



Parameterized Counting of Partially Injective Homomorphisms

Marc Roth^{1,2} 

Received: 17 July 2019 / Accepted: 15 January 2021 / Published online: 11 March 2021
© The Author(s) 2021

Abstract

We study the parameterized complexity of the problem of counting graph homomorphisms with given partial injectivity constraints, i.e., inequalities between pairs of vertices, which subsumes counting of graph homomorphisms, subgraph counting and, more generally, counting of answers to equi-join queries with inequalities. Our main result presents an exhaustive complexity classification for the problem in fixed-parameter tractable and $\#W[1]$ -complete cases. The proof relies on the framework of linear combinations of homomorphisms as independently discovered by Chen and Mengel (PODS 16) and by Curticapean, Dell and Marx in the recent breakthrough result regarding the exact complexity of the subgraph counting problem (STOC 17). Moreover, we invoke Rota’s NBC-Theorem to obtain an explicit criterion for fixed-parameter tractability based on treewidth. The abstract classification theorem is then applied to the problem of counting locally injective graph homomorphisms from small pattern graphs to large target graphs. As a consequence, we are able to fully classify its parameterized complexity depending on the class of allowed pattern graphs.

Keywords Parameterized complexity theory · Counting problems · Graph homomorphisms · Matroid lattices · Möbius function

1 Introduction

In his seminal work on the complexity of computing the permanent, Valiant [56] introduced *counting complexity* which has since then evolved into a well-studied sub-

An extended abstract of this work has been published in the conference proceedings of the 25th Annual European Symposium on Algorithms (ESA 2017) [50].

✉ Marc Roth
marc.roth@merton.ox.ac.uk

¹ Cluster of Excellence (MMCI), Saarland Informatics Campus (SIC), Saarbrücken, Germany

² Merton College, University of Oxford, Oxford, UK

field of computational complexity theory. Despite some surprising positive results, such as polynomial time algorithms for counting perfect matchings in planar graphs by the FKT method [37,38,54], counting spanning trees by Kirchhoff's Matrix Tree Theorem (cf. [44]) or counting Eulerian cycles in directed graphs using the "BEST"-Theorem [1], most of the interesting counting problems turned out to be intractable. Therefore, several relaxations such as restrictions of input classes [55,57,60] and approximate counting [3,23,31,36] were introduced. Another possible relaxation, the one this work deals with, is to consider parameterized counting problems as independently introduced by Flum and Grohe [26] and McCartin [45]. Here, problems come with an additional parameter k and a problem is fixed-parameter tractable (FPT) if it can be solved in time $g(k) \cdot \text{poly}(n)$ where n is the input size and g is a computable function, which yields fast algorithms for large instances with small parameters. On the other hand, a problem is considered intractable if it is $\#W[1]$ -hard. This stems from the fact that $\#W[1]$ -hard problems do not allow an FPT algorithm unless standard assumptions such as the Exponential Time Hypothesis (ETH) are wrong (cf. [18, Chapter 14.4]).

When investigating a family of related (counting) problems, one could aim to simultaneously solve the complexity of as many problems as possible, rather than tackling a (possibly infinite) number of problems by hand. For example, instead of proving that counting paths in a graph is hard, then proving that counting cycles is hard and then proving that counting stars is easy, one should, if possible, find a criterion that allows a classification of those problems in hard and easy cases. Unfortunately, there are results such as Ladner's Theorem [39] (stating that there are problems neither in P nor NP-hard, assuming $P \neq NP$), which give a negative answer to that goal in general. However, there are families of problems that have enough structure to allow so-called dichotomy results. One famous example, and to the best of the author's knowledge the first such result, is Schaefer's dichotomy [52], stating that every instance of the generalized satisfiability problem is either polynomial-time solvable or NP-complete. Since then much work has been done to generalize this result, culminating in recent proofs [5,61] of the Feder-Vardi-Conjecture [24]. This question was open for almost twenty years and indicates the difficulty of proving such dichotomy results, at least for decision problems. In counting complexity, however, it seems that obtaining such results is less cumbersome. One reason for this is the existence of some powerful techniques like polynomial interpolation [55], the Holant framework [2,6–10,33,58] as well as the principle of inclusion-exclusion which all have been used to establish very revealing dichotomy results.

Examples of dichotomies in parameterized counting complexity are the complete classifications of the homomorphism counting problem due to Dalmau and Jonsón [19] and the subgraph counting problem due to Curticapean and Marx [17]. For the latter, one is given graphs H and G and aims to count the number of subgraphs of G isomorphic to H , parameterized by the size of H . It is known that this problem is polynomial-time solvable if there is a constant upper bound on the size of the largest matching of H and $\#W[1]$ -hard otherwise¹. The first step in the proof of that result

¹ On the other hand the complexity of the decision version of this problem, that is, finding a subgraph of G isomorphic to H , is still unresolved. Only recently it was shown in a major breakthrough that finding bicliques is hard [40].

was the hardness result of counting matchings of size k due to Curticapean [14], which turned out to be the minimal obstruction and was then reduced to the general problem.

The approach of first establishing minimal obstructions for tractability and then reducing to the general case seemed to be the canonical way to tackle such problems. However, recently Curticapean, Dell and Marx [16] discovered that a result of Lovász [41] implies the existence of parameterized reductions that, inter alia, allow a far easier proof of the general subgraph counting problem. Lovász result states that, given simple graphs H and G , the following holds true:

$$\#\text{Emb}(H \rightarrow G) = \sum_{\rho \geq \emptyset} \mu(\emptyset, \rho) \cdot \#\text{Hom}(H/\rho \rightarrow G). \tag{1}$$

In Equation (1) the sum is over the elements of the partition lattice of $V(H)$, $\text{Emb}(H \rightarrow G)$ is the set of embeddings² from H to G and $\text{Hom}(H/\rho \rightarrow G)$ is the set of homomorphisms from H/ρ to G , where the quotient graph H/ρ is obtained from H by identifying vertices along ρ . Furthermore, μ is the Möbius function of the partition lattice. In their work Curticapean, Dell and Marx established what is now called the principle of *complexity monotonicity* [11,16,20,51]. Informally, it states that a linear combination

$$G \mapsto \sum_{i=1}^{\ell} \lambda_i \cdot \#\text{Hom}(H_i \rightarrow G)$$

for *pairwise non-isomorphic graphs* H_i is hard to compute if and only if at least one of the pattern graphs H_i for which $\lambda_i \neq 0$ has large treewidth.

Using complexity monotonicity, one only has to show two properties of (1) to obtain the dichotomy for counting subgraph embeddings: First, one has to show that a high matching number of H implies that one of the graphs H/ρ has high treewidth and second, that two (or more) terms with high treewidth and isomorphic graphs H/ρ and H/σ do not cancel out; note that the Möbius function might be negative. As there is a closed form for the Möbius function over the partition lattice it was possible to show that the sign of the Möbius function is equal whenever H/ρ and H/σ are isomorphic.

1.1 Contributions

In the current work, we strengthen the previously outlined framework to the more general case of *partially injective homomorphisms*. This notion of graph homomorphisms interpolates between unrestricted homomorphisms and subgraph embeddings, i.e., fully injective homomorphisms. Informally, we consider pairs (H, I) of graphs H and inequalities I between the vertices of H . Given a further graph G , we say that a homomorphism from H to G is *partially injective* with respect to I if for every inequality “ $u \neq v$ ” $\in I$, the homomorphism maps u and v to different vertices. Note that a subgraph embedding can be modelled by choosing as I the set of all inequalities.

² Note that embeddings and subgraphs are equal up to automorphisms, that is, counting embeddings and counting subgraphs are essentially the same problems.

Furthermore, a reader familiar with database theory might recognize partially injective homomorphisms as answers to *equi-join* queries (cf. [47]) with inequalities: Here one is given a database B and a query

$$\varphi = x_1, \dots, x_\ell : a_1 \wedge a_2 \wedge \dots \wedge a_m,$$

where the x_i are variables and the a_i are inequalities $x_i \neq x_j$ or atoms $R(x_i, x_j, x_k)$ for some relation R of the database. The goal is then to compute the number of tuples (v_1, \dots, v_ℓ) of elements of the database B for which all relational constraints and inequalities are satisfied. We emphasize that *all* results we obtain in this paper for partially injective homomorphisms hold as well for equi-join queries with inequalities.³

Now, for the statement of our results, we adopt the notation of [16] and say that a graph F is a *spasm* of a graph with inequalities (H, I) if F can be obtained from H by contracting vertices along inequalities in I ; the formal definition is given in Sect. 2. Furthermore, we write $t(H, I)$ for the maximum treewidth of spasms of (H, I) . The following result is shown in Sect. 3 and establishes an exhaustive classification for the parameterized complexity of counting partially injective homomorphisms.

Theorem 1 *Let \mathcal{H} be a recursively enumerable class of graphs with inequalities. Given $(H, I) \in \mathcal{H}$ and an arbitrary graph G , we wish to compute the number of partially injective homomorphisms from H to G with respect to I .*

When parameterized by $|V(H)|$, this problem is fixed-parameter tractable if there exists a constant upper bound on the treewidth of all spasms of pairs $(H, I) \in \mathcal{H}$. Otherwise, the problem is #W[1]-complete and cannot be solved in time

$$g(|V(H)|) \cdot |V(G)|^{o(t(H, I)/\log(t(H, I)))},$$

for any computable function g unless the Exponential Time Hypothesis (ETH) fails.

In particular, we obtain the following algorithmic result.

Theorem 2 *There exists a computable function g such that the number of partially injective homomorphisms from a graph H to a graph G with respect to a set of inequalities I can be computed in time*

$$g(|V(H)|) \cdot |V(G)|^{t(H, I)+1}.$$

We remark that Theorems 1 and 2 yield an almost tight conditional lower bound on the fine-grained complexity of counting partially injective homomorphisms. The missing gap of $\log(t(H, I))$ is due to the unresolved open problem of whether it is possible to “beat treewidth” [42].

³ Note that partially injective homomorphisms and answers to equi-join queries are *equivalent* if only one binary relation E is allowed; in this case a database is a graph. Reductions from (counting) answers to database queries to (counting) graph homomorphisms are ubiquitous in the CSP community (cf. [19] and [27, Chapter 13]) and we refer the reader to e.g. Sect. 6.2 in the full version [21] of [20] for a construction that applies without modification to the results for graphs in the current paper.

We will see that the previous results can easily be applied to fully understand the parameterized complexity of a wide range of counting problems. We chose counting of *locally injective homomorphisms* as an exemplary problem, which is studied in Sect. 4. Here, a homomorphism from H to G is called locally injective if it maps two vertices of H to different vertices of G if they have a common neighbour in H . The study of locally injective homomorphisms goes back to Nešetřil [46]; a particular application is given by what is called distance constrained labellings of graphs (cf. [25]).

Corollary 1 *Let \mathcal{H} be a recursively enumerable class of graphs and let $\hat{\mathcal{H}}$ be the set of all graphs that can be obtained from graphs in \mathcal{H} by identifying vertices that have a common neighbour. Given $H \in \mathcal{H}$ and an arbitrary graph G , we wish to count the locally injective homomorphisms from H to G .*

When parameterized by $|V(H)|$, this problem is fixed-parameter tractable if there exists a constant upper bound on the treewidth of all graphs in $\hat{\mathcal{H}}$. Otherwise, the problem is $\#W[1]$ -complete.

We will see that the results on the fine-grained complexity of counting locally injective homomorphisms transfer from the general case as well.

A further question considered is whether counting of locally injective homomorphisms admits “real” fixed-parameter tractable cases, that is, instances of \mathcal{H} for which the previous problem is fixed-parameter tractable but not solvable in polynomial time under standard assumptions; note that neither counting of homomorphisms nor counting of subgraph embeddings admits real fixed-parameter tractable cases [17, 19]. The subsequent result establishes that the set of all trees induces a real fixed-parameter tractable case for counting of locally injective homomorphisms.

Theorem 3 *The problem of counting locally injective homomorphisms from a tree T to a graph G is $\#P$ -hard and fixed-parameter tractable when parameterized by $|V(T)|$.*

The proof of the previous theorem relies on the following $\#P$ -hardness result regarding the subgraph counting problem restricted to trees; it is shown in Sect. 4.1 and might be of independent interest.

Theorem 4 *The problem of, given trees T_1 and T_2 , computing the number of subtrees of T_2 that are isomorphic to T_1 is $\#P$ -hard.*

Finally, Sect. 5 establishes a generalization of Theorem 1 to computing linear combinations of partially injective homomorphisms. Intuitively, this will enable us to consider not only the problem of computing the number of partially injective homomorphisms from a fixed graph, but also from a set of graphs of equal size. A particular example of the latter is the problem of counting acyclic subgraphs of size k .

1.2 Techniques

The main ingredient of our proofs is the framework of linear combinations of homomorphisms and complexity monotonicity. We decided to present this framework using

Lovász' notion of a *quantum graph* [41, Chapter 6], which is defined to be a formal linear combination of simple graphs with finite support:

$$Q := \sum_H \lambda_H \cdot H,$$

where the sum is over all graphs and $\lambda_H \neq 0$ holds only for finitely many graphs. Graph parameters extend to quantum graphs linearly. Consider e.g.

$$\#\text{Hom}(Q \rightarrow G) := \sum_H \lambda_H \cdot \#\text{Hom}(H \rightarrow G).$$

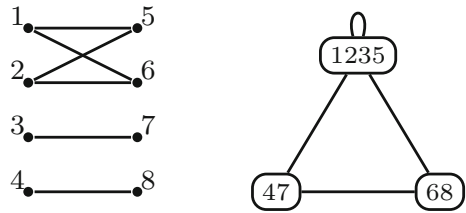
Using quantum graphs, the principle of complexity monotonicity can be rephrased as follows: Computing homomorphisms from Q is precisely as hard as computing the hardest term $\#\text{Hom}(H \rightarrow G)$ for which $\lambda_H \neq 0$. Fortunately, the complexity of counting (unrestricted) graph homomorphisms is fully understood by the classification of Dalmau and Jonsson [19]: Informally, their result states that computing $\#\text{Hom}(H \rightarrow G)$ is fixed-parameter tractable if the treewidth of H is bounded and $\#\text{W}[1]$ -complete otherwise; this is made formal in Sect. 2.

Given a graph with inequalities (H, I) , we use Möbius inversion over a certain geometric lattice to construct a quantum graph $Q[H, I]$ such that the number of partially injective homomorphisms from H to a graph G with respect to I equals $\#\text{Hom}(Q[H, I] \rightarrow G)$. However, the crucial part will be the study of the coefficients of $Q[H, I]$. By complexity monotonicity, it is necessary to understand which coefficients of $Q[H, I]$ are non-zero to resolve the complexity of counting partially injective homomorphisms from (H, I) . Unfortunately, it has turned out that the analysis of the coefficients of a quantum graph seems to be much more involved than establishing its existence, as witnessed by recent publications [16,20,51]. In case of subgraph embeddings, Curticapean, Dell and Marx [16] used Equation (1) and the explicit formula of the Möbius function of the partition lattice to obtain both, a quantum graph and an explicit criterion on whether a coefficient vanishes.

In the current paper, we establish a connection between the coefficients of $Q[H, I]$ and the graphic matroid induced by the inequalities I on the vertex set of H . In particular, it turns out that the lattice over which we perform Möbius inversion in the construction of $Q[H, I]$ is precisely the so-called *lattice of flats* of this matroid. This will allow us to rely on Rota's NBC Theorem [49] and we obtain an explicit expression of the set of graphs for which the coefficients in $Q[H, I]$ are non-zero.

The classification for counting locally injective homomorphisms turns then into an application of the general theorem. For $\#\text{P}$ -hardness of the subgraph counting problem restricted to trees, we adapt the idea of the “skeleton graph” by Goldberg and Jerrum [28] and reduce directly from computing the permanent, the latter of which is $\#\text{P}$ -hard as shown by Valiant [56]. To transfer this result to locally injective homomorphisms we use the well-known observation that every locally injective homomorphism from a tree to a tree is fully injective.

Fig. 1 *Left:* A graph H with vertex set $V(H) = \{1, 2, 3, 4, 5, 6\}$. *Right:* The quotient graph H/δ for the partition $\delta = \{\{1, 2, 3, 5\}, \{47\}, \{68\}\}$



2 Preliminaries

Given a finite set A , we write $\#A$ or $|A|$ for its cardinality and given a natural number $k \in \mathbb{N}$, we write $[k]$ for the set $\{1, \dots, k\}$. The *sign* of a real number r is denoted by $\text{sign}(r)$ and defined to be 1 if $r > 0$, -1 if $r < 0$ and 0 if $r = 0$. Given sets A and B and a function $f : A \rightarrow B$, we write $\text{supp}(f) = f^{-1}(B \setminus \{0\})$ for the *support* of f . Furthermore, given a subset $S \subseteq A$, we write $f|_S$ for the restriction of f to S . A *partition* of a finite set A is a set of pairwise disjoint and non-empty sets, called *blocks*, whose union is A .

2.1 Graph Theory

Graphs in this work are considered simple, undirected and without self-loops, unless stated otherwise. More precisely, a graph G is a pair of a finite set $V(G)$ of vertices and a symmetric and irreflexive relation $E(G) \subseteq V(G)^2$. We might only write V and E for vertices and edges if the graph is clear from the context. Furthermore we write $\{u, v\}$ to denote edges, emphasizing undirectedness. If u and v are vertices of G and $e = \{u, v\}$ is an edge of G we call u and v *adjacent* and say that u and v are *incident* to e . Given a vertex v of G we write $N_G(v)$ for the subset of vertices of G that are adjacent to v ; if G is clear from the context, we just write $N(v)$. Throughout this paper, we use n for the number of vertices and m for the number of edges.

If a graph H is obtained from G by deleting a set of edges and a set of vertices of G , including incident edges, then H is called a *subgraph* of G . Given a subset \hat{V} of $V(G)$ we write $G[\hat{V}]$ for the graph with vertices \hat{V} and edges $E \cap \hat{V}^2$. The resulting graph is called an *induced subgraph* of G .

Given a graph H and a partition δ of $V(H)$, the *quotient graph* H/δ contains a vertex for each block of δ , and two blocks B and B' are made adjacent if there are $u \in B$ and $v \in B'$ such that $\{u, v\} \in E(H)$ (see Fig. 1). Note that B and B' might be equal; in particular H/δ may contain self-loops. Intuitively, H/δ is obtained from H by contracting the vertices that are contained in the same set of δ and deleting multiple edges, but *keeping* self-loops.

2.1.1 Treewidth

Intuitively, the treewidth of a graph measures how tree-like it is. In particular, graphs with small treewidth allow a decomposition in small separators. These decompositions are the basis of many efficient dynamic programming algorithms (DPs) for problems

that are computationally infeasible on graphs without the promise of having small treewidth. Roughly speaking, those algorithms can be seen as generalizations of known and simple DPs that solve the corresponding problems on trees. Although we will mainly need the concept in a black-box manner, we decided give the formal definition for reasons of self-containment.

Definition 1 (See e.g. **Chapt. 7.2 in [18]**) Let G be a graph. A *tree decomposition* of G consists of a tree T and a collection of subsets $B_t \subseteq V(G)$ for all $t \in V(T)$ called *bags* such that

1. $\bigcup_{t \in V(T)} B_t = V(G)$,
2. for every edge $\{u, v\} \in E(G)$ there exists a vertex t of T such that $\{u, v\} \subseteq B_t$, and
3. for every vertex $v \in V(G)$ the graph $T[\{t \in V(T) \mid v \in B_t\}]$ is connected.

The *width* of a tree decomposition is $\max_{t \in V(T)} |B_t| - 1$ and the *treewidth* of a graph G , denoted by $\text{tw}(G)$, is the minimum width a tree decomposition of G can have. We say that a class of graphs has *bounded treewidth* if there exists a constant c such that the treewidth of every graph in the class is at most c .

2.1.2 Homomorphisms and Subgraph Embeddings

A *homomorphism* from a graph H to a graph G is a mapping

$$h : V(H) \rightarrow V(G)$$

that preserves edges. In other words, for every edge $\{u, v\} \in E(H)$ it holds that $\{h(u), h(v)\} \in E(G)$.

An *embedding*, also called *subgraph embedding*, is an injective homomorphism and a *strong embedding* from H to G is an embedding h such that for all pairs u and v of vertices of H we have that

$$\{u, v\} \in E(H) \Leftrightarrow \{h(u), h(v)\} \in E(G).$$

A bijective strong embedding is called an *isomorphism* and we say that two graphs H and G are *isomorphic*, denoted by $H \simeq G$, if there exists an isomorphism from H to G .⁴ An isomorphism from a graph to itself is called an *automorphism* and we write $\text{Aut}(H)$ for the set of all automorphisms of a graph H . We will furthermore use the following notations for sets of homomorphisms and its variants.

Definition 2 Let H and G be graphs.

1. $\text{Hom}(H \rightarrow G)$ denotes the set of homomorphisms from H to G .
2. $\text{Emb}(H \rightarrow G)$ denotes the set of embeddings from H to G .
3. $\text{Sub}(H \rightarrow G)$ denotes the set of subgraphs of G isomorphic to H .

⁴ We will implicitly identify graphs by their isomorphism types which allows us to refer e.g. to *the* complete graph on k vertices.

It will be very convenient to write $\#\text{Hom}(H \rightarrow \star)$ for the function that maps a graph G to the number $\#\text{Hom}(H \rightarrow G)$. The functions

$$\#\text{Emb}(H \rightarrow \star) \text{ and } \#\text{Sub}(H \rightarrow \star)$$

are defined likewise. Observe that the set $\text{Emb}(H \rightarrow G)$ can be partitioned by the images of its elements. Furthermore, all classes of the induced equivalence relations have size $\#\text{Aut}(H)$. As the images of the embeddings from H to G are precisely the subgraphs of G isomorphic to H , we obtain the following well-known fact.

Fact 1 (Folklore) *Let H be a graph. We have that*

$$\#\text{Emb}(H \rightarrow \star) = \#\text{Aut}(H) \cdot \#\text{Sub}(H \rightarrow \star).$$

2.2 Counting Problems

A *counting problem* is a function $P : i\text{Sigma}^* \rightarrow \mathbb{N}$. Here $i\text{Sigma}$ denotes a fixed finite alphabet which the reader can assume to be $\{0, 1\}$ for all purposes in this work.

2.2.1 Parameterized Counting Complexity

The field of parameterized counting was introduced by Flum and Grohe [26] and by McCartin [45] with the goal of adapting the relaxation of parameterization for decision problems to the counting realm. A *parameterized counting problem* is a pair (P, κ) of a counting problem and a computable parameterization $\kappa : i\text{Sigma}^* \rightarrow \mathbb{N}$. The following are two parameterized counting problems that are crucial for this work.

#CLIQUE **Input:** A graph G and a positive integer k .

Parameter: $\kappa(G, k) := k$.

Output: The number of cliques of size k in G .

#HOM(\mathcal{H}) **Input:** A graph G and a graph $H \in \mathcal{H}$.

Parameter: $\kappa(G, H) := |V(H)|$.

Output: $\#\text{Hom}(H \rightarrow G)$, i.e., the number of homomorphisms from H to G .

Note that in the last example, \mathcal{H} is a fixed, possibly infinite set of graphs that is *not* part of the input. If we choose \mathcal{H} as the set of all complete graphs, then $\#\text{HOM}(\mathcal{H})$ and $\#\text{CLIQUE}$ are equivalent up to a factor of $k!$, because the image of a homomorphism from a clique of size k is itself a clique of size k . Let us furthermore remark that we will omit defining κ explicitly from now on as it will be clear from the context.

A parameterized counting problem (P, κ) is called *fixed-parameter tractable (FPT)* if there exists a computable function f and a deterministic algorithm \mathbb{A} that, on input x , computes $P(x)$ in time

$$f(\kappa(x)) \cdot |x|^{O(1)}.$$

In particular, \mathbb{A} is called an *FPT algorithm*. For the theory of parameterized hardness, especially for counting problems, it is common to rely on the following variant of Turing-reductions.

Definition 3 Let (P, κ) and (P', κ') be parameterized counting problems. A *parameterized Turing-reduction* from (P, κ) to (P', κ') is an FPT algorithm \mathbb{A} that solves (P, κ) and is given oracle access to (P', κ') . Furthermore, every oracle query y that is posed by \mathbb{A} on input x must satisfy $\kappa'(y) \leq g(\kappa(x))$ for some computable function g independent of x .

We write $(P, \kappa) \leq_{\text{fpt}}^T (P', \kappa')$ if a parameterized Turing reduction from (P, κ) to (P', κ') exists.

A problem (P, κ) is called *#W[1]-hard* if $\#\text{CLIQUE} \leq_{\text{fpt}}^T (P, \kappa)$ and it is called *#W[1]-complete* if additionally $(P, \kappa) \leq_{\text{fpt}}^T \#\text{CLIQUE}$.⁵ The following parameterized classification of the homomorphism counting problem is due to Dalmou and Jonsson and will be the basis for the hardness results in this paper.

Theorem 5 ([19]) *Let \mathcal{H} be a recursively enumerable class of graphs. If \mathcal{H} has bounded treewidth then $\#\text{HOM}(\mathcal{H})$ is polynomial-time solvable. Otherwise $\#\text{HOM}(\mathcal{H})$ is #W[1]-complete.*

The tractable case in the previous theorem stems from the following algorithmic result for counting graph homomorphisms.

Theorem 6 ([22], see also the full version of [16]) *Let H be a graph with k vertices and treewidth t and let G be a graph with n vertices. Then $\#\text{Hom}(H \rightarrow G)$ can be computed in time*

$$\text{poly}(k) \cdot n^{t+1}.$$

2.2.2 Fine-Grained Complexity Theory

The conditional lower bounds in this work are based on the *Exponential Time Hypothesis (ETH)* which is due to Impagliazzo and Paturi [34] and asserts that 3-SAT cannot be solved in time $\exp(o(m))$, where m is the number of clauses of the input formula.

It is known that #W[1]-hard problems are not fixed-parameter tractable unless ETH fails. An even stronger result states that #CLIQUE cannot be solved in time $g(k) \cdot n^{o(k)}$ for any function g unless ETH fails [12, 13]. We will rely on the following result about the fine-grained complexity of #HOM(\mathcal{H}) which is an easy consequence of Marx' work on the fine-grained complexity of the constraint satisfaction problem.⁶

⁵ Let us remark that we slightly abuse notation here: Containment in #W[1] is usually defined via parameterized parsimonious reductions [26]. Hence, formally, our notion of #W[1]-completeness does not imply containment in #W[1] but rather “equivalence” for #W[1] (see e.g. [20]).

⁶ The reader familiar with Marx' work is encouraged to take a look at [42, Corollary 6.2] which states a similar version of Theorem 7 for the *Partitioned Subgraph Isomorphism Problem*. However, the latter admits a tight reduction to counting of vertex-colourful subgraphs embeddings, the clearest presentation of which can be found in the Thesis of Curticapean [15, Lemma 5.5]. Vertex-colourful subgraph embeddings are precisely vertex-colourful homomorphisms, as colourfulness implies injectivity. Finally, reducing counting of vertex-colourful homomorphisms to counting of uncoloured homomorphism is an easy application of the inclusion-exclusion principle.

Theorem 7 ([42]) *Let \mathcal{H} be a recursively enumerable class of graphs of unbounded treewidth. Then $\#\text{HOM}(\mathcal{H})$ cannot be solved in time*

$$g(|V(H)|) \cdot |V(G)|^{o(\text{tw}(H)/\log(\text{tw}(H)))},$$

for any function g , unless ETH fails.

2.3 Quantum Graphs

We follow the notation of the textbook of Lovász—see Chapt. 6 in [41].

Definition 4 (*Quantum graphs*) A quantum graph is a formal linear combination of simple graphs without self-loops with finite support and rational coefficients. We write

$$Q = \sum \lambda_H \cdot H,$$

where the sum is over all (isomorphism types of) graphs and $\lambda_H \in \mathbb{Q}$ is zero for all but finitely many H . We write $\text{supp}(Q)$ for the support of Q , that is, the set containing all graphs with a non-zero coefficient in Q . Furthermore, we call graphs in $\text{supp}(Q)$ constituents of Q .

The functions we have seen in the preceding section extend to quantum graphs linearly. We make it explicit for the case of homomorphisms:

$$\#\text{Hom}(Q \rightarrow G) := \sum \lambda_H \cdot \#\text{Hom}(H \rightarrow G).$$

The study of homomorphism numbers of quantum graphs has far-reaching consequences for the complexity classification of counting problems that require to count small structures in large graphs. They were first (implicitly) used in this context by Curticapean, Dell and Marx in a breakthrough result regarding the exact complexity of the subgraph counting problem [16]. Roughly speaking, they provided a construction of a quantum graph $Q[H]$, given a graph H , such that

$$\#\text{Hom}(Q[H] \rightarrow \star) = \#\text{Sub}(H \rightarrow \star).$$

After that, they established that computing the number of homomorphisms from a quantum graph is *precisely as hard* as counting homomorphisms from the hardest constituent. This property is referred to as *complexity monotonicity* and allows to obtain exhaustive complexity classifications for counting problems that can be expressed as homomorphisms numbers from quantum graphs by ultimately relying on the fact that the problem of counting homomorphisms is fully understood by Theorem 5.

We are now able to provide a formal statement of complexity monotonicity.

Theorem 8 (Complexity Monotonicity [16]) *Let $Q = \sum_{i \in [k]} \lambda_i \cdot H_i$ be a fixed quantum graph with constituents $\{H_i\}_{i \in [k]}$. There exists a computable function f and an*

algorithm \mathbb{A} that is given a graph G with $n > 0$ vertices as input and has oracle access to

$$\#\text{Hom}(Q \rightarrow \star),$$

and computes $\#\text{Hom}(H_i \rightarrow G)$ for all $i \in [k]$ in time $f(Q) \cdot n$. Additionally, every graph G' that is posed as oracle query has at most $f(Q) \cdot n$ vertices.

We point out that the principle of complexity monotonicity was independently (and less explicitly) discovered by Chen and Mengel [11] in the context of counting answers to conjunctive queries. In particular, their version of complexity monotonicity is proved for the more general case of logical structures with constant arity and, additionally, allows to include existentially quantified variables.

2.4 Combinatorics

A *partially ordered set (poset)* is a pair (L, \leq) of a finite universe L and a binary relation \leq over L which is reflexive, transitive and anti-symmetric. We write $\sigma \leq \rho$ if (σ, ρ) is contained in \leq and we write $\sigma \geq \rho$ if $\rho \leq \sigma$.

A pair $\sigma, \rho \in L$ has a *least upper bound* $\sigma \vee \rho$ if

- $\sigma \vee \rho \geq \sigma$ and $\sigma \vee \rho \geq \rho$, and
- for all $\delta \in L$ with $\delta \geq \sigma$ and $\delta \geq \rho$ it holds that $\delta \geq \sigma \vee \rho$.

A pair $\sigma, \rho \in L$ has a *greatest lower bound* $\sigma \wedge \rho$ if

- $\sigma \wedge \rho \leq \sigma$ and $\sigma \wedge \rho \leq \rho$, and
- for all $\delta \in L$ with $\delta \leq \sigma$ and $\delta \leq \rho$ it holds that $\delta \leq \sigma \wedge \rho$.

A poset (L, \leq) is called a *lattice* if every pair $\sigma, \rho \in L$ has a least upper bound and a greatest lower bound.

2.4.1 Matroids

We will follow the notation of the first chapter of the standard textbook of Oxley [48].

Definition 5 A *matroid* M is a pair (E, \mathcal{I}) of a finite set E and a non-empty subset $\mathcal{I} \subseteq \mathcal{P}(E)$ such that for every pair $A, B \subseteq E$ we have that

1. if $A \in \mathcal{I}$ and $B \subseteq A$ then $B \in \mathcal{I}$, and
2. if $A, B \in \mathcal{I}$ and $|B| < |A|$ then there exists an element $a \in A \setminus B$ such that $B \cup \{a\} \in \mathcal{I}$.

We call E the *ground set* of M . Furthermore, elements of \mathcal{I} are referred to as *independent sets* of M . In particular, a subset-maximal element of \mathcal{I} is called a *basis* of M and the *rank* of M , denoted by $\text{rk}(M)$, is defined to be the size of a basis.

We point out that $\text{rk}(M)$ is well-defined as every basis has the same cardinality due to (2). Now given a subset $X \subseteq E$ we write $\text{rk}(X)$ for the rank of X which is defined to

be the size of the largest independent set $A \subseteq X$. Furthermore, we define the *closure* of X to be

$$\text{cl}(X) := \{e \in E \mid \text{rk}(X \cup \{e\}) = \text{rk}(X)\}.$$

By definition we have that $\text{rk}(X) = \text{rk}(\text{cl}(X))$. Furthermore, we call X a *flat* if $\text{cl}(X) = X$. Now given a matroid $M = (E, \mathcal{I})$ we write $L(M)$ for the set of all flats of M and observe that subset inclusion over $L(M)$ constitutes a lattice, called the *lattice of flats* of M . The least upper bound of two flats X and Y is given by $\text{cl}(X \cup Y)$ and the greatest lower bound is given by $X \cap Y$. We point out that the lattices of flats are precisely the geometric lattices and refer the interested reader to e.g. Chapt. 3 of [59] and Chapt. 1.7 of [48] for a treatment of the latter.

We will be particularly interested in graphic matroids and their lattices of flats in this work. In what follows, given a subset A of edges of a graph $G = (V, E)$, we write $G[A]$ for the graph with vertices V and edges A , that is, $G[A] := (V, A)$.

Definition 6 Let $G = (V, E)$ be a graph. The *graphic matroid* $M(G)$ has as ground set the edges E of G and a set of edges $A \subseteq E$ is independent if the graph $G[A]$ does not contain a cycle.

If $G = (V, E)$ is connected then the bases of $M(G)$ are precisely the spanning trees of G . If G consists of several connected components then every basis of $M(G)$ induces a spanning tree for each of the components and vice versa. Now given a subset of edges $X \subseteq E$, it is known that

$$\text{rk}(X) = |V| - \text{comp}(G[X]),$$

where comp denotes the number of connected components (see, for instance, Chapt. 1.7 of [48]). Note that, in particular, every isolated vertex of $G[X]$ is a connected component.

2.4.2 Möbius Inversion

The Möbius inversion theorem is a crucial ingredient in the construction of the quantum graph $Q[H, I]$ and the analysis of its coefficients in Sect. 3. We provide a concise and self-contained introduction in what follows. An excellent and comprehensive treatment of this topic can be found in the standard textbook of Stanley [53, Chapt. 1-3].

Definition 7 Let (L, \leq) be a poset and let $f : L \rightarrow \mathbb{C}$ be a function. Then the *zeta transformation* $\zeta f : L \rightarrow \mathbb{C}$ is given by

$$\zeta f(\sigma) := \sum_{\rho \geq \sigma} f(\rho).$$

Theorem 9 (Möbius Inversion, cf. Proposition 3.7.2 in [53]) *Let (L, \leq) be a poset. There exists a computable function $\mu_L : L \times L \rightarrow \mathbb{Z}$ such that for all $f : L \rightarrow \mathbb{C}$ and $\sigma \in L$ we have that*

$$f(\sigma) = \sum_{\rho \geq \sigma} \mu_L(\sigma, \rho) \cdot \zeta f(\rho).$$

The function μ_L is called the Möbius function of L .

If the poset is clear from the context we will drop the L and only write μ . We will furthermore depend on the following corollary of Rota's NBC Theorem which is concerned with the sign of the Möbius function over a lattice of flats; note that \emptyset is the (unique) smallest element of every such lattice.

Theorem 10 (Theorem 4 in [49]) *Let (L, \leq) be the lattice of flats of some matroid and let $\rho \in L$ be a flat. Then we have that*

$$\text{sign}(\mu_L(\emptyset, \rho)) = (-1)^{rk(\rho)}.$$

We point out that the empty set \emptyset is the (unique) smallest element of the lattice of flats of any matroid.

3 Partially Injective Homomorphisms

A *graph with inequalities* is a pair (H, I) where H is a graph and I is an irreflexive and symmetric relation $I \subseteq V(H)^2$. We say that I is a set of inequalities. Intuitively, given a graph G and a graph with inequalities (H, I) , a homomorphism $h \in \text{Hom}(H \rightarrow G)$ satisfies I if for every inequality $\{u, v\} \in I$, it holds that $h(u) \neq h(v)$. Formally, we define the set of homomorphisms that satisfy I in terms of partially injective homomorphisms

$$\text{PartInj}(H, I \rightarrow G) := \{h \in \text{Hom}(H \rightarrow G) \mid \forall \{u, v\} \in I : h(u) \neq h(v)\}.$$

Given a class \mathcal{H} of graphs with inequalities, we write $\#\text{PARTINJ}(\mathcal{H})$ for the parameterized counting problem of, given $(H, I) \in \mathcal{H}$ and a graph G , computing $\#\text{PartInj}(H, I \rightarrow G)$; the parameter is $|V(H)|$.

Now let (H, I) be a graph with inequalities. It will be very convenient to assume that I does not contain an edge of H and, indeed, we can make this assumption without losing generality. To see this, observe that whenever $\{u, v\}$ is an edge of H , no homomorphism in $\text{Hom}(H \rightarrow G)$ can map u and v to the same vertex in G , because otherwise the image of u and v would have a self-loop. As we do not allow input graphs G with self-loops, every homomorphism h satisfies $h(u) \neq h(v)$, regardless of whether $\{u, v\} \in I$.

Our classification theorem requires us to generalize quotient graphs to sets of inequalities as follows. Let (H, I) be a graph with inequalities and let ρ be a subset of I . Now consider the graph with vertices $V(H)$ and edges ρ . The connected components of this graph induce a partition $\text{Part}(\rho)$ of $V(H)$. In particular, every isolated vertex constitutes a singleton in the partition. We abuse notation and write H/ρ for the quotient graph $H/\text{Part}(\rho)$ in what follows; consult Fig. 2 for an illustration.

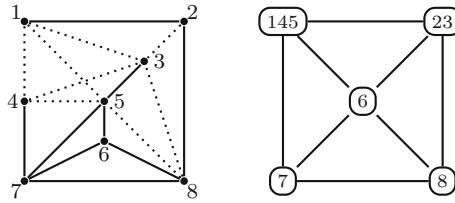


Fig. 2 *Left:* A graph H (solid lines) with inequalities I (dotted lines). *Right:* The graph H/ρ for the subset $\rho = \{\{1, 4\}, \{1, 5\}, \{2, 3\}\}$ of I . Note that the connected components of the graph with vertices $V(H)$ and edges ρ are precisely $\{1, 4, 5\}, \{2, 3\}, \{6\}, \{7\}$ and $\{8\}$, which constitutes a partition of $V(H)$. As H/ρ does not contain a self-loop, it is a spasm of (H, I) and hence a constituent of the quantum graph $Q[H, I]$ by Theorem 11

Graphs H/ρ for $\rho \subseteq I$ will turn out to be constituents of the quantum graph for (H, I) if they do not contain self-loops.⁷ For this reason, we adopt the notation of [16] and call a graph F a *spasm* of (H, I) if F does not contain self-loops and is isomorphic to a graph H/ρ for $\rho \subseteq I$. In other words, F is a spasm of (H, I) if it can be obtained from H by successively identifying vertices of H along inequalities in I without creating self-loops.

We write $\text{Spasm}(H, I)$ for the set of all spasms of (H, I) and we set

$$t(H, I) := \max_{F \in \text{Spasm}(H, I)} \text{tw}(F).$$

Given a class \mathcal{H} of graphs with inequalities, we define $\text{Spasm}(\mathcal{H})$ to be the set containing every graph F that is the spasm of some $(H, I) \in \mathcal{H}$.

The following is the main result of the current section.

Theorem 11 *Let (H, I) be a graph with inequalities. Then there exists a quantum graph $Q[H, I]$ which satisfies*

$$\#\text{PartInj}(H, I \rightarrow \star) = \#\text{Hom}(Q[H, I] \rightarrow \star).$$

In particular, the mapping $(H, I) \mapsto Q[H, I]$ is computable and

$$\text{supp}(Q[H, I]) = \text{Spasm}(H, I).$$

Let us first emphasize the following consequence of Theorem 11 for the parameterized and fine-grained complexity of $\#\text{PARTINJ}(\mathcal{H})$.

Corollary 2 (Theorem 1, restated) *Let \mathcal{H} be a recursively enumerable class of graphs with inequalities. If the treewidth of $\text{Spasm}(\mathcal{H})$ is bounded, then $\#\text{PARTINJ}(\mathcal{H})$ is fixed-parameter tractable. Otherwise $\#\text{PARTINJ}(\mathcal{H})$ is $\#W[1]$ -complete and cannot be solved in time*

⁷ Recall that quotient graphs may contain self-loops. More precisely, this happens if two adjacent vertices are contained in the same block of the partition. However, since the input graphs of our problem do not contain self-loops, we will see that we can safely ignore those partitions.

$$g(|V(H)|) \cdot |V(G)|^{o(t(H,I)/\log(t(H,I)))},$$

for any computable function g unless ETH fails.

Proof We show that the problems $\#\text{PARTINJ}(\mathcal{H})$ and $\#\text{HOM}(\text{Spasm}(\mathcal{H}))$ are interreducible with respect to parameterized Turing-reductions. For the reduction $\#\text{PARTINJ}(\mathcal{H}) \leq_{\text{fpt}}^T \#\text{HOM}(\text{Spasm}(\mathcal{H}))$ we observe that by Theorem 11

$$\#\text{PartInj}(H, I \rightarrow \star) = \#\text{Hom}(Q[H, I] \rightarrow \star) = \sum_{F \in \text{Spasm}(H, I)} \lambda_F \cdot \#\text{Hom}(F \rightarrow \star), \quad (2)$$

for computable coefficients λ_F .

For the reduction $\#\text{HOM}(\text{Spasm}(\mathcal{H})) \leq_{\text{fpt}}^T \#\text{PARTINJ}(\mathcal{H})$, we let F be a spasm in $\text{Spasm}(\mathcal{H})$ and G be an instance of $\#\text{HOM}(\text{Spasm}(\mathcal{H}))$. As \mathcal{H} is recursively enumerable, we can find $(H, I) \in \mathcal{H}$ with $F \in \text{Spasm}(H, I)$ in time only depending on F . By complexity monotonicity (Theorem 8) we can use the oracle for $\#\text{PartInj}(H, I \rightarrow \star) = \#\text{Hom}(Q[H, I] \rightarrow \star)$ to compute $\#\text{Hom}(F \rightarrow G)$.

The classification in fixed-parameter tractable and $\#\text{W}[1]$ -complete cases follows then by Theorem 5. Furthermore, we observe that all reductions are tight in the sense that the running time is linear in the size of the target graphs G and that every oracle query only increases the size of the target graphs by a factor depending on the parameter. Consequently, the conditional lower bound under ETH holds by Theorem 7. \square

The algorithm for counting partially injective homomorphisms follows in a similar way:

Proof of Theorem 2 Holds by Equation (2) and the treewidth-based dynamic programming algorithm for counting homomorphisms as given by Theorem 6. \square

The proof of Theorem 11 consists of two steps. In the first one, we will invoke Möbius inversion over the lattice of flats of the graphic matroid induced by the inequalities. This will allow us to construct the quantum graph $Q[H, I]$. In the second step, we will then use Theorem 10 to prove that the constituents of $Q[H, I]$ are precisely the spasms of (H, I) .

Starting with the first step, we consider the *inequality graph* $\mathbb{I}(H, I)$ of a given graph with inequalities (H, I) . The vertices of $\mathbb{I}(H, I)$ are $V(H)$ and the edges are I . We write $M(H, I)$ for the graphic matroid of $\mathbb{I}(H, I)$ and we write $L(H, I)$ for the lattice of flats of $M(H, I)$. Recall that a flat of $M(H, I)$ is a subset ρ of I satisfying that adding any element $i \in I \setminus \rho$ to ρ will decrease the number of connected components by 1. Now observe that, given $\rho \in I$, we can define $\mathbb{I}(H, I)/\rho$ similar to H/ρ , as both, $\mathbb{I}(H, I)/\rho$ and H have the same vertex set $V(H)$. In particular, we write I/ρ for the edges of $\mathbb{I}(H, I)/\rho$ and obtain that $(H/\rho, I/\rho)$ is a graph with inequalities as well. This allows us to prove correctness of the following zeta transformation.

Lemma 1 *Let (H, I) be a graph with inequalities and let σ be a flat of $M(H, I)$. Then*

$$\#\text{Hom}(H/\sigma \rightarrow \star) = \sum_{\rho \geq \sigma} \#\text{PartInj}(H/\rho, I/\rho \rightarrow \star),$$

where the sum is over flats ρ of $M(H, I)$.

Proof Let G be a graph. We define

$$\text{Hom}(H \rightarrow G)[\sigma] := \{h \in \text{Hom}(H \rightarrow G) \mid \forall\{u, v\} \in \sigma : h(u) = h(v)\}.$$

Observe that $\#\text{Hom}(H/\sigma \rightarrow G) = \#\text{Hom}(H \rightarrow G)[\sigma]$. We will now partition the set $\text{Hom}(H \rightarrow G)[\sigma]$ by the inequalities that are satisfied. To this end, given some $h \in \text{Hom}(H \rightarrow G)[\sigma]$, we set

$$\rho(h) := \{\{u, v\} \in I \mid h(u) = h(v)\} \tag{3}$$

and claim that $\rho(h)$ is a flat of $M(H, I)$ which is greater than, i.e., a superset of σ . Assume for contradiction that $\rho(h)$ is not a flat. Then there exists an inequality $\{a, b\} \in I \setminus \rho(h)$ such that $\rho(h)$ and $\rho(h) \cup \{a, b\}$ induce the same connected components of $\mathbb{I}(H, I)$. Consequently, there is a path from a to b in $\mathbb{I}(H, I)$ only consisting of edges in $\rho(h)$. Inductively applying (3) thus yields $h(a) = h(b)$ and hence $\{a, b\} \in \rho(h)$ which contradicts the assumption.

This allows us to define an equivalence relation on $\text{Hom}(H \rightarrow G)[\sigma]$ by setting $h \sim \hat{h}$ if $\rho(h) = \rho(\hat{h})$. In particular, the equivalence classes are uniquely identified by flats greater or equal to σ . Furthermore, the elements of a class $[\rho]$ for $\rho \geq \sigma$ are precisely those homomorphisms $h \in \text{Hom}(H \rightarrow G)$ satisfying

$$\forall\{u, v\} \in I : h(u) = h(v) \Leftrightarrow \{u, v\} \in \rho,$$

and consequently

$$\#[[\rho]] = \#\text{PartInj}(H/\rho, I/\rho \rightarrow G).$$

It follows that

$$\begin{aligned} \#\text{Hom}(H/\sigma \rightarrow G) &= \#\text{Hom}(H \rightarrow G)[\sigma] \\ &= \sum_{\rho \geq \sigma} \#[[\rho]] \\ &= \sum_{\rho \geq \sigma} \#\text{PartInj}(H/\rho, I/\rho \rightarrow G), \end{aligned}$$

which concludes the proof. □

We are now ready to define the quantum graph for Theorem 11. To this end, let (H, I) be a graph with inequalities and let $L = L(H, I)$. We set

$$Q[H, I] := \sum_{F \in \text{Spasm}(H, I)} \lambda_F \cdot F,$$

where

$$\lambda_F := \sum_{\substack{\rho \in L \\ F \simeq H/\rho}} \mu_L(\emptyset, \rho). \quad (4)$$

Corollary 3 $\#Hom(Q[H, I] \rightarrow \star) = \#PartInj(H, I \rightarrow \star)$.

Proof Invoking Möbius inversion (Theorem 9) on the zeta transformation given by Lemma 1 yields

$$\#PartInj(H, I \rightarrow \star) = \sum_{\rho \geq \emptyset} \mu_L(\emptyset, \rho) \cdot \#Hom(H/\rho \rightarrow \star).$$

The claim then follows by deleting terms for which H/ρ contains a self-loop and collecting for isomorphic graphs afterwards. In particular, there exists a flat ρ for every spasm F such that $F \simeq H/\rho$: Recall that spasms of (H, I) are defined in such a way that they are isomorphic to $H/\hat{\rho}$ for some subset $\hat{\rho}$ of I that is not necessarily a flat. However, we can take the closure $\rho := \text{cl}(\hat{\rho})$ of $\hat{\rho}$, that is, we add elements of I to $\hat{\rho}$ as long as the induced connected components in $\mathbb{I}(H, I)$ do not change. We conclude by the fact that the closure $\rho = \text{cl}(\hat{\rho})$ is a flat by definition and $H/\rho = H/\hat{\rho}$. \square

What follows is one of the main structural insights of this work. We show that all spasms of (H, I) are constituents of $Q[H, I]$. Note that this is far from obvious as there might be many flats ρ for which H/ρ is isomorphic to the same spasm F . Hence λ_F is the sum of different values of the Möbius function that might, a priori, differ in sign and cancel out to zero. The following lemma shows that we rely on Theorem 10 to prove that such cancellations are impossible.

Lemma 2 *Let F be a spasm of (H, I) and let λ_F be as in (4). Then we have that*

$$\text{sign}(\lambda_F) = (-1)^{|V(H)| - |V(F)|}.$$

We provide an illustration of the previous lemma in Fig. 3.

Proof Let $L = L(H, I)$ and let $\rho \in L$ be a flat such that F is isomorphic to H/ρ . It suffices to prove that

$$\text{sign}(\mu_L(\emptyset, \rho)) = (-1)^{|V(H)| - |V(F)|}.$$

First, we observe that $|V(H/\rho)| = |V(F)|$ as otherwise H/ρ and F would not be isomorphic. Now recall that the graph H/ρ is obtained from H by identifying every pair of vertices u and v for which $\{u, v\} \in \rho$. Consequently, the number of connected components of the induced subgraph of $\mathbb{I}(H, I)$ with edges ρ is equal to $|V(H/\rho)|$. As furthermore $\mathbb{I}(H, I)$ is by definition the underlying graph of the graphic matroid $M(H, I)$, we have that

$$\text{rk}(\rho) = |V(\mathbb{I}(H, I))| - |V(H/\rho)| = |V(H)| - |V(F)|.$$

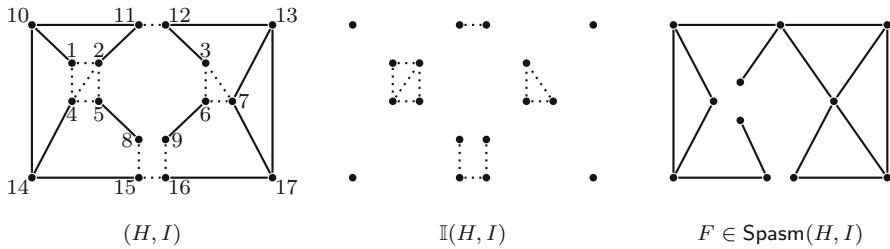


Fig. 3 Illustration of Lemma 2. Despite the fact that there is more than one flat ρ of the matroid $M(H, I)$ for which H/ρ is isomorphic to F , Lemma 2 guarantees that $\lambda_F \neq 0$. Examples are given by the flats $\rho_1 := \{\{11, 12\}, \{1, 4\}, \{3, 6\}, \{6, 7\}, \{7, 3\}, \{8, 15\}, \{9, 16\}\}$, as well as by $\rho_2 := \{\{9, 16\}, \{15, 16\}, \{8, 15\}, \{1, 2\}, \{1, 4\}, \{2, 4\}, \{2, 5\}, \{4, 5\}\}$. Now both, ρ_1 and ρ_2 have rank 6, which equals $|V(H)| - |V(F)| = 17 - 11$. Hence $\text{sign}(\lambda_F) = (-1)^6 = 1$. In particular, the number of vertices of the graph $F \simeq H/\rho_1 \simeq H/\rho_2$ equals the number of connected components induced by ρ_1 and ρ_2 in the inequality graph $\mathbb{I}(H, I)$

Now we can invoke Theorem 10 and obtain that

$$\text{sign}(\mu_L(\emptyset, \rho)) = (-1)^{\text{rk}(\rho)} = (-1)^{|V(H)| - |V(F)|} .$$

□

Proof of Theorem 11 Let $Q[H, I]$ be as in (4). The claim follows then by Corollary 3 and Lemma 2. □

One might wonder, whether the tractable cases in Theorem 11 are not only fixed-parameter tractable, but also polynomial-time solvable. We have seen that this is the case if we count homomorphisms without inequalities (Theorem 5) or homomorphisms with all inequalities, that is, subgraph embeddings [17]. However, we will show that for partial injectivity constraints, there are fixed-parameter tractable cases that are not polynomial-time solvable unless $P = \#P$. We will encounter an example for this phenomenon in the subsequent section, in which we consider the problem of counting locally injective homomorphisms.

4 Locally Injective Homomorphisms

A homomorphism h from H to G is *locally injective* if for every $v \in V(H)$ it holds that $h|_{N(v)}$ is injective. We denote $\text{Li-Hom}(H \rightarrow G)$ as the set of all locally injective homomorphisms from H to G and given a class of graphs \mathcal{H} , we define $\#\text{Li-Hom}(\mathcal{H})$ as the problem of, given graphs $H \in \mathcal{H}$ and G , computing $\#\text{Li-Hom}(H \rightarrow G)$; the parameter is $|V(H)|$.

It is immediate to express local injectivity as partial injectivity.

Fact 2 *Let H be a graph and $I := \{\{u, v\} \mid \exists w \in V(H) : u, v \in N(w)\}$. Then the following holds true for every graph G :*

$$\text{Li-Hom}(H \rightarrow G) = \text{PartInj}(H, I \rightarrow G) .$$

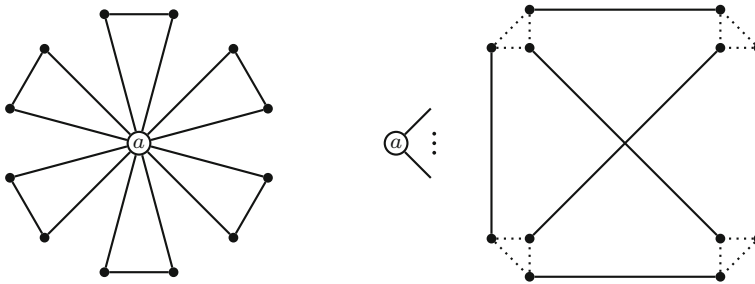


Fig. 4 Illustration of the windmill graph W_6 and a spasm of (W_6, I) that contains the clique of size 4 as a minor: The dotted lines in the right picture constitute a flat of the inequality graph w.r.t. I as every pair of vertices of the matching has the common neighbour a . If the graph is contracted along this flat and afterwards a is removed, we obtain the desired K_4

Consequently, the classification for counting partially injective homomorphisms completely resolves the complexity of counting locally injective homomorphisms:

Proof of Corollary 1 Holds by Fact 2 and Corollary 2. \square

In particular, we are able to transfer the conditional lower bound under ETH as well. We provide an example using *windmill graphs*. To this end, given $n \in \mathbb{N}$, we define W_n as the graph obtained from the matching M_n of size n by adding a new vertex a and connecting it to all vertices of the matching.

Corollary 4 *Let \mathcal{W} be the set of all windmill graphs W_n for $n \in \mathbb{N}$. Then $\#\text{LI-HOM}(\mathcal{W})$ is $\#\text{W}[1]$ -complete and cannot be solved in time*

$$g(|V(W_k)|) \cdot |V(G)|^{o(k/\log k)},$$

for any function g , unless ETH fails.

Proof Let F be an arbitrary graph with k edges and let \hat{F} be the graph obtained from F by adding a new vertex a that is made adjacent to all vertices of F . We claim that \hat{F} is a spasm of (W_k, I) where I is defined as in Fact 2. To see this, we observe that the edges of M_k in W_k can be arranged according to F and then be identified according to the inequalities between every pair of vertices of the matching; consult Fig. 4 for an illustration. Therefore F is the minor of some spasm of (W_k, I) and consequently, the set of spasms of (W_k, I) for all $k \in \mathbb{N}$ contains all graphs as minors and is therefore of unbounded treewidth. $\#\text{W}[1]$ -completeness follows hence by Corollary 2.

For the conditional lower bound, we rely on the fact that there exists an infinite and recursively enumerable class \mathcal{E} of graphs such that every graph $H \in \mathcal{E}$ has treewidth $\Theta(|V(H)|)$ and is sparse, i.e., $|E(H)| \in \Theta(|V(H)|)$. In fact, every explicit construction of d -regular expanders has this property (cf. [30] and [18, Exercise 7.34]). By the previous observation, we have that each graph $H \in \mathcal{E}$ is a minor of a spasm of $W_{|E(H)|}$. Now assume there exists a function g and an algorithm that solves $\#\text{LI-HOM}(\mathcal{W})$ in time

$$g(|V(W_k)|) \cdot |V(G)|^{o(k/\log k)}.$$

By Fact 2, Theorem 11 and complexity monotonicity (Theorem 8), this algorithm can be used to solve the problem #HOM(\mathcal{E}) in time

$$\hat{g}(|V(H)|) \cdot |V(G)|^{o(k/\log k)} = \hat{g}(|V(H)|) \cdot |V(G)|^{o(\text{tw}(H)/\log(\text{tw}(H)))},$$

for some function \hat{g} only depending on g , which violates ETH by Theorem 7. □

On the other hand, we obtain the following tractability result.

Corollary 5 (Second part of Theorem 3, restated) *Let \mathcal{T} be the set of all trees. Then #LI- HOM(\mathcal{T}) is fixed-parameter tractable.*

Proof Let T be a tree and let I as in Fact 2. Then every spasm of (T, I) is a tree as well and has hence treewidth 1. Fixed-parameter tractability thus follows by Fact 2 and Corollary 2. □

However, #LI- HOM(\mathcal{T}) is an exemplary instance of the problem of counting locally injective homomorphisms that is most likely not solvable in polynomial time.

Lemma 3 (First part of Theorem 3, restated) *#LI- HOM(\mathcal{T}) is #P-hard.*

The class #P is the classical counting analogue of NP. A formal definition, including a rough introduction to classical counting, and the proof of the above lemma can be found in the following subsection. Roughly speaking, the idea is to reduce from subgraph counting on trees, which is shown to be #P-hard as well.

4.1 Counting Subtrees of a Tree

We start by giving an introduction to classical counting complexity which was established by Valiant in his seminal work about the complexity of computing the permanent [56]. Recall that a (non-parameterized) counting problem is a function $F : \{0, 1\}^* \rightarrow \mathbb{N}$. The class of all counting problems solvable in polynomial time is called FP. On the other hand, the notion of intractability is #P-hardness. #P is the class of all counting problems that are polynomial-time (many-one) reducible⁸ to #SAT, the problem of computing the number of satisfying assignments of a given CNF formula. A counting problem F is #P-hard if there exists a polynomial-time Turing-reduction from #SAT to F , that is, an algorithm with oracle access to F that solves #SAT in polynomial time.

For the proof of Lemma 3, we will first show #P-hardness of the following intermediate problem: Given two trees T_1, T_2 , compute the number #Sub($T_1 \rightarrow T_2$) of subtrees of T_2 that are isomorphic to T_1 . We call this problem #SUB(\mathcal{T}, \mathcal{T}).

Lemma 4 (Theorem 4, restated) *#SUB(\mathcal{T}, \mathcal{T}) is #P-hard.*

⁸ (Many-one) reductions in counting complexity differ slightly from many-one reductions in the decision world. However, for the purpose of this work we only need Turing-reductions. We recommend Chapter 6.2 in [29] to the interested reader.

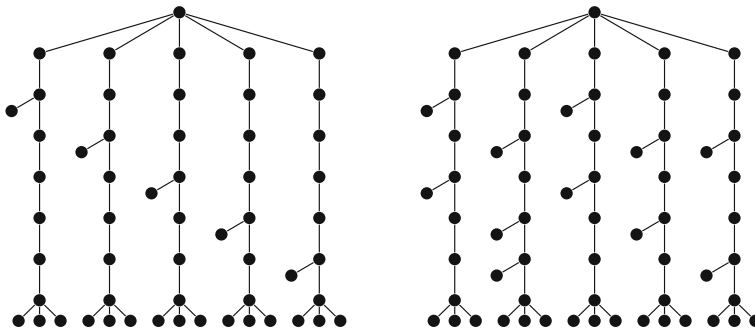


Fig. 5 Trees T_{id_5} (left) and T_B (right)

Related results are #P-hardness for counting *all* subtrees of a given graph due to Jerrum [35] or even counting *all* subtrees of a given tree due to Goldberg and Jerrum [28]. As the number of non-isomorphic trees with n vertices is not bounded by a polynomial in n , we do not know how to reduce directly from those problems. Instead we use a construction quite similar to the “skeleton” graph in the construction of Goldberg and Jerrum [28] to reduce from the problem of computing the permanent: Recall that a matrix with the same number of rows and columns is called *quadratic*. Given a quadratic matrix A with elements $(a_{i,j})_{i,j \in [n]}$ the *permanent* of A is defined by

$$\text{perm}(A) := \sum_{\pi \in \text{Sym}_n} \prod_{i=1}^n a_{i,\pi(i)},$$

where Sym_n is the symmetric group with n elements.

Theorem 12 ([56]) *Computing the permanent is #P-hard even when restricted to matrices with entries in $\{0, 1\}$.*

Proof of Lemma 4 We reduce from computing the permanent of matrices with entries from $\{0, 1\}$. Given a quadratic matrix A of size n , we construct a tree T_A as follows:

1. For every entry $a_{i,j}$ we create a vertex $v_{i,j}$ and add edges $\{v_{i,j}, v_{i+1,j}\}$ for every $i \in [n - 1]$ and every $j \in [n]$.
2. Whenever $a_{i,j} = 1$ we create a vertex $b_{i,j}$ and add edges $\{b_{i,j}, v_{i,j}\}$.
3. For every column c_j we create a vertices u_j, w_j, x_j, y_j, z_j and add the edges $\{u_j, v_{1,j}\}, \{v_{n,j}, w_j\}, \{w_j, x_j\}, \{w_j, y_j\}$ and $\{w_j, z_j\}$.
4. Finally, we create a vertex r and add edges $\{r, u_j\}$ for all $j \in [n]$. In what follows, we call r the root.

We give an example in Fig. 5 for the matrix

$$B := \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

We claim that for all quadratic matrices A of size $n \geq 5$ with entries from the set $\{0, 1\}$ the following holds true

$$\text{perm}(A) = \#\text{Sub}(T_{\text{id}_n} \rightarrow T_A),$$

where id_n is the quadratic matrix of size n with 1s on the diagonal and 0s everywhere else. In the following we write v for a vertex in T_A and v' for a vertex in T_{id_n} . To prove the claim we first observe that whenever a subtree of T_A is isomorphic to T_{id_n} , the root r' of T_{id_n} has to be mapped to the root r of T_A by the isomorphism as the roots are the only vertices with degree n (which is why we needed $n \geq 5$ as every other vertex has degree ≤ 4). It follows that the vertices u'_1, \dots, u'_n of T_{id_n} are mapped to the vertices u_1, \dots, u_n of T_A which induces a permutation on n elements, that is, an element $\pi \in \text{Sym}_n$.

We will now partition the subtrees of T_A isomorphic to T_{id_n} by those permutations and write $\#\text{Sub}(T_{\text{id}_n} \rightarrow T_A)[\pi]$ for the number of subtrees that induce π . Now fix π and consider a subtree that induces π . It holds that for all $j \in [n]$ the vertex w'_j has to be mapped to $w_{\pi(j)}$ as those are the only vertices with degree exactly 4 and furthermore, the vertices x'_j, y'_j, z'_j have to be mapped to $x_{\pi(j)}, y_{\pi(j)}, z_{\pi(j)}$ (possibly permuted but the subtree of T_A is the same). Now $v'_{i,i}$ is adjacent to $b'_{i,i}$ for each $i \in [n]$ and thus $v_{i,\pi(i)}$ has to be adjacent to $b_{i,\pi(i)}$, that is $a_{i,\pi(i)} = 1$. If this is not the case then there is no subtree that induces partition π . Furthermore there is at most one subtree isomorphic to T_{id_n} inducing π because the image is enforced by the vertices r', w'_j and $v'_{i,i}$ for all $i, j \in [n]$. Consequently

$$\#\text{Sub}(T_{\text{id}_n} \rightarrow T_A)[\pi] = 1,$$

if for all $i \in [n]$ it holds that $a_{i,\pi(i)} = 1$, and otherwise

$$\#\text{Sub}(T_{\text{id}_n} \rightarrow T_A)[\pi] = 0.$$

Hence $\#\text{Sub}(T_{\text{id}_n} \rightarrow T_A)[\pi] = \prod_{i=1}^n a_{i,\pi(i)}$ and therefore

$$\text{perm}(A) = \sum_{\pi \in \text{Sym}_n} \prod_{i=1}^n a_{i,\pi(i)} = \sum_{\pi \in S_n} \#\text{Sub}(T_{\text{id}_n} \rightarrow T_A)[\pi] = \#\text{Sub}(T_{\text{id}_n} \rightarrow T_A).$$

Now the reduction works as follows: If the input matrix A has size ≤ 4 we brute-force the output and otherwise we compute $\#\text{Sub}(T_{\text{id}_n} \rightarrow T_A)$ using the oracle for $\#\text{SUB}(T, T)$. □

The proof of Lemma 3 relies on the fact that locally injective homomorphisms from a tree to a tree are embeddings.

Proof of Lemma 3 It is a well-known fact that a locally injective homomorphism h from a tree T_1 to a tree T_2 is (fully) injective. To see this, assume that there are vertices

v and u in T_1 that are mapped to the same vertex in T_2 . As T_1 is a tree there exists exactly one path

$$v = w_0, w_1, \dots, w_\ell, w_{\ell+1} = u$$

between v and u in T_1 . It holds that $\ell \geq 1$ as otherwise v and u would be adjacent and hence $h(u) = h(v)$ would have a self-loop in T_2 which is impossible. As h is locally injective we have that $h(v) \neq h(w_2)$, hence $u \neq w_2$, and as h is edge-preserving there are edges $\{h(v), h(w_1)\}$ and $\{h(w_1), h(w_2)\}$ and a path from $h(w_2)$ to $h(w_{\ell+1}) = h(u) = h(v)$ in T_2 . This induces a cycle and contradicts the fact that T_2 is a tree.

Therefore $\#\text{Emb}(T_1 \rightarrow T_2) = \#\text{Li-Hom}(T_1 \rightarrow T_2)$. By Fact 1, we have that for all H and G the following is true

$$\#\text{Sub}(H \rightarrow G) = \frac{\#\text{Emb}(H \rightarrow G)}{\#\text{Aut}(H)}.$$

If H is a tree then H is planar and thus $\#\text{Aut}(H)$ can be computed in polynomial time [32,43]. Therefore $\#\text{P-hardness}$ of $\#\text{LI-HOM}(\mathcal{T})$ follows by reducing from $\#\text{SUB}(\mathcal{T}, \mathcal{T})$, which is hard by Lemma 4: Given trees T_1 and T_2 we obtain $\#\text{Li-Hom}(T_1 \rightarrow T_2)$ by querying the oracle and compute $\#\text{Aut}(T_1)$ in polynomial time. Finally, we output

$$\frac{\#\text{Li-Hom}(T_1 \rightarrow T_2)}{\#\text{Aut}(T_1)} = \frac{\#\text{Emb}(T_1 \rightarrow T_2)}{\#\text{Aut}(T_1)} = \#\text{Sub}(T_1 \rightarrow T_2).$$

□

5 Quantum Graphs with Inequalities

We will now go one step further and consider linear combinations of partially injective homomorphisms. In particular, this allows for generalizing subgraph counting in the sense, that we do not only wish to count subgraphs isomorphic to a single graph H , but rather to count subgraphs that are isomorphic to some graph in a given set of graphs. An example for the latter is the problem of counting acyclic subgraphs of size k . We emphasize that the subsequent results for subgraphs have already been observed by Curticapean, Dell and Marx [16]. We extend their results to partial injectivity constraints.

It will be convenient to express the problems in this section by *quantum graphs with inequalities*, which are defined to be formal linear combinations of graphs with inequalities of finite support. We write

$$\mathcal{I} = \sum_{(H,I)} \lambda_{(H,I)} \cdot (H, I),$$

where the sum is over all (isomorphism types of) graphs and inequality constraints. In particular, \mathcal{I} might contain the constituents (H, I) and (H, I') for $I \neq I'$. Counting partially injective homomorphisms extends to \mathcal{I} linearly.

$$\#PartInj(\mathcal{I} \rightarrow \star) := \sum_{(H,I)} \lambda_{(H,I)} \cdot \#PartInj(H, I \rightarrow \star).$$

If all constituents (H, I) of \mathcal{I} satisfy that I is the full injectivity constraint, then $\#PartInj(\mathcal{I} \rightarrow \star)$ computes a linear combination of subgraph embeddings, given by

$$\#PartInj(\mathcal{I} \rightarrow \star) = \sum_H \lambda_H \cdot \#Emb(H \rightarrow \star).$$

where the sum is over all graphs and $\lambda_H = \lambda_{(H,I)}$ if (H, I) is a constituent of \mathcal{I} and $\lambda_H = 0$ otherwise.

We point out that $\#PartInj(\mathcal{I} \rightarrow \star)$ does not allow for complexity monotonicity in general, even if there are no negative coefficients. To see this, recall that counting homomorphisms is the zeta transformation of counting subgraph embeddings (invoke Lemma 1 with I being the set of all inequalities). In particular, the following holds true for the k -matching M_k :

$$\#Hom(M_k \rightarrow \star) = \sum_{\rho} \#Emb(M_k/\rho \rightarrow \star).$$

Consequently, assuming that $k = \binom{r}{2}$, this constitutes a linear combination of subgraph embeddings, such that the r -clique is contained with a non-zero coefficient. However, as matchings have treewidth 1, $\#Hom(M_k \rightarrow \star)$ can be computed in time $\text{poly}(k) \cdot n^{O(1)}$ by Theorem 6, whereas cliques of size r cannot be counted in time

$$\text{poly}(r) \cdot n^{O(1)} = \text{poly}(k) \cdot n^{O(1)},$$

unless ETH fails [12,13].

Now observe that quantum graphs with inequalities yield linear combinations of partially injective homomorphisms and the latter are again linear combinations of homomorphisms. Consequently, it is possible to express the number of partially injective homomorphisms from a quantum graph with inequalities as the number of homomorphisms from a quantum graph without inequalities; recall the definition of $Q[H, I]$ in (4):

$$\#PartInj(\mathcal{I} \rightarrow \star) \tag{5}$$

$$= \sum_{(H,I)} \lambda_{(H,I)} \cdot \#PartInj(H, I \rightarrow \star) \tag{6}$$

$$= \sum_{(H,I)} \lambda_{(H,I)} \cdot \#Hom(Q[H, I] \rightarrow \star) \tag{7}$$

$$= \sum_{(H,I)} \lambda_{(H,I)} \cdot \sum_{F \in \text{Spasm}(H,I)} \left(\sum_{\substack{\rho \in L(H,I) \\ F \simeq H/\rho}} \mu_{L(H,I)}(\emptyset, \rho) \right) \# \text{Hom}(F \rightarrow \star), \tag{8}$$

where (6) is the definition of $\# \text{PartInj}(\mathcal{I} \rightarrow \star)$, (7) is Corollary 3 and (8) is the definition of $Q[H, I]$. This induces the following quantum graph $Q[\mathcal{I}]$ (without inequalities) which given by

$$Q[\mathcal{I}] := \sum_F v_F \cdot F,$$

where

$$v_F := \sum_{\substack{(H,I) \\ F \in \text{Spasm}(H,I)}} \lambda_{(H,I)} \left(\sum_{\substack{\rho \in L(H,I) \\ F \simeq H/\rho}} \mu_{L(H,I)}(\emptyset, \rho) \right).$$

We will now provide a criterion that, if satisfied, allows us to give the support of $Q[\mathcal{I}]$ explicitly.

Theorem 13 *Let \mathcal{I} be a quantum graph with inequalities. Then*

$$\# \text{PartInj}(\mathcal{I} \rightarrow \star) = \# \text{Hom}(Q[\mathcal{I}] \rightarrow \star).$$

If, additionally, no coefficient of \mathcal{I} is negative and every pair $(H, I), (H', I')$ of constituents of \mathcal{I} satisfies $|V(H)| = |V(H')|$, then

$$\text{supp}(Q[\mathcal{I}]) = \bigcup_{(H,I) \in \text{supp}(\mathcal{I})} \text{Spasm}(H, I).$$

Proof $\# \text{PartInj}(\mathcal{I} \rightarrow \star) = \# \text{Hom}(Q[\mathcal{I}] \rightarrow \star)$ holds by (8) and collecting for isomorphic terms. For the second claim, we first observe that

$$\text{supp}(Q[\mathcal{I}]) \subseteq \bigcup_{(H,I) \in \text{supp}(\mathcal{I})} \text{Spasm}(H, I)$$

by the definition of the coefficients v_F . Now let $F \in \text{Spasm}(H, I)$ for some constituent (H, I) of \mathcal{I} and let $k = |V(H)|$. Then we have that

$$\text{sign} \left(\sum_{\substack{\rho \in L(H,I) \\ F \simeq H/\rho}} \mu_{L(H,I)}(\emptyset, \rho) \right) = (-1)^{k - |V(F)|}$$

by Lemma 2. Consequently

$$\text{sign} \left(\lambda_{(H,I)} \sum_{\substack{\rho \in L(H,I) \\ F \simeq H/\rho}} \mu_{L(H,I)}(\emptyset, \rho) \right) = (-1)^{k-|V(F)|}$$

as $\lambda_{(H,I)} \geq 0$ by assumption and $\lambda_{(H,I)} \neq 0$ as (H, I) is a constituent of \mathcal{I} . Thus $\text{sign}(v_F) = (-1)^{k-|V(F)|}$ and hence $v_F \neq 0$. □

Note that it was crucial in the above proof that all graphs H for which (H, I) is a constituent of \mathcal{I} have the same number of vertices. If this is not the case, then we cannot guarantee that the terms

$$\text{sign} \left(\lambda_{(H,I)} \sum_{\substack{\rho \in L(H,I) \\ F \simeq H/\rho}} \mu_{L(H,I)}(\emptyset, \rho) \right) = (-1)^{|V(H)-|V(F)|}$$

are equal for different constituents and hence there might be cancellations. Going back to the example of computing $\#\text{Hom}(M_k \rightarrow \star)$, we observe that the expression as a linear combination of subgraph embeddings does indeed contain constituents with different numbers of vertices. In other words, those cancellations *must* occur in the general case (unless ETH fails).

We furthermore point out that the proof of Theorem 13 actually shows the following, more general statement.

Remark 1 The second claim of Theorem 13 holds true even if we only require that all non-zero coefficients have the same sign and that the number of vertices of constituents of \mathcal{I} have the same parity.

Now, arguably, Theorem 13 might seem artificial at first glance. Let us hence provide a concrete application which shows its utility. We define the problem $\#\text{VERTEXFORESTS}$ as follows: Given a graph G and a positive integer k , the task is to compute the number of acyclic subgraphs with k vertices⁹ in G . The problem is parameterized by k . The following hardness result follows implicitly from [16]. We include a proof only for illustrating an application of Theorem 13.

Lemma 5 $\#\text{VERTEXFORESTS}$ is $\#W[1]$ -complete.

Proof Let \mathcal{F}_k be the set of all acyclic graphs with k vertices. Furthermore, given some graph F , we write $\text{full}(F)$ for the set of all possible inequalities between vertices of

⁹ We emphasize in the name of the problem, that we are interested in subgraphs with k vertices, not with k edges. The latter is also known to be hard, but the proof is more involved and does not use quantum graphs [4].

F . Now the number of acyclic subgraphs of size k of a graph G equals

$$\begin{aligned} \sum_{F \in \mathcal{F}_k} \#\text{Sub}(F \rightarrow G) &= \sum_{F \in \mathcal{F}_k} \#\text{Aut}(F)^{-1} \cdot \#\text{Emb}(F \rightarrow G) \\ &= \sum_{F \in \mathcal{F}_k} \#\text{Aut}(F)^{-1} \cdot \#\text{PartInj}(F, \text{full}(F) \rightarrow G) \\ &= \#\text{PartInj}(\mathcal{I} \rightarrow G), \end{aligned}$$

where

$$\mathcal{I} = \sum_{\substack{(F, \text{full}(F)) \\ F \in \mathcal{F}_k}} \#\text{Aut}(F)^{-1} \cdot (F, \text{full}(F))$$

is a quantum graph with inequalities. As, by definition, all graphs in \mathcal{F}_k have k vertices and the terms $\#\text{Aut}(F)^{-1}$ are all greater than zero, we can invoke Theorem 13 and obtain the quantum graph $Q[\mathcal{I}]$ such that $\sum_{F \in \mathcal{F}_k} \#\text{Sub}(F \rightarrow G) = \#\text{Hom}(Q[\mathcal{I}] \rightarrow G)$ and

$$\text{supp}(Q[\mathcal{I}]) = \bigcup_{F \in \mathcal{F}_k} \text{Spasm}(F, \text{full}(F)).$$

Now, similarly to the proof of Corollary 2, we invoke complexity monotonicity and obtain that $\#\text{VERTEXFORESTS}$ is interreducible with $\#\text{HOM}(Q)$ with respect to parameterized Turing reductions, where

$$Q = \left\{ H \in \text{Spasm}(F, \text{full}(F)) \mid F \in \bigcup_{k \in \mathbb{N}} \mathcal{F}_k \right\}.$$

As for every $k \in \mathbb{N}$, \mathcal{F}_{2k} contains the matching M_k , we conclude that Q contains for every k all (connected) graphs with k edges and is therefore of unbounded treewidth. The lemma hence holds by Theorem 5. \square

Remark 2 The problem $\#\text{TREES}$ of counting connected acyclic subgraphs with k vertices can be proved $\#\text{W}[1]$ -complete similarly, as shown in [16]. We emphasize that the framework of quantum graphs significantly simplifies the original $\#\text{W}[1]$ -completeness proof of $\#\text{TREES}$ [4].

6 Conclusions and Future Work

We established an exhaustive classification of the parameterized complexity of counting partially injective homomorphisms and have thus fully understood the complexity of counting answers to equi-join queries with inequalities, subsuming not only the prior

classification of the subgraph counting problem, but also inducing a novel dichotomy for counting locally injective homomorphisms.

Building upon the works of Chen and Mengel [11] and Curticapean, Dell and Marx [16] our proof illustrates once more the potential of the framework of complexity monotonicity for the field of parameterized counting. In particular, it allows a canonical generalization to logical structures of bounded arity such as equi-join queries and, even further, to conjunctive queries (implicitly in [11], explicitly in [20]), which, together with our result on quantum graphs with inequalities, might find further applications for first-order model counting problems, especially in the context of relational databases.

Acknowledgements The author is very grateful to Holger Dell and Radu Curticapean for fruitful discussions, as well as to an anonymous reviewer for pointing out the applicability of the framework to equi-join queries. Furthermore the author thanks Cornelius Brand for saying “Tutte Polynomial” every once in a while, and Philip Wellnitz for providing valuable feedback on early drafts of the full version of this work.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. van Aardenne-Ehrenfest, T., de Bruijn, N.G.: Circuits and trees in oriented linear graphs. *Simon Stevin: Wis- en Natuurkundig Tijdschrift* **28**, 203–217 (1951)
2. Backens, M.: A Complete Dichotomy for Complex-Valued Holant^c. In: 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9–13, 2018, Prague, Czech Republic, pp. 12:1–12:14 (2018). <https://doi.org/10.4230/LIPIcs.ICALP.2018.12>
3. Barvinok, A.I.: *Combinatorics and Complexity of Partition Functions, Algorithms and combinatorics*, vol. 30. Springer (2016). <https://doi.org/10.1007/978-3-319-51829-9>
4. Brand, C., Roth, M.: Parameterized Counting of Trees, Forests and Matroid Bases. In: *Computer Science - Theory and Applications - 12th International Computer Science Symposium in Russia, CSR 2017, Kazan, Russia, June 8–12, 2017, Proceedings*, pp. 85–98 (2017). https://doi.org/10.1007/978-3-319-58747-9_10
5. Bulatov, A.A.: A Dichotomy Theorem for Nonuniform CSPs. In: 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15–17, 2017, pp. 319–330 (2017). <https://doi.org/10.1109/FOCS.2017.37>
6. Cai, J., Fu, Z., Guo, H., Williams, T.: A Holant Dichotomy: Is the FKT Algorithm Universal? In: IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17–20 October, 2015, pp. 1259–1276 (2015). <https://doi.org/10.1109/FOCS.2015.81>
7. Cai, J., Huang, S., Lu, P.: From Holant to #CSP and Back: Dichotomy for Holant^c Problems. *Algorithmica* **64**(3), 511–533 (2012). <https://doi.org/10.1007/s00453-012-9626-6>
8. Cai, J., Lu, P.: Holographic algorithms: From art to science. *J. Comput. Syst. Sci.* **77**(1), 41–61 (2011). <https://doi.org/10.1016/j.jcss.2010.06.005>
9. Cai, J., Lu, P., Xia, M.: Holographic Algorithms with Matchgates Capture Precisely Tractable Planar #CSP. *SIAM J. Comput.* **46**(3), 853–889 (2017). <https://doi.org/10.1137/16M1073984>
10. Cai, J., Lu, P., Xia, M.: Dichotomy for Real Holant^c Problems. In: *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7–10, 2018*, pp. 1802–1821 (2018). <https://doi.org/10.1137/1.9781611975031.118>

11. Chen, H., Mengel, S.: Counting Answers to Existential Positive Queries: A Complexity Classification. In: Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016, pp. 315–326 (2016). <https://doi.org/10.1145/2902251.2902279>
12. Chen, J., Chor, B., Fellows, M., Huang, X., Juedes, D.W., Kanj, I.A., Xia, G.: Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput.* **201**(2), 216–231 (2005). <https://doi.org/10.1016/j.ic.2005.05.001>
13. Chen, J., Huang, X., Kanj, I.A., Xia, G.: Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.* **72**(8), 1346–1367 (2006). <https://doi.org/10.1016/j.jcss.2006.04.007>
14. Curticapean, R.: Counting Matchings of Size k is $W[1]$ -Hard. In: Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I, pp. 352–363 (2013). https://doi.org/10.1007/978-3-642-39206-1_30
15. Curticapean, R.: The simple, little and slow things count: On parameterized counting complexity. Ph.D. thesis, Saarland University (2015). <http://scidok.sulb.uni-saarland.de/volltexte/2015/6217/>
16. Curticapean, R., Dell, H., Marx, D.: Homomorphisms are a good basis for counting small subgraphs. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017, pp. 210–223 (2017). <https://doi.org/10.1145/3055399.3055502>
17. Curticapean, R., Marx, D.: Complexity of Counting Subgraphs: Only the Boundedness of the Vertex-Cover Number Counts. In: 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014, pp. 130–139 (2014). <https://doi.org/10.1109/FOCS.2014.22>
18. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015). <https://doi.org/10.1007/978-3-319-21275-3>
19. Dalmau, V., Jonsson, P.: The complexity of counting homomorphisms seen from the other side. *Theor. Comput. Sci.* **329**(1–3), 315–323 (2004). <https://doi.org/10.1016/j.tcs.2004.08.008>
20. Dell, H., Roth, M., Wellnitz, P.: Counting Answers to Existential Questions. In: 46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece., pp. 113:1–113:15 (2019). <https://doi.org/10.4230/LIPIcs.ICALP.2019.113>
21. Dell, H., Roth, M., Wellnitz, P.: Counting Answers to Existential Questions. CoRR abs/1902.04960 (2019)
22. Díaz, J., Serna, M.J., Thilikos, D.M.: Counting H -colorings of partial k -trees. *Theor. Comput. Sci.* **281**(1–2), 291–309 (2002). [https://doi.org/10.1016/S0304-3975\(02\)00017-8](https://doi.org/10.1016/S0304-3975(02)00017-8)
23. Dyer, M.E., Goldberg, L.A., Jerrum, M.: An approximation trichotomy for Boolean $\#CSP$. *J. Comput. Syst. Sci.* **76**(3–4), 267–277 (2010). <https://doi.org/10.1016/j.jcss.2009.08.003>
24. Feder, T., Vardi, M.Y.: The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. Comput.* **28**(1), 57–104 (1998). <https://doi.org/10.1137/S0097539794266766>
25. Fiala, J., Kratochvíl, J.: Locally constrained graph homomorphisms - structure, complexity, and applications. *Computer Science Review* **2**(2), 97–111 (2008). <https://doi.org/10.1016/j.cosrev.2008.06.001>
26. Flum, J., Grohe, M.: The Parameterized Complexity of Counting Problems. *SIAM J. Comput.* **33**(4), 892–922 (2004). <https://doi.org/10.1137/S0097539703427203>
27. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series. Springer (2006). <https://doi.org/10.1007/3-540-29953-X>
28. Goldberg, L.A., Jerrum, M.: Counting Unlabelled Subtrees of a Tree is $\#P$ -complete. *LMS Journal of Computation and Mathematics* **3**, 117–124 (2000). <https://doi.org/10.1112/S1461157000000243>
29. Goldreich, O.: Computational Complexity - A Conceptual Perspective. Cambridge University Press, (2008)
30. Grohe, M., Marx, D.: On tree width, bramble size, and expansion. *J. Comb. Theory, Ser. B* **99**(1), 218–228 (2009). <https://doi.org/10.1016/j.jctb.2008.06.004>
31. Guo, H., Liao, C., Lu, P., Zhang, C.: Zeros of Holant problems: locations and algorithms. In: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019, pp. 2262–2278 (2019). <https://doi.org/10.1137/1.9781611975482.137>
32. Hopcroft, J.E., Tarjan, R.E.: A V^2 Algorithm for Determining Isomorphism of Planar Graphs. *Inf. Process. Lett.* **1**(1), 32–34 (1971). [https://doi.org/10.1016/0020-0190\(71\)90019-6](https://doi.org/10.1016/0020-0190(71)90019-6)

33. Huang, S., Lu, P.: A Dichotomy for Real Weighted Holant Problems. *Computational Complexity* **25**(1), 255–304 (2016). <https://doi.org/10.1007/s00037-015-0118-3>
34. Impagliazzo, R., Paturi, R.: On the Complexity of k-SAT. *J. Comput. Syst. Sci.* **62**(2), 367–375 (2001). <https://doi.org/10.1006/jcss.2000.1727>
35. Jerrum, M.: Counting Trees in a Graph is #P-Complete. *Inf. Process. Lett.* **51**(3), 111–116 (1994). [https://doi.org/10.1016/0020-0190\(94\)00085-9](https://doi.org/10.1016/0020-0190(94)00085-9)
36. Jerrum, M., Sinclair, A.: Approximating the Permanent. *SIAM J. Comput.* **18**(6), 1149–1178 (1989). <https://doi.org/10.1137/0218077>
37. Kasteleyn, P.W.: The statistics of dimers on a lattice: I. The number of dimer arrangements on a quadratic lattice. *Physica* **27**(12), 1209–1225 (1961). [https://doi.org/10.1016/0031-8914\(61\)90063-5](https://doi.org/10.1016/0031-8914(61)90063-5)
38. Kasteleyn, P.W.: Dimer Statistics and Phase Transitions. *Journal of Mathematical Physics* **4**(2), 287–293 (1963). <https://doi.org/10.1063/1.1703953>
39. Ladner, R.E.: On the structure of polynomial time reducibility. *J. ACM* **22**(1), 155–171 (1975). <https://doi.org/10.1145/321864.321877>
40. Lin, B.: The Parameterized Complexity of the k-Biclique Problem. *J. ACM* **65**(5), 34:1–34:23 (2018). <https://doi.org/10.1145/3212622>
41. Lovász, L.: Large Networks and Graph Limits, *Colloquium Publications*, vol. 60. American Mathematical Society (2012). <http://www.ams.org/bookstore-getitem/item=COLL-60>
42. Marx, D.: Can You Beat Treewidth? *Theory of Computing* **6**(1), 85–112 (2010). <https://doi.org/10.4086/toc.2010.v006a005>
43. Mathon, R.: A Note on the Graph Isomorphism Counting Problem. *Inf. Process. Lett.* **8**(3), 131–132 (1979). [https://doi.org/10.1016/0020-0190\(79\)90004-8](https://doi.org/10.1016/0020-0190(79)90004-8)
44. Maurer, S.B.: Matrix Generalizations of Some Theorems on Trees, Cycles and Cocycles in Graphs. *SIAM Journal on Applied Mathematics* **30**(1), 143–148 (1976). <https://doi.org/10.1137/0130017>
45. McCartin, C.: Parameterized counting problems. *Ann. Pure Appl. Logic* **138**(1–3), 147–182 (2006). <https://doi.org/10.1016/j.apal.2005.06.010>
46. Nešetřil, J.: Homomorphisms of derivative graphs. *Discrete Mathematics* **1**(3), 257–268 (1971). [https://doi.org/10.1016/0012-365X\(71\)90014-8](https://doi.org/10.1016/0012-365X(71)90014-8)
47. Olteanu, D., Závodný, J.: Size Bounds for Factorised Representations of Query Results. *ACM Trans. Database Syst.* **40**(1), 2:1–2:44 (2015). <https://doi.org/10.1145/2656335>
48. Oxley, J.G.: *Matroid theory*, 2nd edn. Oxford University Press, (2011)
49. Rota, G.C.: On the foundations of combinatorial theory I. Theory of Möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* **2**(4), 340–368 (1964)
50. Roth, M.: Counting Restricted Homomorphisms via Möbius Inversion over Matroid Lattices. In: 25th Annual European Symposium on Algorithms, ESA 2017, September 4–6, 2017, Vienna, Austria, pp. 63:1–63:14 (2017). <https://doi.org/10.4230/LIPIcs.ESA.2017.63>
51. Roth, M., Schmitt, J.: Counting induced subgraphs: A Topological Approach to #W[1]-hardness. In: 13th International Symposium on Parameterized and Exact Computation, IPEC 2018, August 20–24, 2018, Helsinki, Finland, pp. 24:1–24:14 (2018). <https://doi.org/10.4230/LIPIcs.IPEC.2018.24>
52. Schaefer, T.J.: The Complexity of Satisfiability Problems. In: Proceedings of the 10th Annual ACM Symposium on Theory of Computing, May 1–3, 1978, San Diego, California, USA, pp. 216–226 (1978). <https://doi.org/10.1145/800133.804350>
53. Stanley, R.P.: *Enumerative Combinatorics: vol. 1*. Cambridge University Press, (2011)
54. Temperley, H.N.V., Fisher, M.E.: Dimer problem in statistical mechanics—an exact result. *The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics* **6**(68), 1061–1063 (1961). <https://doi.org/10.1080/14786436108243366>
55. Vadhan, S.P.: The Complexity of Counting in Sparse, Regular, and Planar Graphs. *SIAM J. Comput.* **31**(2), 398–427 (2001). <https://doi.org/10.1137/S0097539797321602>
56. Valiant, L.G.: The Complexity of Computing the Permanent. *Theor. Comput. Sci.* **8**, 189–201 (1979). [https://doi.org/10.1016/0304-3975\(79\)90044-6](https://doi.org/10.1016/0304-3975(79)90044-6)
57. Valiant, L.G.: The Complexity of Enumeration and Reliability Problems. *SIAM J. Comput.* **8**(3), 410–421 (1979). <https://doi.org/10.1137/0208032>
58. Valiant, L.G.: Holographic Algorithms. *SIAM J. Comput.* **37**(5), 1565–1594 (2008). <https://doi.org/10.1137/070682575>
59. Welsh, D.J.: *Matroid theory*. Courier Corporation (2010)
60. Xia, M., Zhang, P., Zhao, W.: Computational complexity of counting problems on 3-regular planar graphs. *Theor. Comput. Sci.* **384**(1), 111–125 (2007). <https://doi.org/10.1016/j.tcs.2007.05.023>

61. Zhuk, D.: A Proof of CSP Dichotomy Conjecture. In: 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017, pp. 331–342 (2017). <https://doi.org/10.1109/FOCS.2017.38>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.