{[j]}$ be the edge weight of u_jv with representation $(\alpha^{[j]}, \beta^{[j]})$, and $\mathbf{D}^{[j]}$ be the edge weight of vw_j with representation $(\gamma^{[j]}, \delta^{[j]})$. Then

$$w_k^{[v]} = \sum_{x \in B_k \cap A_k} \beta_x^{[1]} \dots \beta_x^{[s]} \cdot \gamma_x^{[1]} \dots \gamma_x^{[t]} \cdot w_x, \quad \text{for all } k \in [n].$$

We then multiply (using Hadamard product) all the vectors $\mathbf{w}^{[v]}$ over all vertices v that only appear in R_i to get the vertex weight vector $\mathbf{w}$ of $i \in [f]$ in $\mathcal{G}^*$. By definition we have that $\mathbf{w} \in \mathfrak{X}^*$, and this finishes the construction of $\mathcal{G}^*$. Note that both G^* and edge and vertex weights of $\mathcal{G}^*$ can be computed in polynomial time (in the input size of $\mathcal{G}$). Next, we show that $Z(\mathcal{G}^*) = Z(\mathcal{G})$.

Let $\phi : V^* = [f] \rightarrow [n]$ be any assignment. We use Ξ_ϕ to denote

$$\left\{ \xi : V \rightarrow [m] \mid \forall i \in [f], \forall uv \in R_i, \xi(u) \in A_{\phi(i)} \text{ and } \xi(v) \in B_{\phi(i)} \right\}.$$

Equivalently, ϕ defines for each vertex $v \in V$ a set $U_v \subseteq [m]$, where

1. If v appears in both the subgraph spanned by R_i and the subgraph spanned by R_j , for some $i \neq j \in [f]$; and v is Type-R in R_i and Type-L in R_j , then $U_v = B_{\phi(i)} \cap A_{\phi(j)}$;
2. Otherwise, assume v only appears in the subgraph spanned by R_i . Then
 - (a) If v is Type-L, then $U_v = A_{\phi(i)}$;
 - (b) If v is Type-R, then $U_v = B_{\phi(i)}$; and
 - (c) If v is Type-M, then $U_v = B_{\phi(i)} \cap A_{\phi(i)}$,

such that $\xi \in \Xi_\phi \iff \xi(v) \in U_v$ for all $v \in V$. In particular, $\Xi_\phi = \emptyset$ if $U_v = \emptyset$ for some $v \in V$.

By Observation 1, if $\text{wt}(\mathcal{G}, \xi) \neq 0$ then $\xi \in \Xi_\phi$ for some unique ϕ . For any $v \in V$, we let $\mathbf{w}^{[v]}$ denote its vertex weight in $\mathcal{G}$; and for any $uv \in E$, we let $\mathbf{D}^{[uv]}$ denote its edge weight in $\mathcal{G}$, with representation $(\alpha^{[uv]}, \beta^{[uv]})$. Then by the definition of Ξ_ϕ , we have for all $\xi \in \Xi_\phi$,

$$D_{\xi(u), \xi(v)}^{[uv]} = \alpha_{\xi(u)}^{[uv]} \cdot \beta_{\xi(v)}^{[uv]}, \quad \text{for all } uv \in E.$$

Therefore, we have the following equation:

$$\sum_{\xi \in \Xi_\phi} \text{wt}(\mathcal{G}, \xi) = \sum_{\xi \in \Xi_\phi} \left(\prod_{v \in V} w_{\xi(v)}^{[v]} \prod_{uv \in E} \alpha_{\xi(u)}^{[uv]} \cdot \beta_{\xi(v)}^{[uv]} \right).$$

This sum can be written as a product:

$$\sum_{\xi \in \Xi_\phi} \text{wt}(\mathcal{G}, \xi) = \prod_{v \in V} H_v,$$

in which for every $v \in V$, the factor H_v is a sum over $\xi(v) \in U_v$.

By the construction of $\mathcal{G}^*$, we can show that

$$\text{wt}(\mathcal{G}^*, \phi) = \sum_{\xi \in \Xi_\phi} \text{wt}(\mathcal{G}, \xi) = \prod_{v \in V} H_v. \quad (3)$$

This follows from the following observations:

1. If v appears in both the subgraph spanned by R_i and the subgraph spanned by R_j , for some $i \neq j \in [n]$, and this v defines an edge $ij \in E^*$, then the edge weight of this edge ij in $\mathcal{G}^*$ with respect to ϕ is exactly H_v ;
2. For every $i \in [n]$, we let $V_i \subseteq V$ denote the set of vertices that only appear in the subgraph spanned by R_i . We also let $\mathbf{w}$ denote the vertex weight of $i \in [n]$ in $\mathcal{G}^*$. Then we have

$$w_{\xi(i)} = \prod_{v \in V_i} H_v.$$

As a result, it follows from (3) that

$$Z(\mathcal{G}^*) = \sum_{\phi} \text{wt}(\mathcal{G}^*, \phi) = \sum_{\phi} \sum_{\xi \in \Xi_{\phi}} \text{wt}(\mathcal{G}, \xi) = Z(\mathcal{G}).$$

5.1.2 Construction of $\mathcal{G}'$ from $\mathcal{G}^*$

Let $\mathcal{G}^* = (G^*, \mathcal{V}^*, \mathcal{E}^*)$ be the labeled directed graph constructed above, where $G^* = (V^*, E^*)$. We know that $\mathcal{V}^*(v) \in \mathfrak{X}^*$ for all $v \in V^*$; $\mathcal{E}^*(uv) \in \mathfrak{Y}^*$ for all $uv \in E^*$; the underlying undirected graph of G^* is connected. As $(\mathfrak{X}^*, \mathfrak{Y}^*)$ is a generalized $\mathcal{P}$ -pair, each $\mathbf{D} \in \mathfrak{Y}^*$ is either a $\mathcal{P}$ -matrix or a $\mathcal{P}$ -diagonal matrix.

We will build a new labeled directed graph $\mathcal{G}' = (G', \mathcal{V}', \mathcal{E}')$ with $G' = (V', E')$ such that $\mathcal{V}'(v) \in \mathfrak{X}'$ for all $v \in V'$; $\mathcal{E}'(uv) \in \mathfrak{Y}'$ for all $uv \in E'$; the underlying undirected graph of G' is connected; and

$$Z(\mathcal{G}') = Z(\mathcal{G}^*).$$

Let $E^* = E_0 \cup E_1$, where E_0 consists of edges in E^* whose weight is a $\mathcal{P}$ -matrix and E_1 consists of edges in E^* whose weight is a $\mathcal{P}$ -diagonal matrix. We write $V_1, \dots, V_g$, for some $g \geq 1$, to denote connected components of (V^*, E_1) , where we view E_1 as a set of *undirected* edges and (V^*, E_1) as an undirected graph. Then we have the following observation:

Observation 3. Let $\phi : V^* \rightarrow [n]$ be an assignment with non-zero weight: $\text{wt}(\mathcal{G}^*, \phi) \neq 0$. Then for any $i \in [g]$, there exists a unique $k_i \in [n]$ such that $\phi(v) = k_i$ for all $v \in V_i$.

Now we construct $\mathcal{G}' = (G', \mathcal{V}', \mathcal{E}')$. First we construct $G' = (V', E')$. V' is exactly $[g]$ in which vertex $i \in [g]$ corresponds to V_i . For every edge $uv \in E_0$ such that $u \in V_i$, $v \in V_j$, and $i \neq j \in [g]$, we add an edge from i to j in G' . This finishes the construction of G' . It is easy to verify that the underlying undirected graph of G' is also *connected*.

Finally, we assign vertex and edge weights. For each edge ij in G' , suppose it is created because of $uv \in E_0$. Then the edge weight of ij is the same as that of uv . As a result, all the edge weight matrices of $\mathcal{G}'$ come from $\mathfrak{Y}'$ (since by definition of **gen-pair**, $\mathfrak{Y}'$ contains all the $\mathcal{P}$ -matrices in $\mathfrak{Y}^*$).

We define the vertex weights of $\mathcal{G}'$ as follows. If $V_i = \{v\}$ is a singleton, then the vertex weight of i in $\mathcal{G}'$ is the same as the weight of v in $\mathcal{G}^*$. Otherwise, we let $v_1, \dots, v_r$ be the vertices in V_i with $r > 1$, let $e_1, \dots, e_s$ be the edges in E_1 with both vertices in V_i for some $s \geq 1$, and let $e'_1, \dots, e'_t$ be the edges in E_0 with both vertices in V_i for some $t \geq 0$. We use $\mathbf{w}^{[j]} \in \mathfrak{X}^*$ to denote the vertex weight of v_j in $\mathcal{G}'$ $\mathbf{C}^{[j]} \in \mathfrak{Y}^*$ to denote the $\mathcal{P}$ -diagonal matrix of e_j and $\mathbf{D}^{[j]} \in \mathfrak{Y}^*$ to denote the $\mathcal{P}$ -matrix of e'_j . Then we assign the following vertex weight vector $\mathbf{w}$ to $i \in V'$:

$$w_k = w_k^{[1]} \cdots w_k^{[r]} \cdot C_{k,k}^{[1]} \cdots C_{k,k}^{[s]} \cdot D_{k,k}^{[1]} \cdots D_{k,k}^{[t]}, \quad \text{for every } k \in [n].$$

By definition $\mathbf{w} \in \mathfrak{Y}'$. We also have that $\mathcal{G}'$ can be computed in polynomial time (in the input size of $\mathcal{G}^*$). Using Observation 3, it is easy to verify that $Z(\mathcal{G}') = Z(\mathcal{G}^*)$.

This completes the proof of Lemma 2.

6 Reduction: Normalized Matrices are Free to Use

To give a polynomial-time reduction from $(\mathfrak{X}', \mathfrak{Y}') = \text{gen-pair}(\mathfrak{X}, \mathfrak{Y})$ to $(\mathfrak{X}, \mathfrak{Y})$, we need to first prove a technical lemma on *normalized* block-rank-1 matrices as defined below.

Let $\mathbf{C}$ be an $m \times m$ block-rank-1 matrix of block pattern $\mathcal{T}$ and representation $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, where $\mathcal{T} = \{(A_1, B_1), \dots, (A_r, B_r)\}$ for some $r \geq 1$. By definition, $\boldsymbol{\alpha}$ satisfies

$$\sum_{j \in A_i} \alpha_j = 1, \quad \text{for all } i \in [r].$$

We say $\mathbf{C}'$ is the *normalized* version of $\mathbf{C}$ if it is an $m \times m$ block-rank-1 matrix of block pattern $\mathcal{T}$ and representation $(\boldsymbol{\alpha}, \boldsymbol{\delta})$, where

$$\delta_j = \frac{\beta_j}{\sum_{k \in B_i} \beta_k}, \quad \text{for all } j \in B_i \text{ and } i \in [r],$$

so that $\boldsymbol{\delta}$ also satisfies

$$\sum_{j \in B_i} \delta_j = 1, \quad \text{for all } i \in [r].$$

Let $(\mathfrak{P}, \mathfrak{Q})$ be a finite $\mathcal{T}$ -pair for some non-trivial $m \times m$ block pattern $\mathcal{T}$, and

$$\mathfrak{Q} = \{\mathbf{C}^{[1]}, \dots, \mathbf{C}^{[s]}\},$$

in which every $\mathbf{C}^{[i]}$ is block-rank-1 and has representation $(\boldsymbol{\alpha}^{[i]}, \boldsymbol{\beta}^{[i]})$. For each $i \in [s]$, we let $\mathbf{D}^{[i]}$ denote the normalized version of $\mathbf{C}^{[i]}$ with representation $(\boldsymbol{\alpha}^{[i]}, \boldsymbol{\delta}^{[i]})$, and

$$\mathfrak{Q}' = \{\mathbf{C}^{[1]}, \dots, \mathbf{C}^{[s]}, \mathbf{D}^{[1]}, \dots, \mathbf{D}^{[s]}\}.$$

In the rest of this section, we show in Lemma 7 that $Z_{\mathfrak{P}, \mathfrak{Q}}(\cdot)$ and $Z_{\mathfrak{P}, \mathfrak{Q}'}(\cdot)$ are computationally equivalent. It will be crucially used in Section 7, where we give a polynomial-time reduction from $(\mathfrak{X}', \mathfrak{Y}')$ to $(\mathfrak{X}, \mathfrak{Y})$. To obtain such a reduction, it follows from Lemma 7 that it suffices to give a polynomial-time reduction from $(\mathfrak{X}', \mathfrak{Y}')$ to $(\mathfrak{X}, \mathfrak{Y}^\dagger)$ where $\mathfrak{Y}^\dagger$ contains all matrices in $\mathfrak{Y}$ as well as their normalized versions.

Lemma 7. *$Z_{\mathfrak{P}, \mathfrak{Q}}(\cdot)$ and $Z_{\mathfrak{P}, \mathfrak{Q}'}(\cdot)$ are computationally equivalent.*

Proof. In the proof, we use two levels of interpolations and Vandermonde systems.

We start with some notation. Let $\mathcal{G} = (G, \mathcal{V}, \mathcal{E})$ be the input labeled directed graph of $Z_{\mathfrak{P}, \mathfrak{Q}'}(\cdot)$ with $G = (V, E)$. For $v \in V$, we use $\mathbf{w}^{[v]} \in \mathfrak{P}$ to denote its vertex weight. We use $E_i \subseteq E$, $i \in [s]$, to denote the set of edges labeled with $\mathbf{C}^{[i]}$, and $F_i \subseteq E$, $i \in [s]$, to denote the set of edges labeled with $\mathbf{D}^{[i]}$. For every assignment $\xi : V \rightarrow [m]$, we define

$$\text{vw}(\xi) = \prod_{v \in V} w_{\xi(v)}^{[v]}, \quad \text{cw}(\xi) = \prod_{i \in [s]} \prod_{uv \in E_i} C_{\xi(u), \xi(v)}^{[i]}, \quad \text{dw}(\xi) = \prod_{i \in [s]} \prod_{uv \in F_i} D_{\xi(u), \xi(v)}^{[i]}.$$

Note that a product over an empty set is equal to 1. Then we need to compute the following sum

$$Z_{\mathfrak{P}, \mathfrak{Q}'}(\mathcal{G}) = \sum_{\xi} \text{vw}(\xi) \cdot \text{cw}(\xi) \cdot \text{dw}(\xi).$$

For all $a \in [s]$ and $b \in [r]$, we use $K_b^{[a]} > 0$ to denote the number such that

$$C_{i,j}^{[a]} = K_b^{[a]} \cdot D_{i,j}^{[a]}, \quad \text{for all } i \in A_b \text{ and } j \in B_b.$$

Actually, this gives us the following equation

$$C_{i,j}^{[a]} = K_b^{[a]} \cdot D_{i,j}^{[a]}, \quad \text{for all } i \in A_b \text{ and } j \in [m],$$

since $\mathbf{C}^{[a]}$ and $\mathbf{D}^{[a]}$ have the same block pattern $\mathcal{T}$. Then we use $\text{kw}(\xi)$, where $\xi : V \rightarrow [m]$, to denote

$$\text{kw}(\xi) = \prod_{a \in [s]} \left(\prod_{uv \in F_a \text{ with } \xi(u) \in A_b} K_b^{[a]} \right).$$

We use X to denote the following set:

$$X = \left\{ \prod_{a \in [s]} \prod_{b \in [r]} \left(K_b^{[a]} \right)^{m_{a,b}} : m_{a,b} \text{ are non-negative integers that sum to } |E| \right\}.$$

It is clear that $|X|$ is polynomial in $|E|$, since both s and r are constants, and that X can be computed in polynomial time. We also have that $\text{kw}(\xi) \in X$ for all ξ . Below we use L to denote $|X|$.

For all $k \in [0 : L - 1]$, we build a new graph $\mathcal{G}^{[k]} = (G^{[k]}, \mathcal{V}^{[k]}, \mathcal{E}^{[k]})$ below, where $G^{[k]} = (V^{[k]}, E^{[k]})$:

1. $V^{[k]}$ contains V as a subset, and every $v \in V$ is labeled with the same vertex weight as in $\mathcal{G}$;
2. For all $i \in [s]$ and $uv \in E_i$, we add one edge $uv \in E^{[k]}$ and label it with the same matrix $\mathbf{C}^{[i]}$;
3. For all $i \in [s]$ and all $e = uv \in F_i$, we add $L - k$ parallel edges from u to v with $\mathbf{C}^{[i]}$ as their edge weights; we also add $2k$ new vertices $u_{e,j}$ and $v_{e,j}$, $j \in [k]$, to $V^{[k]}$; we add one edge from u to $u_{e,j}$ and one edge from $v_{e,j}$ to v for all $j \in [k]$, all of which are labeled with $\mathbf{C}^{[i]}$. For each new vertex, we assign $\mathbf{1}$ as its vertex weight.

It is clear that $\mathcal{G}^{[k]}$ can be constructed in polynomial time and is a valid input of $Z_{\mathfrak{P}, \Omega}(\cdot)$.

Fix $k \in [0 : L - 1]$. For every assignment $\phi : V \rightarrow [m]$, we let Ξ_ϕ denote the set of all $\xi : V^{[k]} \rightarrow [m]$ such that $\xi(v) = \phi(v)$ for all $v \in V$. We also define

$$\text{wt}^{[k]}(\phi) = \sum_{\xi \in \Xi_\phi} \text{wt}(\mathcal{G}^{[k]}, \xi).$$

Then we have the following equation

$$Z_{\mathfrak{P}, \Omega}(\mathcal{G}^{[k]}) = \sum_{\xi : V^{[k]} \rightarrow [m]} \text{wt}(\mathcal{G}^{[k]}, \xi) = \sum_{\phi : V \rightarrow [m]} \text{wt}^{[k]}(\phi).$$

By the construction, we show that

$$\text{wt}^{[k]}(\phi) = \text{vw}(\phi) \cdot \text{cw}(\phi) \cdot \left(\text{dw}(\phi) \right)^L \cdot \left(\text{kw}(\phi) \right)^{L+k}, \quad \text{for all } k \in [0 : L - 1]. \quad (4)$$

First, we have

$$\text{wt}^{[k]}(\phi) = \text{vw}(\phi) \cdot \text{cw}(\phi) \cdot \sum_{\xi \in \Xi_\phi} \left(\prod_{i \in [s]} \left(\prod_{e=uv \in F_i} \left(C_{\xi(u), \xi(v)}^{[i]} \right)^{L-k} \left(\prod_{j \in [k]} C_{\xi(u), \xi(u_{e,j})}^{[i]} C_{\xi(v_{e,j}), \xi(v)}^{[i]} \right) \right) \right). \quad (5)$$

For each edge $e = uv \in F_i$ for some $i \in [s]$, there must exist an index $b_e \in [r]$ such that $\phi(u) \in A_{b_e}$ and $\phi(v) \in B_{b_e}$; otherwise both sides of (4) are 0 and we are done. In this case, the sum in (5) becomes

$$\prod_{i \in [s]} \left(\prod_{e=uv \in F_i} \left(K_{b_e}^{[i]} \cdot D_{\xi(u), \xi(v)}^{[i]} \right)^{L-k} \left(\sum_{x \in B_{b_e}} C_{\xi(u), x}^{[i]} \right)^k \left(\sum_{x \in A_{b_e}} C_{x, \xi(v)}^{[i]} \right)^k \right). \quad (6)$$

By the definition of $(\alpha^{[i]}, \beta^{[i]})$ and $(\alpha^{[i]}, \delta^{[i]})$, we have

$$\sum_{x \in B_{b_e}} C_{\xi(u), x}^{[i]} = \alpha_{\xi(u)}^{[i]} \sum_{x \in B_{b_e}} \beta_x^{[i]} = \alpha_{\xi(u)}^{[i]} \cdot K_{b_e}^{[i]} \quad \text{and} \quad \sum_{x \in A_{b_e}} C_{x, \xi(v)}^{[i]} = \beta_{\xi(v)}^{[i]}.$$

As a result, (6) becomes

$$\prod_{i \in [s]} \left(\prod_{e=uv \in F_i} \left(K_{b_e}^{[i]} \cdot D_{\xi(u), \xi(v)}^{[i]} \right)^{L-k} \left(\alpha_{\xi(u)}^{[i]} \cdot K_{b_e}^{[i]} \right)^k \left(\beta_{\xi(v)}^{[i]} \right)^k \right) = \prod_{i \in [s]} \left(\prod_{e=uv \in F_i} \left(K_{b_e}^{[i]} \right)^{L+k} \left(D_{\xi(u), \xi(v)}^{[i]} \right)^L \right).$$

This finishes the proof of equation (4).

Since L is polynomial in the input size, we can use $Z_{\mathfrak{P}, \Omega}(\cdot)$ as an oracle to compute

$$\sum_{\phi: V \rightarrow [m]} \text{vw}(\phi) \cdot \text{cw}(\phi) \cdot \left(\text{dw}(\phi) \right)^L \cdot \left(\text{kw}(\phi) \right)^{L+k}, \quad \text{for all } k \in [0 : L-1].$$

in a polynomial number of steps.

For every $x \in X$, we use Φ_x to denote the set of $\phi : V \rightarrow [m]$ with $\text{kw}(\phi) = x$, then we computed

$$\sum_{x \in X} \left(\sum_{\phi \in \Phi_x} \text{vw}(\phi) \cdot \text{cw}(\phi) \cdot \left(\text{dw}(\phi) \right)^L \right) \cdot x^{L+k}, \quad \text{for all } k \in [0 : L-1].$$

Because $x > 0$ for all $x \in X$, we can solve this Vandermonde system and obtain

$$\sum_{\phi \in \Phi_x} \text{vw}(\phi) \cdot \text{cw}(\phi) \cdot \left(\text{dw}(\phi) \right)^L, \quad \text{for each } x \in X,$$

in a polynomial number of steps.

It is also clear that the whole process can be repeated for any $L' \geq L$ with

$$L' \leq L + \text{poly}(\text{input size}),$$

and we can use $Z_{\mathfrak{P}, \Omega}(\cdot)$ as an oracle to compute

$$\sum_{\phi \in \Phi_x} \text{vw}(\phi) \cdot \text{cw}(\phi) \cdot \left(\text{dw}(\phi) \right)^{L'}, \quad \text{for all } x \in X \text{ and } L \leq L' \leq L + \text{poly}(\text{input size}),$$

in a polynomial number of steps.

Similar to the definition of X for kw earlier, we can define a set Y such that $|Y|$ is polynomial, Y can be computed in polynomial time and contains all possible values of $\text{dw}(\phi)$ for $\phi : V \rightarrow [m]$ (note that it is possible that $0 \in Y$). Let $M = |Y|$. For every $x \in X$, we can compute

$$\sum_{\phi \in \Phi_x} \text{vw}(\phi) \cdot \text{cw}(\phi) \cdot \left(\text{dw}(\phi) \right)^{L+k}, \quad \text{for all } k \in [0 : M-1].$$

Let $\Phi_{x,y}$ denote the set of ϕ with $\text{kw}(\phi) = x$ and $\text{dw}(\phi) = y$. Solving this Vandermonde system, we get

$$\sum_{\phi \in \Phi_{x,y}} \text{vw}(\phi) \cdot \text{cw}(\phi), \quad \text{for all } x \in X \text{ and } 0 < y \in Y.$$

Finally, using all these items, we can compute $Z_{\mathfrak{P}, \Omega'}(\mathcal{G})$ in a polynomial number of steps:

$$Z_{\mathfrak{P}, \Omega'}(\mathcal{G}) = \sum_{x \in X, 0 < y \in Y} \left(\sum_{\phi \in \Phi_{x,y}} \text{vw}(\phi) \cdot \text{cw}(\phi) \right) \cdot y.$$

This proves the lemma since the other direction from $Z_{\mathfrak{P}, \Omega}(\cdot)$ to $Z_{\mathfrak{P}, \Omega'}(\cdot)$ is trivial. $\square$

7 Polynomial-Time Reduction from $(\mathfrak{X}', \mathfrak{Y}')$ to $(\mathfrak{X}, \mathfrak{Y})$

Let $(\mathfrak{X}, \mathfrak{Y})$ be a $\mathcal{T}$ -pair, where $\mathcal{T}$ is a non-trivial $m \times m$ block pattern $\mathcal{T} = \{(A_1, B_1), \dots, (A_r, B_r)\}$ with $r \geq 1$ and every matrix in $\mathfrak{Y}$ is block-rank-1. Let $\mathcal{P}$ be the $r \times r$ pattern where $\mathcal{P} = \text{gen}(\mathcal{T})$ and $(\mathfrak{X}', \mathfrak{Y}')$ be the $\mathcal{P}$ -pair generated from $(\mathfrak{X}, \mathfrak{Y})$ using the **gen-pair** operation: $(\mathfrak{X}', \mathfrak{Y}') = \text{gen-pair}(\mathfrak{X}, \mathfrak{Y})$. We also use $(\mathfrak{X}^*, \mathfrak{Y}^*)$ to denote the generalized $\mathcal{P}$ -pair defined in Section 4.

In this section, we prove that $(\mathfrak{X}', \mathfrak{Y}')$ is polynomial-time reducible to $(\mathfrak{X}, \mathfrak{Y})$. To this end, we first reduce $(\mathfrak{X}', \mathfrak{Y}')$ to $(\mathfrak{X}^*, \mathfrak{Y}^*)$, and then reduce $(\mathfrak{X}^*, \mathfrak{Y}^*)$ to $(\mathfrak{X}, \mathfrak{Y})$. The first step is trivial, so we will only give a polynomial-time reduction from $(\mathfrak{X}^*, \mathfrak{Y}^*)$ to $(\mathfrak{X}, \mathfrak{Y})$ below.

Let $\mathfrak{P}^* = \{\mathbf{p}^{[i]} : i \in [s]\}$ be a finite subset of vectors in $\mathfrak{X}^*$ with $\mathbf{1} \in \mathfrak{P}^*$ and $\mathfrak{Q}^* = \{\mathbf{F}^{[i]} : i \in [t]\}$ be a finite subset of matrices in $\mathfrak{Y}^*$. By the definition of **gen-pair**, they can be generated by a finite subset $\mathfrak{P} = \{\mathbf{w}^{[i]} : i \in [h]\} \subseteq \mathfrak{X}$ with $\mathbf{1} \in \mathfrak{P}$ and a finite subset $\mathfrak{Q} = \{\mathbf{C}^{[i]} : i \in [g]\} \subseteq \mathfrak{Y}$ in the following sense. (We let $(\alpha^{[i]}, \beta^{[i]})$ denote the representation of $\mathbf{C}^{[i]}$ for every $i \in [g]$.)

For every matrix $\mathbf{F} \in \mathfrak{Q}^*$, there exists a $(2g + 1)$ -tuple

$$(k \in [h]; \mathbf{k} = (k_1, \dots, k_g); \ell = (\ell_1, \dots, \ell_g)),$$

where $k_i, \ell_i \geq 0$, $\mathbf{k} \neq \mathbf{0}$ and $\ell \neq \mathbf{0}$, such that

$$F_{i,j} = \sum_{x \in B_i \cap A_j} \left(\beta_x^{[1]}\right)^{k_1} \dots \left(\beta_x^{[g]}\right)^{k_g} \cdot \left(\alpha_x^{[1]}\right)^{\ell_1} \dots \left(\alpha_x^{[g]}\right)^{\ell_g} \cdot w_x^{[k]}. \quad (7)$$

This $(2g + 1)$ -tuple is also call the (not necessarily unique) representation of $\mathbf{F}$ with respect to $(\mathfrak{P}, \mathfrak{Q})$.

For every $\mathbf{p} \in \mathfrak{P}^*$, there exist three finite (and possibly empty) sets $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$ of tuples, where every tuple in $\mathcal{S}_1$ and $\mathcal{S}_2$ is of the form

$$(k \in [h]; \mathbf{k} = (k_1, \dots, k_g))$$

with $k_i \geq 0$ and $\mathbf{k} \neq \mathbf{0}$, and every tuple in $\mathcal{S}_3$ is of the form

$$(k \in [h]; \mathbf{k} = (k_1, \dots, k_g); \ell = (\ell_1, \dots, \ell_g))$$

with $k_i, \ell_i \geq 0$, $\mathbf{k} \neq \mathbf{0}$ and $\ell \neq \mathbf{0}$. Every tuple in $\mathcal{S}_1$ gives us a vector whose i th entry, $i \in [r]$, is equal to

$$\sum_{x \in A_i} \left(\alpha_x^{[1]}\right)^{k_1} \dots \left(\alpha_x^{[g]}\right)^{k_g} \cdot w_x^{[k]};$$

every tuple in $\mathcal{S}_2$ gives us a vector whose i th entry, $i \in [r]$, is equal to

$$\sum_{x \in B_i} \left(\beta_x^{[1]}\right)^{k_1} \dots \left(\beta_x^{[g]}\right)^{k_g} \cdot w_x^{[k]};$$

and every $(2g + 1)$ -tuple in $\mathcal{S}_3$ gives us a vector whose i th entry, $i \in [r]$, is equal to

$$\sum_{x \in B_i \cap A_i} \left(\beta_x^{[1]}\right)^{k_1} \dots \left(\beta_x^{[g]}\right)^{k_g} \cdot \left(\alpha_x^{[1]}\right)^{\ell_1} \dots \left(\alpha_x^{[g]}\right)^{\ell_g} \cdot w_x^{[k]}.$$

Vector $\mathbf{p}$ is then the Hadamard product of all these vectors.

We remark that all exponents k_i, ℓ_i in the equations above are considered as constants because both $(\mathfrak{P}, \mathfrak{Q})$ and $(\mathfrak{P}^*, \mathfrak{Q}^*)$ are fixed (when we are concerned about $Z_{\mathfrak{P}, \mathfrak{Q}}(\cdot)$ and $Z_{\mathfrak{P}^*, \mathfrak{Q}^*}(\cdot)$ as two computational problems). We now prove the following lemma.

Lemma 8. $Z_{\mathfrak{P}^*, \mathfrak{Q}^*}(\cdot)$ is polynomial-time reducible to $Z_{\mathfrak{P}, \mathfrak{Q}}(\cdot)$.

7.1 Proof Sketch

We first give a proof sketch. Again, we will use interpolations and Vandermonde systems.

First, by Lemma 7, we only need to give a reduction from $Z_{\mathfrak{P}^*, \Omega^*}(\cdot)$ to $Z_{\mathfrak{P}, \mathfrak{R}}(\cdot)$, where

$$\mathfrak{R} = \left\{ \mathbf{C}^{[i]}, \mathbf{D}^{[i]} : i \in [g] \right\}$$

contains both $\mathbf{C}^{[i]}$ and its *normalized* version $\mathbf{D}^{[i]}$, $i \in [g]$.

Let $\mathcal{G} = (G, \mathcal{V}, \mathcal{E})$ be an input labeled graph of $Z_{\mathfrak{P}^*, \Omega^*}(\cdot)$, where $G = (V, E)$. For every assignment $\xi : V \rightarrow [r]$, we will define $\text{nw}(\xi) > 0$. Moreover, let X be the set of all possible values of $\text{nw}(\xi)$, and $L = |X|$, then L is polynomially bounded. For every $k \in [L]$, we will build a new labeled directed graph $\mathcal{G}^{[k]}$ from $\mathcal{G}$. $\mathcal{G}^{[k]}$ is a valid input graph of $Z_{\mathfrak{P}, \mathfrak{R}}(\cdot)$ (with domain $[m]$) and satisfies

$$Z_{\mathfrak{P}, \mathfrak{R}}(\mathcal{G}^{[k]}) = \sum_{\xi : V \rightarrow [r]} \text{wt}(\mathcal{G}, \xi) \cdot \left(\text{nw}(\xi) \right)^k. \quad (8)$$

For each $x \in X$, we use Ξ_x to denote the set of all $\xi : V \rightarrow [r]$ with $\text{nw}(\xi) = x$. Then by solving the Vandermonde system which consists of equations (8) for $k = 1, 2, \dots, L$, we can compute

$$\sum_{\xi \in \Xi_x} \text{wt}(\mathcal{G}, \xi), \quad \text{for every } x \in X,$$

which allow us to compute in polynomial time

$$Z_{\mathfrak{P}^*, \Omega^*}(\mathcal{G}) = \sum_{\xi : V \rightarrow [r]} \text{wt}(\mathcal{G}, \xi) = \sum_{x \in X} \left(\sum_{\xi \in \Xi_x} \text{wt}(\mathcal{G}, \xi) \right).$$

7.2 Construction of $\mathcal{G}^{[k]}$

We start with the construction of $\mathcal{G}^{[1]} = (G^{[1]}, \mathcal{V}^{[1]}, \mathcal{E}^{[1]})$. It will become clear that the construction can be generalized to get $\mathcal{G}^{[k]}$ for every $k \in [L]$.

Let $V = [n]$, then the vertex set $V^{[1]}$ of $G^{[1]} = (V^{[1]}, E^{[1]})$ will be defined as a union:

$$V^{[1]} = R_1 \cup R_2 \cup \dots \cup R_n,$$

where R_k corresponds to vertex $k \in V$ and any edge $uv \in E^{[1]}$ will be between two vertices $u, v \in V^{[1]}$ such that $u, v \in R_k$ for some unique $k \in [n]$. R_i and R_j , $i \neq j \in [n]$, are not necessarily disjoint and there could be vertices shared by (at most) two different sets R_i and R_j . We further divide the vertices of R_i , $i \in [n]$, into three types: In the subgraph of $G^{[1]}$ spanned by R_i ,

1. The Type-L vertices only have outgoing edges;
2. The Type-R vertices only have incoming edges; and
3. The Type-M vertices have both incoming and outgoing edges.

When adding a new vertex, we will also specify which type it is. The construction also guarantees that the underlying undirected graph spanned by every R_i is connected.

7.2.1 Construction of $G^{[1]} = (V^{[1]}, E^{[1]})$

We start with the vertex set $V^{[1]}$.

1. First, for every $i \in [n]$ and $a \in [g]$, we add a new Type-L vertex $u_{i,a}$ in R_i and add a new Type-R vertex $w_{i,a}$ in R_i . All these vertices appear in R_i only.
2. Second, for every $e = ij \in E$, where $i, j \in [n]$, we add a vertex $v_e \in R_i \cap R_j$, which is a Type-R vertex in R_i and a Type-L vertex in R_j .
3. Finally, for every $i \in V$ let $\mathbf{p} \in \mathfrak{P}^*$ be its vertex weight in $\mathcal{G}$. Then by the discussion earlier, it can be generated from $(\mathfrak{P}, \mathfrak{Q})$ using three finite sets of tuples $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$. For each tuple $\mathbf{s}$ in $\mathcal{S}_1$ we add a new Type-L vertex $v_{i,\mathbf{s}}$ in R_i ; for each tuple $\mathbf{s}$ in $\mathcal{S}_2$, we add a new Type-R vertex in R_i ; and for each tuple $\mathbf{s}$ in $\mathcal{S}_3$ we add a new Type-M vertex in R_i . All these vertices appear in R_i only.

We will add some more vertices later. Now we start to create edges, and assign edge/vertex weights.

First, for every $i \in [n]$, we add $2g$ edges to connect $u_{i,a}$ and $w_{i,a}$, $a \in [g]$:

1. For every $a \in [g]$, add one edge from $u_{i,a}$ to $w_{i,a}$, and label the edge with $\mathbf{C}^{[1]}$;
2. For every $a \in [g]$, add one edge from $u_{i,a}$ to $w_{i,a+1}$ (with $w_{i,g+1} = w_{i,1}$), and label it with $\mathbf{C}^{[1]}$;
3. For every $a \in [g]$, the vertex weight vector of both $u_{i,a}$ and $w_{i,a}$ is the all-one vector $\mathbf{1}$.

Second, for each edge $e = ij \in E$, we add the incident edges of $v_e \in R_i \cap R_j$ as follows. Assume the edge weight matrix of ij in $\mathcal{G}$ is generated by $(\mathfrak{P}, \mathfrak{Q})$ using the following $(2g+1)$ -tuple:

$$(k \in [h]; \mathbf{k} = (k_1, \dots, k_g); \boldsymbol{\ell} = (\ell_1, \dots, \ell_g)),$$

where $k_i, \ell_i \geq 0$, $\mathbf{k} \neq \mathbf{0}$ and $\boldsymbol{\ell} \neq \mathbf{0}$. Then we add the following incident edges of v_e :

1. For each $b \in [g]$, we add k_b parallel edges from $u_{i,b}$ to v_e in R_i , all of which are labeled with $\mathbf{C}^{[b]}$;
2. For each $b \in [g]$, we add ℓ_b parallel edges from v_e to $w_{j,b}$ in R_j , all of which are labeled with $\mathbf{C}^{[b]}$;
3. Assign the vertex weight vector $\mathbf{w}^{[k]} \in \mathfrak{P}$ to v_e .

Finally, for every vertex $i \in V$ we use $\mathbf{p}$ to denote its vertex weight in $\mathcal{G}$. Assume $\mathbf{p}$ is generated by $(\mathfrak{P}, \mathfrak{Q})$ using three finite sets $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$ of tuples. For each $\mathbf{s} = (k \in [h]; \mathbf{k} = (k_1, \dots, k_g))$ in $\mathcal{S}_1$ with $k_i \geq 0$ and $\mathbf{k} \neq \mathbf{0}$, we already added a Type-L vertex $v_{i,\mathbf{s}}$ in R_i (which appears in R_i only). We add the following incident edges of $v_{i,\mathbf{s}}$:

1. For each $b \in [g]$, add k_b parallel edges from $v_{i,\mathbf{s}}$ to $w_{i,b}$ in R_i , all of which are labeled with $\mathbf{C}^{[b]}$;
2. Assign the vertex weight vector $\mathbf{w}^{[k]} \in \mathfrak{P}$ to $v_{i,\mathbf{s}}$.

For every $\mathbf{s} = (k \in [h]; \mathbf{k} = (k_1, \dots, k_g))$ in $\mathcal{S}_2$, we already added a Type-R vertex $v_{i,\mathbf{s}} \in R_i$. We add the following incident edges of $v_{i,\mathbf{s}}$ in R_i :

1. For each $b \in [g]$, add k_b parallel edges from $u_{i,b}$ to $v_{i,\mathbf{s}}$ in R_i , all of which are labeled with $\mathbf{C}^{[b]}$;
2. Assign the vertex weight vector $\mathbf{w}^{[k]} \in \mathfrak{P}$ to $v_{i,\mathbf{s}}$.

For every tuple $\mathbf{s} = (k \in [h]; \mathbf{k} = (k_1, \dots, k_g); \boldsymbol{\ell} = (\ell_1, \dots, \ell_g))$ in $\mathcal{S}_3$, we already added a Type-M vertex $v_{i,\mathbf{s}}$ in R_i . We add the following incident edges of $v_{i,\mathbf{s}}$ in R_i :

1. For every $b \in [g]$, add k_b parallel edges from $u_{i,b}$ to $v_{i,s}$, all of which are labeled with $\mathbf{C}^{[b]}$;
2. For every $b \in [g]$, add ℓ_b parallel edges from $v_{i,s}$ to $w_{i,b}$, all of which are labeled with $\mathbf{C}^{[b]}$; and
3. Assign the vertex weight vector $\mathbf{w}^{[k]} \in \mathfrak{P}$ to $v_{i,s}$.

It can be checked that the (undirected) subgraph spanned by R_i , for all $i \in [n]$, is connected.

This almost finishes the construction. The only thing left is to add some more vertices and edges so that the out-degree of $u_{i,a}$ and the in-degree of $w_{i,a}$ are the same for all $i \in [n]$ and $a \in [g]$.

To this end, we notice that for all $i \in [n]$ and $a \in [g]$, both the out-degree of $u_{i,a}$ and the in-degree of $w_{i,a}$ constructed so far are linear in the maximum degree of G , because all the parameters k_i, ℓ_i and the sets $\mathcal{S}_i$ are considered as constants. As a result, we can pick a large enough positive integer $M \geq 2$ which is linear in the maximum degree of G , such that

$$M \geq \text{the out-degree of } u_{i,a} \text{ and the in-degree of } w_{i,a} \text{ constructed so far, for all } i \text{ and } a.$$

We now add vertices and edges so that the out-degree of $u_{i,a}$ and the in-degree of $w_{i,a}$ all become M .

Let $i \in [n]$ and $a \in [g]$. Assume the current out-degree of $u_{i,a}$ is $k \leq M$. Then we add $M - k$ new Type-R vertices in R_i and add one edge from $u_{i,a}$ to each of these vertices. The vertex weights of all the new vertices are $\mathbf{1}$, and the edge weights of all the new edges are $\mathbf{D}^{[a]}$ (recall that we are allowed to use the normalized version $\mathbf{D}^{[a]}$ of $\mathbf{C}^{[a]}$, and this is actually the only place we use it).

Similarly, assume the current in-degree of $w_{i,a}$ is $k \leq M$. Then we add $M - k$ new Type-L vertices in R_i and add one edge from each of these vertices to $w_{i,a}$. The vertex weights of all the new vertices are $\mathbf{1}$ while the edge weights of all the new edges are $\mathbf{C}^{[a]}$.

This finishes the construction of the new labeled directed graph $\mathcal{G}^{[1]} = (G^{[1]}, \mathcal{V}^{[1]}, \mathcal{E}^{[1]})$.

7.3 Proof of Equation (8)

We start with the definition of $\text{nw}(\xi)$, for any assignment $\xi : V = [n] \rightarrow [r]$.

First, for each $a \in [g]$, we let $\boldsymbol{\mu}^{[a]}$ denote the following positive r -dimensional vector:

$$\mu_i^{[a]} = \sum_{x \in A_i} \left(\alpha_x^{[1]} \right)^2 \cdot \left(\alpha_x^{[a]} \right)^{M-2}, \quad \text{for every } i \in [r].$$

For every $a \in [g]$, we let $\boldsymbol{\nu}^{[a]}$ denote the following positive r -dimensional vector:

$$\nu_i^{[a]} = \sum_{x \in B_i} \left(\beta_x^{[1]} \right)^2 \cdot \left(\beta_x^{[a]} \right)^{M-2}, \quad \text{for every } i \in [r].$$

Finally, we define $\text{nw}(\xi)$ as follows:

$$\text{nw}(\xi) = \prod_{i \in [n]} \prod_{a \in [g]} \mu_{\xi(i)}^{[a]} \cdot \nu_{\xi(i)}^{[a]}, \quad \text{for any } \xi : V = [n] \rightarrow [r].$$

It is easy to check that $\text{nw}(\xi) > 0$ and the number of possible values of $\text{nw}(\xi)$ is polynomial in n .

Now we prove equation (8) for $k = 1$:

$$Z_{\mathfrak{P}, \mathfrak{R}}(\mathcal{G}^{[1]}) = \sum_{\xi : V \rightarrow [r]} \text{wt}(\mathcal{G}, \xi) \cdot \text{nw}(\xi). \quad (9)$$

Let ξ be an assignment from V to $[r]$. We use Φ_ξ to denote the set of all assignments $\phi : V^{[1]} \rightarrow [m]$ such that for every edge uv in the subgraph spanned by R_i , $i \in [n]$, we have

$$\phi(u) \in A_{\xi(i)} \quad \text{and} \quad \phi(v) \in B_{\xi(i)}.$$

In other words, for all $i \in [n]$ and $v \in R_i$, if v is a Type-L vertex then $\phi(v) \in A_{\xi(i)}$; if v is a Type-R vertex then $\phi(v) \in B_{\xi(i)}$; and if v is a Type-M of R_i , then $\phi(v) \in A_{\xi(i)} \cap B_{\xi(i)}$. Equivalently, we can associate every vertex $v \in V^{[1]}$ with a subset $U_v \subseteq [m]$, where

1. If v appears in both R_i and R_j for some $i \neq j \in V = [n]$, and v is Type-R in R_i and Type-L in R_j , then $U_v = B_{\xi(i)} \cap A_{\xi(j)}$;
2. Otherwise, assume v only appears in R_i for some $i \in V = [n]$. Then
 - (a) If v is Type-L, then $U_v = A_{\xi(i)}$;
 - (b) If v is Type-R, then $U_v = B_{\xi(i)}$; and
 - (c) If v is Type-M, then $U_v = B_{\xi(i)} \cap A_{\xi(i)}$,

such that $\phi \in \Phi_\xi$ if and only if $\phi(v) \in U_v$ for all $v \in V^{[1]}$. In particular, $\Phi_\xi = \emptyset$ iff $U_v = \emptyset$ for some v .

By the construction, we know the subgraph spanned by R_i is *connected*, for any $i \in [n]$. It implies that $\text{wt}(\mathcal{G}^{[1]}, \phi) \neq 0$ only if $\phi \in \Phi_\xi$ for a unique $\xi : V \rightarrow [r]$. As a result, we have

$$Z_{\mathfrak{P}, \mathfrak{R}}(\mathcal{G}^{[1]}) = \sum_{\phi} \text{wt}(\mathcal{G}^{[1]}, \phi) = \sum_{\xi} \sum_{\phi \in \Phi_\xi} \text{wt}(\mathcal{G}^{[1]}, \phi),$$

and to prove (9) we only need to show that

$$\sum_{\phi \in \Phi_\xi} \text{wt}(\mathcal{G}^{[1]}, \phi) = \text{wt}(\mathcal{G}, \xi) \cdot \text{nvw}(\xi), \quad \text{for any assignment } \xi : V = [n] \rightarrow [r].$$

We use $\mathbf{w}^{[v]}$ to denote the weight vector of $v \in V^{[1]}$, E_i to denote the set of edges in $E^{[1]}$ labeled with $\mathbf{C}^{[i]}$, and F_i to denote the set of edges in $E^{[1]}$ labeled with $\mathbf{D}^{[i]}$, then we have

$$\sum_{\phi \in \Phi_\xi} \text{wt}(\mathcal{G}^{[1]}, \phi) = \sum_{\phi \in \Phi_\xi} \left(\prod_{v \in V^{[1]}} \mathbf{w}_{\phi(v)}^{[v]} \prod_{i \in [g]} \left(\prod_{uv \in E_i} C_{\phi(u), \phi(v)}^{[i]} \right) \left(\prod_{uv \in F_i} D_{\phi(u), \phi(v)}^{[i]} \right) \right).$$

By the definition of Φ_ξ , if $\Phi_\xi \neq \emptyset$, then every $\phi \in \Phi_\xi$ satisfies

$$C_{\phi(u), \phi(v)}^{[i]} = \alpha_{\phi(u)}^{[i]} \cdot \beta_{\phi(v)}^{[i]} \quad \text{and} \quad D_{\phi(u), \phi(v)}^{[i]} = \alpha_{\phi(u)}^{[i]} \cdot \delta_{\phi(v)}^{[i]},$$

where $(\alpha^{[i]}, \delta^{[i]})$ is the representation of $\mathbf{D}^{[i]}$. As a result, we have

$$\sum_{\phi \in \Phi_\xi} \text{wt}(\mathcal{G}^{[1]}, \phi) = \sum_{\phi \in \Phi_\xi} \left(\prod_{v \in V^{[1]}} \mathbf{w}_{\phi(v)}^{[v]} \prod_{i \in [g]} \left(\prod_{uv \in E_i} \alpha_{\phi(u)}^{[i]} \cdot \beta_{\phi(v)}^{[i]} \right) \left(\prod_{uv \in F_i} \alpha_{\phi(u)}^{[i]} \cdot \delta_{\phi(v)}^{[i]} \right) \right).$$

Because $\phi \in \Phi_\xi$ iff $\phi(v) \in U_v$ for all v , we can express this sum of products as a product of sums:

$$\prod_{v \in V^{[1]}} H_v,$$

in which every H_v , $v \in V^{[1]}$, is a sum over $\phi(v) \in U_v$.

Finally, we show the following equation:

$$\prod_{v \in V^{[1]}} H_v = \text{wt}(\mathcal{G}, \xi) \cdot \text{nvw}(\xi). \tag{10}$$

This follows from the construction of $\mathcal{G}^{[1]}$ and the following observations:

1. For each $v_e \in R_i \cap R_j$, which is added because of edge $ij \in E$, it can be checked that the sum H_{v_e} over $U_{v_e} = B_{\xi(i)} \cap A_{\xi(j)}$ is exactly $F_{\xi(i), \xi(j)}$, where $\mathbf{F}$ is the weight of ij in $\mathcal{G}$ (as defined in (7)).
2. Let $\mathbf{p}$ denote the vertex weight of $i \in V$, which is generated using $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$. Then we have

$$p_{\xi(i)} = \prod_{s \in \mathcal{S}_1} H_{v_{i,s}} \prod_{s \in \mathcal{S}_2} H_{v_{i,s}} \prod_{s \in \mathcal{S}_3} H_{v_{i,s}}.$$

3. For all $i \in [n]$ and $a \in [g]$, we have

$$\mu_{\xi(i)}^{[a]} = H_{u_{i,a}} \quad \text{and} \quad \nu_{\xi(i)}^{[a]} = H_{w_{i,a}}.$$

4. Finally, it can be checked that $H_v = 1$ for all other vertices in $V^{[1]}$, which is the reason we need to use the normalized matrices $\mathbf{D}^{[a]}$ in the construction.

7.3.1 Construction of $\mathcal{G}^{[k]}$

We can similarly construct $\mathcal{G}^{[k]}$ for every $k \in [L]$.

The only difference is that, instead of $u_{i,a}$ and $w_{i,a}$, we add the following $2kg$ vertices in R_i :

$$u_{i,j,a} \quad \text{and} \quad w_{i,j,a}, \quad \text{for all } j \in [k] \text{ and } a \in [g].$$

We also connect these vertices by adding $4kg$ edges, whose underlying undirected graph is a cycle. All these edges are labeled with $\mathbf{C}^{[1]}$. We also add extra vertices and edges so that the out-degree of $u_{i,j,a}$ and the in-degree of $w_{i,j,a}$ are M for all $i \in [n]$, $j \in [k]$ and $a \in [g]$. It then can be proved similarly that

$$Z_{\mathfrak{P}, \mathfrak{M}}(\mathcal{G}^{[k]}) = \sum_{\xi: V \rightarrow [r]} \text{wt}(\mathcal{G}, \xi) \cdot \left(\text{nvw}(\xi) \right)^k.$$

This completes the proof of Lemma 1.

8 Decidability

In this section, we prove Lemma 3 (restated below) and show that the rank condition is decidable.

Lemma 3. *Let $(\mathfrak{X}_0, \mathfrak{Y}_0)$ be a finite $\mathcal{T}_0$ -pair with $\mathfrak{X}_0 = \{\mathbf{1}\}$. Assuming that $\mathcal{T}_0, \dots, \mathcal{T}_{\ell-1}, \mathcal{T}_\ell$ and $(\mathfrak{X}_0, \mathfrak{Y}_0), \dots, (\mathfrak{X}_{\ell-1}, \mathfrak{Y}_{\ell-1})$ all satisfy their conditions in **(R)** for some $\ell \in [h]$, the rank property for $\mathfrak{Y}_\ell$ can be checked in a finite number of steps.*

Let $\mathcal{T}_\ell = \{(A_1, B_1), \dots, (A_r, B_r)\}$. To check the rank property for $\mathfrak{Y}_\ell$ (i.e., matrices $\mathbf{D} \in \mathfrak{Y}_\ell$ are block-rank-1), it suffices to check whether every $\mathbf{D} \in \mathfrak{Y}_\ell$ satisfies that

$$D_{i,j} \cdot D_{i',j'} - D_{i,j'} \cdot D_{i',j} = 0,$$

for all $k \in [r]$, $i, i' \in A_k$ and $j, j' \in B_k$. In Section 8.2, we introduce the notion of *matrix polynomials* and say that $\mathfrak{Y}_\ell$ *satisfies* f if f is a polynomial over variables

$$\left\{ x_{i,j} : i \in A_k \text{ and } j \in B_k \text{ for some } k \in [r] \right\}$$

and evaluates to 0 when $x_{i,j}$ is assigned $D_{i,j}$, for all $\mathbf{D} \in \mathfrak{Y}_\ell$. Thus, $\mathfrak{Y}_\ell$ satisfies the rank property if and only if it satisfies all polynomials $f_{i,i',j,j'}$ of the following form:

$$x_{i,j} \cdot x_{i',j'} - x_{i,j'} \cdot x_{i',j},$$

where $i, i' \in A_k$ and $j, j' \in B_k$ for some $k \in [r]$. The main component of the proof (Section 8.3) shows that to check whether $\mathfrak{Y}_\ell$ satisfies a polynomial f or not, it suffices to check a finite number of polynomials over $\mathfrak{Y}_{\ell-1}$ and a finite number of polynomials over $\mathfrak{X}_{\ell-1}$ (see Section 8.2 for a similar definition of vector polynomials applied over vectors in $\mathfrak{X}_{\ell-1}$); a similar reduction also holds for vector polynomials. This is used in Section 8.4 to show that, to check the rank property for $\mathfrak{Y}_\ell$, it suffices to check a finite number of polynomials on $\mathfrak{X}_0$ and $\mathfrak{Y}_0$, which can be done in a finite number of steps since both of them are finite.

We start the proof with a technical lemma.

8.1 A Technical Lemma

Lemma 9. *Let L, n and m be three positive integers. For each $i \in [L]$, let $a_1^{[i]}, \dots, a_n^{[i]}$ be a sequence of n positive numbers and $b_1^{[i]}, \dots, b_m^{[i]}$ be a sequence of m positive numbers. If*

$$\sum_{i \in [n]} \prod_{j \in [L]} \left(a_i^{[j]}\right)^{k_j} = \sum_{i \in [m]} \prod_{j \in [L]} \left(b_i^{[j]}\right)^{k_j}, \quad \text{for all } k_1, k_2, \dots, k_L \geq 1,$$

then we must have $m = n$ and there exists a permutation π of $[n]$ such that

$$a_i^{[j]} = b_{\pi(i)}^{[j]}, \quad \text{for all } i \in [n] \text{ and } j \in [L].$$

Proof. We prove it by induction on L . The base case when $L = 1$ is trivial.

Assume the lemma is true for $L - 1 \geq 1$. Without loss of generality, we assume that $\{a_1^{[L]}, \dots, a_n^{[L]}\}$ and $\{b_1^{[L]}, \dots, b_m^{[L]}\}$ are already sorted:

$$a_1^{[L]} \geq \dots \geq a_n^{[L]} > 0 \quad \text{and} \quad b_1^{[L]} \geq \dots \geq b_m^{[L]} > 0.$$

We let $s \geq 1$ and $t \geq 1$ be the two maximum integers such that

$$a_1^{[L]} = a_2^{[L]} = \dots = a_s^{[L]} = a > 0 \quad \text{and} \quad b_1^{[L]} = b_2^{[L]} = \dots = b_t^{[L]} = b > 0.$$

First it is easy to show that $a = b$. Otherwise assume $a > b$, then we set $k_1 = \dots = k_{L-1} = 1$, divide both sides by $(a)^{k_L}$, and let k_L go to infinity. It is easy to check that the left side converges to

$$\sum_{i \in [s]} \prod_{j \in [L-1]} a_i^{[j]} > 0,$$

while the right side converges to 0, which contradicts the assumption.

Second, we fix $k_1, \dots, k_{L-1}$ to be any positive integers, divide both sides by $(a)^{k_L} = (b)^{k_L}$ and let k_L go to infinity. It is easy to check that the left side converges to

$$\sum_{i \in [s]} \prod_{j \in [L-1]} \left(a_i^{[j]}\right)^{k_j},$$

while the right hand side converges to

$$\sum_{i \in [t]} \prod_{j \in [L-1]} \left(b_i^{[j]}\right)^{k_j}.$$

So these two sums are equal for all $k_1, \dots, k_{L-1} \geq 1$. Then we apply the inductive hypothesis to claim that $s = t$ and there exists a permutation π from $[s]$ to itself such that

$$a_i^{[j]} = b_{\pi(i)}^{[j]}, \quad \text{for all } j \in [L-1] \text{ and } i \in [s]. \quad (11)$$

It is also easy to see that for any $i \in [s]$, (11) also holds for $j = L$.

We then repeat the whole process after removing the first s elements from the $2L$ sequences. $\square$

Additionally, we also need the following simple lemma in the proof.

Lemma 10. *Let $m \geq 1$ be an integer and $(P_1, P_2, \dots)$ be a sequence of subsets of $[m]$. If for any finite subset $\{i_1, \dots, i_k\} \subset \mathbb{N}$, $P_{i_1} \cap P_{i_2} \cap \dots \cap P_{i_k} \neq \emptyset$, then there exists a $j \in [m]$ such that $j \in P_i$ for all i .*

Proof. If for every $j \in [m]$, there exists some $i_j \geq 1$ such that $j \notin P_{i_j}$, then the finite intersection

$$\bigcap_{j=1}^m P_{i_j} = \emptyset,$$

which contradicts the assumption. $\square$

8.2 Matrix and Vector Polynomials

Let $(\mathfrak{X}, \mathfrak{Y})$ be a generalized $\mathcal{P}$ -pair, for some $m \times m$ pattern $\mathcal{P}$. So every vector $\mathbf{w} \in \mathfrak{X}$ is either *positive* or *$\mathcal{P}$ -weakly positive* and every $\mathbf{D} \in \mathfrak{Y}$ is either a *$\mathcal{P}$ -matrix* or a *$\mathcal{P}$ -diagonal matrix*. Note that if $\mathfrak{Y}$ only has $\mathcal{P}$ -matrices, then $(\mathfrak{X}, \mathfrak{Y})$ is a $\mathcal{P}$ -pair. The definitions below also apply to $\mathcal{P}$ -pairs.

We say f is a *$\mathcal{P}$ -matrix polynomial* if f is a polynomial over variables

$$\{x_{i,j} : (i,j) \in \mathcal{P}\}$$

with integer coefficients and zero constant term. We say $\mathfrak{Y}$ satisfies f if for every $\mathcal{P}$ -matrix $\mathbf{D} \in \mathfrak{Y}$, we have $f(\mathbf{D}) = 0$, in which we substitute $x_{i,j}$ by $D_{i,j} > 0$ for all $(i,j) \in \mathcal{P}$. We also say $(\mathfrak{X}, \mathfrak{Y})$ satisfies f if $\mathfrak{Y}$ satisfies f .

We say f is a *$\mathcal{P}$ -diagonal matrix polynomial* if f is a polynomial over variables

$$\{x_i : (i,i) \in \mathcal{P}\}$$

with integer coefficients and zero constant term. We say $\mathfrak{Y}$ satisfies f if every $\mathcal{P}$ -diagonal matrix $\mathbf{D} \in \mathfrak{Y}$ satisfies $f(\mathbf{D}) = 0$. We also say $(\mathfrak{X}, \mathfrak{Y})$ satisfies f if $\mathfrak{Y}$ satisfies f .

We say g is an *m -vector polynomial* if g is a polynomial over variables

$$\{y_i : i \in [m]\}$$

with integer coefficients and zero constant term. Similarly, we say $\mathfrak{X}$ satisfies g if every positive vector $\mathbf{w} \in \mathfrak{X}$ satisfies $g(\mathbf{w}) = 0$. We also say $(\mathfrak{X}, \mathfrak{Y})$ satisfies g if $\mathfrak{X}$ satisfies g .

Finally, we say g is a *$\mathcal{P}$ -weakly positive vector polynomial* if g is a polynomial over variables

$$\{y_i : (i,i) \in \mathcal{P}\}$$

with integer coefficients and zero constant term. We say $\mathfrak{X}$ satisfies g if every $\mathcal{P}$ -weakly positive vector $\mathbf{w} \in \mathfrak{X}$ satisfies $g(\mathbf{w}) = 0$. We also say $(\mathfrak{X}, \mathfrak{Y})$ satisfies g if $\mathfrak{X}$ satisfies g .

Let F be a finite set of $\mathcal{P}$ -matrix, $\mathcal{P}$ -diagonal matrix, m -vector, and $\mathcal{P}$ -weakly positive vector polynomials. Then we say $(\mathfrak{X}, \mathfrak{Y})$ satisfies F if $(\mathfrak{X}, \mathfrak{Y})$ satisfies every polynomial $f \in F$.

Similarly, given any block pattern $\mathcal{T}$, we can define $\mathcal{T}$ -matrix polynomials, $\mathcal{T}$ -diagonal matrix polynomials, and $\mathcal{T}$ -weakly positive vector polynomials for $\mathcal{T}$ -pairs and *generalized $\mathcal{T}$ -pairs*.

As discussed at the beginning of the section, when $(\mathfrak{X}, \mathfrak{Y})$ is a $\mathcal{T}$ -pair, to check whether $\mathfrak{Y}$ satisfies the rank condition (i.e., every matrix $\mathbf{D} \in \mathfrak{Y}$ is block-rank-1), one only needs to check whether $\mathfrak{Y}$ satisfies all the $\mathcal{T}$ -matrix polynomials $f_{i,i',j,j'}$ of the following form

$$f_{i,i',j,j'}(\mathbf{x}) = x_{i,j} \cdot x_{i',j'} - x_{i,j'} \cdot x_{i',j}, \quad \text{where } i, i' \in A_k \text{ and } j, j' \in B_k \text{ for some } k \in [r].$$

8.3 Checking Matrix and Vector Polynomials

Now let $(\mathfrak{X}, \mathfrak{Y})$ be a $\mathcal{T}$ -pair for some non-trivial $m \times m$ block pattern $\mathcal{T} = \{(A_1, B_1), \dots, (A_r, B_r)\}$ with $r \geq 1$. We also assume that every matrix in $\mathfrak{Y}$ is block-rank-1, and $\mathfrak{X}$ is *closed*.

We can apply the **gen-pair** operation to get a new $\mathcal{P}$ -pair

$$(\mathfrak{X}', \mathfrak{Y}') = \text{gen-pair}(\mathfrak{X}, \mathfrak{Y}), \quad \text{where } \mathcal{P} = \text{gen}(\mathcal{T}).$$

We also let $(\mathfrak{X}^*, \mathfrak{Y}^*)$ denote the *generalized* $\mathcal{P}$ -pair defined in Section 4. By definition, $\mathfrak{X}^*$ is also closed.

In this section, we first show that to check whether $(\mathfrak{X}^*, \mathfrak{Y}^*)$ satisfies a matrix or vector polynomial, one only needs to check finitely many polynomials for $(\mathfrak{X}, \mathfrak{Y})$. One can prove a similar relation between $(\mathfrak{X}', \mathfrak{Y}')$ and $(\mathfrak{X}^*, \mathfrak{Y}^*)$. As a result, to check whether $(\mathfrak{X}', \mathfrak{Y}')$ satisfies a polynomial or not, we only need to check finitely many polynomials for $(\mathfrak{X}, \mathfrak{Y})$.

We start with the following lemma.

Lemma 11. *Let f be a $\mathcal{P}$ -matrix or $\mathcal{P}$ -diagonal matrix polynomial. Then one can construct a finite set $\{F_1, \dots, F_L\}$ in a finite number of steps, in which every F_i , $i \in [L]$, is a finite set of $\mathcal{T}$ -matrix, m -vector, and $\mathcal{T}$ -weakly positive vector polynomials, such that*

$$(\mathfrak{X}^*, \mathfrak{Y}^*) \text{ satisfies } f \iff \exists i \in [L] \text{ such that } (\mathfrak{X}, \mathfrak{Y}) \text{ satisfies } F_i.$$

Proof. We first prove the case when f is a $\mathcal{P}$ -matrix polynomial.

If f is the zero polynomial, then the lemma follows by setting $L = 1$ and F_1 to be the set consists of the zero polynomial only. From now on we assume that f is not the zero polynomial.

Let $\{\mathbf{C}^{[1]}, \dots, \mathbf{C}^{[s]}\}$ and $\{\mathbf{D}^{[1]}, \dots, \mathbf{D}^{[t]}\}$ be two finite subsets of $\mathcal{T}$ -matrices in $\mathfrak{Y}$ and $\{\mathbf{w}^{[1]}, \dots, \mathbf{w}^{[h]}\}$ be a finite subset of *positive* vectors in $\mathfrak{X}$, where $s, t, h \geq 1$. We also let $(\boldsymbol{\alpha}^{[i]}, \boldsymbol{\beta}^{[i]})$ and $(\boldsymbol{\gamma}^{[i]}, \boldsymbol{\delta}^{[i]})$ denote the representations of $\mathbf{C}^{[i]}$ and $\mathbf{D}^{[i]}$, respectively. By the definition of $\mathfrak{Y}^*$ and the assumption that $\mathfrak{Y}$ is *closed*, we can construct from every $(s + t + h)$ -tuple

$$\mathbf{p} = (k_1, \dots, k_s, \ell_1, \dots, \ell_t, e_1, \dots, e_h), \quad \text{where } k_i, \ell_i, e_i \geq 1,$$

the following $\mathcal{P}$ -matrix $\mathbf{C}^{[\mathbf{p}]}$ in $\mathfrak{Y}^*$: the (i, j) th entry of $\mathbf{C}^{[\mathbf{p}]}$ is

$$\sum_{y \in B_i \cap A_j} \left(\beta_y^{[1]} \right)^{k_1} \cdots \left(\beta_y^{[s]} \right)^{k_s} \cdot \left(\gamma_y^{[1]} \right)^{\ell_1} \cdots \left(\gamma_y^{[t]} \right)^{\ell_t} \cdot \left(w_y^{[1]} \right)^{e_1} \cdots \left(w_y^{[h]} \right)^{e_h}, \quad \text{for all } i, j \in [r]. \quad (12)$$

This follows from the fact that the Hadamard product of $(\mathbf{w}^{[1]})^{e_1}, \dots, (\mathbf{w}^{[h]})^{e_h}$ is actually a vector in $\mathfrak{X}$, because $\mathfrak{X}$ is known to be *closed*.

Now we assume $(\mathfrak{X}^*, \mathfrak{Y}^*)$ satisfies f , then by definition we must have

$$f(\mathbf{C}^{[\mathbf{p}]}) = 0, \quad \text{for all } \mathbf{p}, \quad (13)$$

since $\mathbf{C}^{[\mathbf{p}]}$ is a $\mathcal{P}$ -matrix in $\mathfrak{Y}^*$. By combining (13) and (12) and *rearranging* terms, we have

$$\begin{aligned} & \sum_{i \in [n_1]} \left(\prod_{j \in [s]} \left(f_i(\beta_1^{[j]}, \dots, \beta_m^{[j]}) \right)^{k_j} \right) \left(\prod_{j \in [t]} \left(f_i(\gamma_1^{[j]}, \dots, \gamma_m^{[j]}) \right)^{\ell_j} \right) \left(\prod_{j \in [h]} \left(f_i(w_1^{[j]}, \dots, w_m^{[j]}) \right)^{e_j} \right) \\ &= \sum_{i \in [n_2]} \left(\prod_{j \in [s]} \left(g_i(\beta_1^{[j]}, \dots, \beta_m^{[j]}) \right)^{k_j} \right) \left(\prod_{j \in [t]} \left(g_i(\gamma_1^{[j]}, \dots, \gamma_m^{[j]}) \right)^{\ell_j} \right) \left(\prod_{j \in [h]} \left(g_i(w_1^{[j]}, \dots, w_m^{[j]}) \right)^{e_j} \right) \end{aligned}$$

for all $\mathbf{p}$. In the equation above, n_1 and n_2 are two non-negative integers. For all $i \in [n_1]$ and $j \in [n_2]$, both $f_i(x_1, \dots, x_m)$ and $g_j(x_1, \dots, x_m)$ are monomials in $x_1, \dots, x_m$. Also note that all the monomials f_i, g_j only depend on the $\mathcal{P}$ -matrix polynomial f but do not depend on the choices of $\mathbf{p}$ and the subsets $\{\mathbf{C}^{[1]}, \dots, \mathbf{C}^{[s]}\}$, $\{\mathbf{D}^{[1]}, \dots, \mathbf{D}^{[t]}\}$, and $\{\mathbf{w}^{[1]}, \dots, \mathbf{w}^{[h]}\}$. Moreover, because we assumed that f is not the zero polynomial, at least one of n_1 and n_2 is nonzero.

It follows directly from Lemma 9 that if $(\mathfrak{X}^*, \mathfrak{Y}^*)$ satisfies f , then we must have $n_1 = n_2$ which we denote by n . (If $n_1 \neq n_2$, then we already know that $f(\mathbf{C}^{[\mathbf{p}]}) = 0$ cannot hold for all $\mathbf{p}$. The lemma then follows by setting $L = 1$ and F_1 to be the set that consists of the following m -vector polynomial: $g(\mathbf{x}) = x_1$ so that $(\mathfrak{X}, \mathfrak{Y})$ does not satisfy F_1 .) Moreover, by Lemma 9, if $(\mathfrak{X}^*, \mathfrak{Y}^*)$ satisfies f then there also exists a permutation π from $[n]$ to itself such that

$$\begin{aligned} f_i(\beta_1^{[j]}, \dots, \beta_m^{[j]}) &= g_{\pi(i)}(\beta_1^{[j]}, \dots, \beta_m^{[j]}), & \text{for all } j \in [s] \text{ and } i \in [n]; \\ f_i(\gamma_1^{[j]}, \dots, \gamma_m^{[j]}) &= g_{\pi(i)}(\gamma_1^{[j]}, \dots, \gamma_m^{[j]}), & \text{for all } j \in [t] \text{ and } i \in [n]; \text{ and} \\ f_i(w_1^{[j]}, \dots, w_m^{[j]}) &= g_{\pi(i)}(w_1^{[j]}, \dots, w_m^{[j]}), & \text{for all } j \in [h] \text{ and } i \in [n]. \end{aligned}$$

Since all the discussion above and all the monomials f_i, g_i do not depend on the choice of the three subsets, we can apply Lemma 10 to claim that if $(\mathfrak{X}^*, \mathfrak{Y}^*)$ satisfies f , then there must exist a (universal) permutation π from $[n]$ to itself such that for all $\mathbf{D} \in \mathfrak{X}$ (since $(\mathfrak{X}, \mathfrak{Y})$ is a $\mathcal{T}$ -pair, $\mathbf{D}$ is a $\mathcal{T}$ -matrix),

$$\begin{aligned} f_i(\alpha_1, \dots, \alpha_m) - g_{\pi(i)}(\alpha_1, \dots, \alpha_m) &= 0, & \text{for all } i \in [n] \text{ and} \\ f_i(\beta_1, \dots, \beta_m) - g_{\pi(i)}(\beta_1, \dots, \beta_m) &= 0, & \text{for all } i \in [n], \end{aligned}$$

where $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the representation of $\mathbf{D}$; and for every positive vector $\mathbf{w} \in \mathfrak{Y}$,

$$f_i(w_1, \dots, w_m) - g_{\pi(i)}(w_1, \dots, w_m) = 0, \quad \text{for all } i \in [n].$$

It is also easy to check that these conditions are sufficient.

Furthermore, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can be expressed by the positive entries of $\mathbf{D}$ as follows. For every $i \in A_k$, where $k \in [r]$, let d be the smallest index in B_k , then we have

$$\alpha_i = \frac{D_{i,d}}{\sum_{j \in A_k} D_{j,d}}.$$

For every $i \in B_k$, where $k \in [r]$, let d be the smallest index in A_k , then $\beta_i = D_{d,i}/\alpha_d$. Now it is easy to see that for every permutation π from $[n]$ to itself, we can construct a finite set F_π of $\mathcal{T}$ -matrix and m -vector polynomials, such that, if $(\mathfrak{X}^*, \mathfrak{Y}^*)$ satisfies f then $(\mathfrak{X}, \mathfrak{Y})$ satisfies F_π for some π .

The case when f is a $\mathcal{P}$ -diagonal matrix polynomial can be proved similarly. The only difference is that every F_π is now a finite set of $\mathcal{T}$ -matrix and $\mathcal{T}$ -weakly positive vector polynomials. $\square$

It also follows directly by definition that $\mathfrak{Y}'$ satisfies a $\mathcal{P}$ -matrix polynomial if and only if $\mathfrak{Y}^*$ satisfies the same polynomial, because $\mathfrak{Y}'$ contains precisely all the $\mathcal{P}$ -matrices in $\mathfrak{Y}^*$. Next, we deal with vector polynomials.

Lemma 12. *Let g be an r -vector or a $\mathcal{P}$ -weakly positive vector polynomial. One can construct a finite set $\{G_1, \dots, G_L\}$ in a finite number of steps, in which every G_i is a finite set of $\mathcal{T}$ -matrix, m -vector, and $\mathcal{T}$ -weakly positive vector polynomials, such that*

$$(\mathfrak{X}^*, \mathfrak{Y}^*) \text{ satisfies } g \iff \exists i \in [L] \text{ such that } (\mathfrak{X}, \mathfrak{Y}) \text{ satisfies } G_i.$$

Proof. We only prove the case when g is $\mathcal{P}$ -weakly positive. The other case can be proved similarly.

Again, we assume that g is not the zero polynomial.

Recall that when defining $\mathfrak{X}^*$ in Section 4, we first define $\mathfrak{X}^\#$ and $\mathfrak{X}^*$ is then the closure of $\mathfrak{X}^\#$: $\mathbf{w}$ is a $\mathcal{P}$ -weakly positive vector in $\mathfrak{X}^*$ if and only if there exist a finite and possibly empty subset of positive vectors $\{\mathbf{w}^{[1]}, \dots, \mathbf{w}^{[s]}\} \subseteq \mathfrak{X}^\#$ for some $s \geq 0$, a finite and nonempty subset of $\mathcal{P}$ -weakly positive vectors $\{\mathbf{u}^{[1]}, \dots, \mathbf{u}^{[t]}\} \subseteq \mathfrak{X}^\#$ for some $t \geq 1$, and positive integers $k_1, \dots, k_s, \ell_1, \dots, \ell_t$, such that

$$\mathbf{w} = (\mathbf{w}^{[1]})^{k_1} \circ \dots \circ (\mathbf{w}^{[s]})^{k_s} \circ (\mathbf{u}^{[1]})^{\ell_1} \circ \dots \circ (\mathbf{u}^{[t]})^{\ell_t}.$$

To prove Lemma 12, we first construct a finite set $\{F_1, \dots, F_M\}$, in which every F_i is a finite set of r -vector and $\mathcal{P}$ -weakly positive vector polynomials, such that

$$\mathfrak{X}^* \text{ satisfies } g \iff \exists i \in [M] \text{ such that } \mathfrak{X}^\# \text{ satisfies } F_i. \quad (14)$$

To this end, we let $\{\mathbf{w}^{[1]}, \dots, \mathbf{w}^{[s]}\}$ be a finite subset of positive vectors in $\mathfrak{X}^\#$; and $\{\mathbf{u}^{[1]}, \dots, \mathbf{u}^{[t]}\}$ be a finite subset of $\mathcal{P}$ -weakly positive vectors in $\mathfrak{X}^\#$, with $s \geq 0$ and $t \geq 1$. Then from any tuple

$$\mathbf{p} = (k_1, \dots, k_s, \ell_1, \dots, \ell_t), \quad \text{where } k_i, \ell_i \geq 1,$$

we get a $\mathcal{P}$ -weakly positive vector $\mathbf{w}^{[\mathbf{p}]} \in \mathfrak{X}^*$, where

$$\mathbf{w}^{[\mathbf{p}]} = (\mathbf{w}^{[1]})^{k_1} \circ \dots \circ (\mathbf{w}^{[s]})^{k_s} \circ (\mathbf{u}^{[1]})^{\ell_1} \circ \dots \circ (\mathbf{u}^{[t]})^{\ell_t}.$$

Assume $\mathfrak{X}^*$ satisfies g , then we have $g(\mathbf{w}^{[\mathbf{p}]}) = 0$ for all $\mathbf{p}$. Combining these two equations, we have

$$\sum_{i \in [n_1]} \left(\prod_{j \in [s]} (f_i(\mathbf{w}^{[j]}))^{k_j} \right) \left(\prod_{j \in [t]} (f_i(\mathbf{u}^{[j]}))^{l_j} \right) = \sum_{i \in [n_2]} \left(\prod_{j \in [s]} (g_i(\mathbf{w}^{[j]}))^{k_j} \right) \left(\prod_{j \in [t]} (g_i(\mathbf{u}^{[j]}))^{l_j} \right)$$

for all $\mathbf{p}$. In the equation, $f_i(\mathbf{x})$ and $g_i(\mathbf{x})$ are both monomials over x_i , $(i, i) \in \mathcal{P}$. Again, f_i and g_i only depend on the polynomial g but do not depend on the choices of $\mathbf{p}$ and the two subsets $\{\mathbf{w}^{[1]}, \dots, \mathbf{w}^{[s]}\}$ and $\{\mathbf{u}^{[1]}, \dots, \mathbf{u}^{[t]}\}$.

Because g is not the zero polynomial, one of n_1 and n_2 must be positive, and we have the following two cases. If $n_1 \neq n_2$, then by Lemma 9, $\mathfrak{X}^*$ cannot satisfy g and (14) follows by setting $L = 1$ and F_1 to be the set consists of the following r -vector polynomial: $f(\mathbf{x}) = x_1$.

Otherwise, we have $n_1 = n_2 > 0$, which we denote by n . It follows from Lemma 9 and Lemma 10 that if $\mathfrak{X}^*$ satisfies g , then there exists a universal permutation π from $[n]$ to itself such that for every positive and $\mathcal{P}$ -weakly positive vector $\mathbf{w} \in \mathfrak{X}^\#$,

$$f_i(\mathbf{w}) = g_{\pi(i)}(\mathbf{w}), \quad \text{for all } i \in [n].$$

As a result, we can construct F_π for each π , and $\mathfrak{X}^*$ satisfies g if and only if $\mathfrak{X}^\#$ satisfies F_π for some π .

In the second step, we show that for any r -vector or $\mathcal{P}$ -weakly positive vector polynomial f , one can construct $\{F_1, \dots, F_L\}$ in a finite number of steps, in which each F_i is a finite set of $\mathcal{T}$ -matrix, m -vector and $\mathcal{T}$ -weakly positive vector polynomials, such that, $\mathfrak{X}^\#$ satisfies f if and only if $(\mathfrak{X}, \mathfrak{Y})$ satisfies F_i for some $i \in [L]$. The idea of the proof is very similar to the proof of Lemma 11 so we omit it here.

Lemma 12, for the case when g is $\mathcal{P}$ -weakly positive, then follows by combining these two steps. $\square$

We can also prove the following lemma similarly.

Lemma 13. *Let g be an r -vector or a $\mathcal{P}$ -weakly positive vector polynomial. Then one can construct a finite set $\{G_1, \dots, G_L\}$ in a finite number of steps, in which every G_i , $i \in [L]$, is a finite set of $\mathcal{P}$ -matrix, $\mathcal{P}$ -diagonal matrix, r -vector, and $\mathcal{P}$ -weakly positive vector polynomials, such that*

$$(\mathfrak{X}', \mathfrak{Y}') \text{ satisfies } g \iff \exists i \in [L] \text{ such that } (\mathfrak{X}^*, \mathfrak{Y}^*) \text{ satisfies } G_i.$$

8.4 Decidability of the Rank Condition

Finally, we use these lemmas to prove Lemma 3, the decidability of the rank condition.

We start with the following simple observation. Let $F = \{f_1, \dots, f_s\}$ be a finite set of matrix and vector polynomials. For each $i \in [s]$, there is a finite set $\{F_{i,1}, \dots, F_{i,L_i}\}$ in which every $F_{i,j}$ is some finite set of polynomials, and we have the following statement:

$$(\mathfrak{X}', \mathfrak{Y}') \text{ satisfies } f_i \iff \exists j \in [L_i] \text{ such that } (\mathfrak{X}, \mathfrak{Y}) \text{ satisfies } F_{i,j}.$$

Then the conjunction of these statements over $f_i \in F$, $i \in [s]$, can be expressed in the same form: One can construct from $\{F_{i,j} : i \in [s], j \in [L_i]\}$ a new finite set $\{G_1, \dots, G_L\}$ in which every G_j is some finite set of polynomials, such that

$$\forall f \in F, [(\mathfrak{X}', \mathfrak{Y}') \text{ satisfies } f] \iff \exists j \in [L] \text{ such that } (\mathfrak{X}, \mathfrak{Y}) \text{ satisfies } G_j.$$

Now we prove Lemma 3. After $\ell - 1 \geq 0$ steps, we get a sequence of ℓ pairs

$$(\mathfrak{X}_0, \mathfrak{Y}_0), (\mathfrak{X}_1, \mathfrak{Y}_1), \dots, (\mathfrak{X}_{\ell-1}, \mathfrak{Y}_{\ell-1}),$$

and ℓ block patterns $\mathcal{T}_0, \dots, \mathcal{T}_{\ell-1}$ which satisfy condition $(\mathbf{R}_{\ell-1})$. Since we assumed that $\mathfrak{X}_0 = \{\mathbf{1}\}$, every $\mathfrak{X}_i$ in the sequence is *closed*.

We show how to check whether every matrix $\mathbf{D} \in \mathfrak{Y}_\ell$, where

$$(\mathfrak{X}_\ell, \mathfrak{Y}_\ell) = \text{gen-pair}(\mathfrak{X}_{\ell-1}, \mathfrak{Y}_{\ell-1}),$$

is block-rank-1 or not. To this end we first check if $\mathcal{P} = \text{gen}(\mathcal{T}_{\ell-1})$ is consistent with a block pattern. If not we conclude that $\mathfrak{Y}_{\ell+1}$ does not satisfy the rank condition.

Otherwise, we use $\mathcal{T}_\ell$ to denote the block pattern consistent with $\mathcal{P}$. To check the rank condition, it is equivalent to check whether $\mathfrak{Y}_\ell$ satisfies the following $\mathcal{T}_\ell$ -matrix polynomials:

$$f_{i,i',j,j'}(\mathbf{x}) = x_{i,j} \cdot x_{i',j'} - x_{i,j'} \cdot x_{i',j}, \quad \text{where } i, i' \in A_k \text{ and } j, j' \in B_k \text{ for some } k \in [r]$$

and $(A_1, B_1), \dots, (A_r, B_r)$ are the pairs in $\mathcal{T}_\ell$.

By Lemmas 11-13, we can construct a finite set $\{F_1, \dots, F_L\}$ in which every F_i is a finite set of

$$\mathcal{T}_{\ell-1}\text{-matrix, } m_{\ell-1}\text{-vector, and } \mathcal{T}_{\ell-1}\text{-weakly positive vector polynomials}$$

such that

$$\mathfrak{Y}_\ell \text{ satisfies the rank condition if and only if } (\mathfrak{X}_{\ell-1}, \mathfrak{Y}_{\ell-1}) \text{ satisfies } F_i \text{ for some } i \in [L].$$

If $\ell = 1$, then we are done, since $(\mathfrak{X}_0, \mathfrak{Y}_0)$ is finite and we can check all the polynomials in F_i for all $i \in [L]$ in a finite number of steps. Otherwise, $\ell \geq 2$ and we can use Lemmas 11–13 and the observation above to construct, for each F_i , a finite set $\{F_{i,1}, \dots, F_{i,L_i}\}$ in which every $F_{i,j}$ is a finite set of

$$\mathcal{T}_{\ell-2}\text{-matrix, } m_{\ell-2}\text{-vector, and } \mathcal{T}_{\ell-2}\text{-weakly positive vector polynomials}$$

such that

$$(\mathfrak{X}_{\ell-1}, \mathfrak{Y}_{\ell-1}) \text{ satisfies } F_i \text{ if and only if } (\mathfrak{X}_{\ell-2}, \mathfrak{Y}_{\ell-2}) \text{ satisfies } F_{i,j} \text{ for some } j \in [L_i].$$

We repeat this process until we reach the finite pair $(\mathfrak{X}_0, \mathfrak{Y}_0)$. So the checking procedure looks like a huge tree of depth ℓ . Every leaf v of the tree is associated with a finite set F_v of

$$\mathcal{T}_0\text{-matrix, } m_0\text{-vector, and } \mathcal{T}_0\text{-weakly positive vector polynomials,}$$

and $\mathfrak{Y}_\ell$ satisfies the rank condition if and only if $(\mathfrak{X}_0, \mathfrak{Y}_0)$ satisfies F_v for some leaf v of the tree.

References

- [1] BULATOV, A. The complexity of the counting constraint satisfaction problem. *Journal of the ACM* 60, 5 (2013), 34:1–34:41.
- [2] BULATOV, A., AND DALMAU, V. Towards a dichotomy theorem for the counting constraint satisfaction problem. *Information and Computation* 205, 5 (2007), 651–678.
- [3] BULATOV, A., DYER, M., GOLDBERG, L., JALSENIOUS, M., JERRUM, M., AND RICHERBY, D. The complexity of weighted and unweighted $\#CSP$. *Journal of Computer and System Sciences* 78, 2 (2012), 681–688.
- [4] BULATOV, A., AND GROHE, M. The complexity of partition functions. *Theoretical Computer Science* 348, 2 (2005), 148–186.
- [5] BULATOV, A. A. The complexity of the counting constraint satisfaction problem. In *Proceedings of the 35th International Colloquium on Automata, Languages, and Programming* (2008), pp. 646–661.
- [6] CAI, J.-Y., AND CHEN, X. A decidable dichotomy theorem on directed graph homomorphisms with non-negative weights. In *Proceedings of the 51st IEEE Annual Symposium on Foundations of Computer Science* (2010), pp. 437–446.
- [7] CAI, J.-Y., AND CHEN, X. Complexity of counting CSP with complex weights. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing* (2012), pp. 909–920.
- [8] CAI, J.-Y., CHEN, X., AND LU, P. Nonnegatively weighted $\#CSPs$: An effective complexity dichotomy. In *Proceedings of the 25th IEEE Conference on Computational Complexity* (2011), pp. 45–54.
- [9] CAI, J.-Y., CHEN, X., AND LU, P. Graph homomorphisms with complex values: A dichotomy theorem. *SIAM Journal on Computing* 42, 3 (2013), 924–1029.
- [10] CAI, J.-Y., GUO, H., AND WILLIAMS, T. A complete dichotomy rises from the capture of vanishing signatures: Extended abstract. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing* (2013), pp. 635–644.
- [11] CAI, J.-Y., GUO, H., AND WILLIAMS, T. The complexity of counting edge colorings and a dichotomy for some higher domain Holant problems. In *Proceedings of the IEEE 55th Annual Symposium on Foundations of Computer Science* (2014), pp. 601–610.
- [12] DYER, M., GOLDBERG, L., AND PATERSON, M. On counting homomorphisms to directed acyclic graphs. *Journal of the ACM* 54, 6 (2007), Article 27.
- [13] DYER, M., AND GREENHILL, C. The complexity of counting graph homomorphisms. *Random Structures & Algorithms* 17, 3–4 (2000), 260–289.
- [14] DYER, M., AND RICHERBY, D. On the complexity of $\#CSP$. In *Proceedings of the 42nd ACM Symposium on Theory of Computing* (2010), pp. 725–734.
- [15] DYER, M., AND RICHERBY, D. An effective dichotomy for the counting constraint satisfaction problem. *SIAM Journal on Computing* 42, 3 (2013), 1245–1274.
- [16] FEDER, T., AND VARDI, M. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing* 28, 1 (1999), 57–104.

- [17] FREEDMAN, M., LOVÁSZ, L., AND SCHRIJVER, A. Reflection positivity, rank connectivity, and homomorphism of graphs. *Journal of the American Mathematical Society* 20 (2007), 37–51.
- [18] GOLDBERG, L., GROHE, M., JERRUM, M., AND THURLEY, M. A complexity dichotomy for partition functions with mixed signs. *SIAM Journal on Computing* 39, 7 (2010), 3336–3402.
- [19] GROHE, M., AND THURLEY, M. Counting homomorphisms and partition functions. In *Model Theoretic Methods in Finite Combinatorics*, M. Grohe and J. Makowsky, Eds., Contemporary Mathematics, Vol. 558. American Mathematical Society, 2011, pp. 243–292.
- [20] HELL, P., AND NEŠETŘIL, J. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B* 48, 1 (1990), 92–110.
- [21] HELL, P., AND NEŠETŘIL, J. *Graphs and Homomorphisms*. Oxford University Press, 2004.
- [22] LENSTRA, H. Algorithms in algebraic number theory. *Bulletin of the American Mathematical Society* 26, 2 (1992).
- [23] THURLEY, M. The complexity of partition functions. *PhD Thesis, Humboldt Universitat zu Berlin* (2009).