Counting Homomorphisms in Plain Exponential Time

Andrei A. Bulatov
School of Computing Science, Simon Fraser University, Burnaby, Canada
https://www.cs.sfu.ca/~abulatov/
abulatov@sfu.ca

Amineh Dadsetan
School of Computing Science, Simon Fraser University, Burnaby, Canada
amineh.dadsetan@gmail.com

Abstract

In the counting Graph Homomorphism problem (#GraphHom) the question is: Given graphs G, H, find the number of homomorphisms from G to H. This problem is generally #P-complete, moreover, Cygan et al. proved that unless the Exponential Time Hypothesis fails there is no algorithm that solves this problem in time $O(|V(H)|^{o(|V(G)|)})$. This, however, does not rule out the possibility that faster algorithms exist for restricted problems of this kind. Wahlström proved that #GraphHom can be solved in plain exponential time, that is, in time $O((2k+1)^{|V(G)|+|V(H)|}\text{poly}(|V(H)|, |V(G)|))$ provided H has clique width k. We generalize this result to a larger class of graphs, and also identify several other graph classes that admit a plain exponential algorithm for #GraphHom.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases graph homomorphisms, plain exponential time, clique width

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.21

Category Track A: Algorithms, Complexity and Games

Funding *Andrei A. Bulatov*: This work was supported by an NSERC Discovery grant.

1 Introduction

The Exponential Time Hypothesis (ETH) [16] essentially suggests that the Satisfiability problem does not admit an algorithm that is significantly faster than the straightforward brute force algorithm. The ETH has been widely used to obtain (conditional) lower bounds on the complexity of various problems, see [18] for a fairly recent survey. It however does not rule out nontrivial algorithms for many other hard problems.

One of such problems is the Graph Homomorphism problem (GraphHOM for short). A homomorphism from a graph G to a graph H is a mapping $\varphi\colon V(G) \to V(H)$ such that for any edge $ab \in E(G)$ the pair $\varphi(a)\varphi(b)$ is an edge of H. GraphHOM asks, given graphs G and H, whether or not there exists a homomorphism from G to H [14]. In the counting version of this problem, denoted #GraphHOM, the goal is to find the number of homomorphisms from G to H. These two problems can be solved just by checking all possible mappings from a given graph G to a given graph H, which takes time $O^*(|V(H)|^{|V(G)|})$, where O^* denotes asymptotics up to a polynomial factor. Assuming the ETH Cygan et al. [6] proved that the general GraphHom and therefore #GraphHom cannot be solved in time $O(|V(H)|^{o(|V(G)|)})$. A similar bound for the more general Constraint Satisfaction Problem (CSP) was established in [22], and some related hardness results have also been obtained in [5].

47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).
Editors: Artur Czumaj, Anuj Dawar, and Emanuela Merelli; Article No. 21; pp. 21:1–21:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In spite of these results, there are several ways to restrict GraphHom which sometimes result in a problem admitting a faster algorithm. For graph classes $\mathcal{G}, \mathcal{H}$, GraphHom$(\mathcal{G}, \mathcal{H})$ denotes the problem GraphHom in which the input graphs G, H belong to $\mathcal{G}, \mathcal{H}$, respectively. #GraphHOM can be restricted in the same way. Both problems have received much attention in their own right and as a special case of the general CSP, and much is known about their computational complexity. We will use the symbol $-$ to indicate that an input graph is not restricted. In particular, it is known that GraphHom$(-, \mathcal{H})$ is solvable in polynomial time only if every graph in $\mathcal{H}$ contains a loop or is bipartite [15]. It is also known that #GraphHom$(-, \mathcal{H})$ is solvable in polynomial time only if every graph in $\mathcal{H}$ is complete with all the loops present or is complete bipartite [9]. In the remaining cases these problems are shown to be NP- and #P-complete, respectively. Similarly, it is known that GraphHom$(\mathcal{G}, -)$ [13] and #GraphHom$(\mathcal{G}, -)$ [8] are solvable in polynomial time if and only if the class of cores of the graphs from $\mathcal{G}$ in the former case, and the class $\mathcal{G}$ itself in the latter case have bounded tree width, respectively.

Here we focus on such restrictions that give rise to problems solvable still in exponential time but much faster than brute force. Specifically, GraphHom$(\mathcal{G}, \mathcal{H})$ or #GraphHom$(\mathcal{G}, \mathcal{H})$ are said to be solvable in *plain exponential* time if there is a solution algorithm running in time $O^*(c^{|V(G)|+|V(H)|})$, where c is a constant. In this paper we study problems of the form #GraphHom$(-, \mathcal{H})$, however, clearly, all the easiness results for #GraphHom$(-, \mathcal{H})$ also hold for GraphHom$(-, \mathcal{H})$. If the problem #GraphHom$(-, \mathcal{H})$ is solvable in plain exponential time, we call the class $\mathcal{H}$ a *plain exponential class.*

Plain exponential classes of graphs have received substantial attention in the literature. The most well known such class is $\mathcal{K}$, the class of all cliques. Note that #GraphHom$(-, \mathcal{K})$ is equivalent to the #Graph Colouring problem, in which the problem is, given a graph G and a number k, to find the number of k-colourings of G. A fairly straightforward dynamic programming algorithm solves this problem in time $O^*(3^{|V(G)|})$; we outline this algorithm in Example 8. A more sophisticated algorithm [17] solves #GraphHom$(-, \mathcal{K})$ in time $O^*(2^{|V(G)|})$. If $\mathcal{H}$ is a class of graphs of tree width k then #GraphHom$(-, \mathcal{H})$ is solvable in time $O^*((k+3)^{|V(G)|})$, see [11]. For the class $\mathcal{D}_c$ of graphs of degree at most c the problem #GraphHom$(-, \mathcal{D}_c)$ can be solved in time $O^*(c^{|V(G)|})$ by a minor modification of the brute force enumeration algorithm, see Example 7. Finally, Wahlström [23] obtained probably the most general result so far on plain exponential graph classes, proving that if $\mathcal{H}$ only contains graphs of clique width at most k then #GraphHom$(-, \mathcal{H})$ can be solved in time $O^*((2k+1)^{|V(G)|+|V(H)|})$. The algorithm from [23] is also dynamic programming and uses the representation of (labeled) graphs of bounded clique width through a sequence of operations such as disjoint union, connecting vertices with certain labels, and relabeling vertices. Such sequences are called *k-expressions.*

In this paper we aim at a systematic study of plain exponential classes of graphs. As the first step we further expand the class of graphs for which plain exponential counting algorithms are possible by adding one more operation to the construction of graphs of bounded clique width. In a nutshell, the new operation expands a graph H to a graph H' in such a way that H is a retract of H', and the preimages of vertices of H are connected in a regular way. The new class of graphs includes families of graphs of unbounded clique width, for instance, hypercubes, grids, cliques with subdivided edges, and therefore is strictly larger than the class of graphs of bounded clique width. By means of this new set of operations one can define a new graph "width" measure that we call *extended clique width*, only this new measure involves two parameters rather than one. Graphs of extended clique width (k, r) can also be represented by *extended (k, r)-expressions.* Let $\mathcal{X}_{k,r}$ denote the class of graphs whose extended width parameters are at most k, r, respectively. (In this case we will say that such a graph has extended clique width *at most* (k, r).) Although in most cases in this paper the only parameter that matters is $\max\{k, r\}$, we think that further stratification is useful for a number of more precise results.

We then show that given an arbitrary graph G, a graph H of extended clique width (k, r), and an extended (k, r)-expression Φ representing H, the number $\mathsf{hom}(G, H)$ of homomorphisms from G to H can be found in time $O^*((2\max(k, r) + 1)^{2|V(G)|})$. Similar to [23], the algorithm is dynamic programming and iteratively computes numbers $\mathsf{hom}(G', H')$, where G' is an induced subgraph of G and H' is a graph represented by a subexpression of Φ. Clearly, as one cannot assume that an extended (k, r)-expression representing H is known in advance, this algorithm alone does not guarantee that $\mathcal{X}_{k,r}$ is plain exponential. However, we also show that given a graph H of extended clique width at most (k, r), an extended (k, r)-expression representing H can be found in time $O^*((4\max(k, r) + 4)^{|V(H)|})$. Combined with the previous result we thus obtain the following

▶ **Theorem 1.** *For any fixed k, r the class of graphs of extended clique width at most (k, r) is plain exponential.*

Next, we show that the classes of graphs of bounded extended clique width are quite general. Let Hypercubes denote the class of all hypercubes and let Grids denote the class of all rectangular grids. Also, for a class $\mathcal{H}$ of graphs, $\mathcal{K}(\mathcal{H})$ denotes the class of graphs H obtained as follows. Take $H' \in \mathcal{H}$, a clique on vertices $\{v_1, \ldots, v_n\}$, and for every edge v_iv_j of the clique, $i \neq j$, replace this edge with a copy of H', that is, connect v_i, v_j to all vertices of H' and include all the edges of H'. It is known that all three classes have unbounded clique width [20, 3], and $\mathcal{K}(\mathcal{H})$ has unbounded clique width even when $\mathcal{H}$ contains just one graph with one vertex.

▶ **Theorem 2.** Hypercubes *has extended clique width at most (2,1),* Grids *has extended clique width at most (6,1). For any class $\mathcal{H}$ of extended clique width (k, r), the class $\mathcal{K}(\mathcal{H})$ has extended clique width at most $(k + 5, \max(r, 1))$.*

By Theorem 1 this immediately implies that classes Hypercubes and Grids are plain exponential. For subdivisions of cliques we prove a stronger result.

▶ **Proposition 3.** *For any plain exponential class $\mathcal{H}$ of graphs (not necessarily of bounded extended clique width), the class $\mathcal{K}(\mathcal{H})$ is also plain exponential.*

It seems that there are two general reasons for a class of graphs to be plain exponential: to have bounded (extended) clique width or to have bounded degree. Classes Hypercubes and $\mathcal{K}(\mathcal{H})$ witness that bounded extended clique width (and in fact even bounded clique width) does not imply bounded degree. By proving that graphs from $\mathcal{X}_{k,r}$ satisfy certain nontrivial property and showing that a random c-regular graph for $c > 3$ (unsurprisingly) does not satisfy this property with high probability, we show that $\mathcal{D}_c$ does not have bounded extended clique width. The two types of classes can be combined together to obtain new plain exponential classes. Let G, H be graphs. The Cartesian product $G\square H$ of G and H is defined to be the graph with vertex set $V(G) \times V(H)$ and edges $(u_1, v_1)(u_2, v_2)$ such that either $u_1 = u_2$ and $v_1v_2 \in E(H)$, or $v_1 = v_2$ and $u_1u_2 \in E(G)$.

▶ **Theorem 4.** *Let $\mathcal{G}$ be a plain exponential class of graphs and $\mathcal{H}$ of bounded degree. Then $\mathcal{G}\square\mathcal{H} = \{G\square H \mid G \in \mathcal{G}, H \in \mathcal{H}\}$ is plain exponential.*

Note that for another standard graph product, $G \times H$, where edges are given by the rule: $(u_1, v_1)(u_2, v_2)$ is an edge if and only if $u_1u_2 \in E(G)$ and $v_1v_2 \in E(H)$, a similar result is almost trivial, as we observe in Example 9.

There is no doubt that plain exponential classes are much more diverse than what is shown above. For instance, for a class $\mathcal{G}$ of graphs, let $\mathcal{G}^{+d}$ denote the class of graphs G such that it is possible to remove up to d vertices from G to obtain a graph from $\mathcal{G}$. Then as is easily seen, $\mathcal{G}^{+d}$ is plain exponential whenever $\mathcal{G}$ is plain exponential. Also, there are some odd plain exponential class of graphs (odd in the sense we could not fit it into any of the types above). Let Kneser_k denote the well studied class of Kneser graphs, see, e.g. [19]: Kneser_k is the class of graphs, whose vertices are the k-element subsets of a certain set, and two vertices are connected if and only if the corresponding subsets are disjoint. A plain exponential algorithm for Kneser_k (for a fixed k) exists, see [2, 21]. We find an alternative algorithm for this class of graphs.

▶ **Theorem 5.** *For every k the class* Kneser_k *is plain exponential.*

The class Kneser_k however does not fit in any of the more general classes of plain exponential graphs.

Due to space restrictions not all proofs are included in this paper. For the missing proofs the reader is referred to the full version of the paper [7].

2 Preliminaries: Homomorphisms and Clique width

2.1 Homomorphisms, plain exponential time

By $[n]$ we denote the set $\{1, \ldots, n\}$ and by $[[n]]$ the set $\{0, 1, \ldots, n\}$. As always we denote the vertex set of a graph G by $V(G)$, and its edge set by $E(G)$. A *homomorphism* of a graph G to a graph H is a mapping $\varphi\colon V(G) \to V(H)$ such that $\varphi(u)\varphi(v) \in E(H)$ for any $uv \in E(G)$. By $\mathsf{hom}(G, H)$ we denote the number of homomorphisms from G to H. The *Counting Graph Homomorphism* problem, #GraphHom, is defined as follows: given graphs G, H, find the number of homomorphisms from G to H. Its decision version – does there exist a homomorphism from G to H? – is denoted by GraphHom. For more on graph homomorphisms see [14]. If H is allowed only from a class $\mathcal{H}$ of graphs, the resulting counting and decision problems are denoted #GraphHom$(-, \mathcal{H})$ and GraphHom$(-, \mathcal{H})$, respectively.

We will be concerned with the complexity and the best running time of algorithms for #GraphHom$(-, \mathcal{H})$. In particular, we say that a class $\mathcal{H}$ of graphs is *plain exponential* if there is an algorithm that solves the problem #GraphHom$(-, \mathcal{H})$ in *plain exponential time*: there exists a constant c such that on input G, H, $H \in \mathcal{H}$, the algorithm runs in time $O^*(c^{|V(G)|+|V(H)|})$, where O^* means asymptotics up to a factor polynomial in $|V(G)|, |V(H)|$. Note that we will always assume that G and H are connected, since otherwise the existence or the number of homomorphisms from G to H can be deduced from those of their connected components.

▶ **Example 6.** (H-Colouring.) If $\mathcal{H}$ consists of just one graph, H, the problems #GraphHom$(-, \mathcal{H})$, GraphHom$(-, \mathcal{H})$ are known as #H-Colouring and H-Colouring, respectively. The #H-Colouring problem is solvable in polynomial time if H is a complete graph with all loops present, or is a complete bipartite graph [9]. The H-Colouring problem is solvable in polynomial time if H contains a loop or is bipartite [15]. Otherwise these problems are #P- and NP-complete, respectively. Since the brute force algorithm for this problems runs in $O(|V(H)|^{|V(G)|})$ time, #H-Colouring and H-Colouring are always solvable in plain exponential time. Also, by inspecting the solution algorithms from [9, 15] these results can be slightly generalized: #GraphHom$(-,\mathcal{H})$ is solvable in polynomial time whenever every graph from $\mathcal{H}$ is a complete graph with all loops, or a complete bipartite graph. Similarly GraphHom$(-,\mathcal{H})$ is polynomial time solvable if every graph from H contains a loop or is bipartite.

▶ **Example 7.** (Graphs of bounded degree.) As is mentioned in the introduction, if the degrees of graphs from $\mathcal{H}$ are bounded by a number c, the (improved) brute force algorithm solves #GraphHom$(-,\mathcal{H})$, GraphHom$(-,\mathcal{H})$ in time $O^*(c^{|V(G)|})$. Let G,H be input graphs, $H\in\mathcal{H}$. Recall that we assume G is connected; otherwise the procedure below has to be performed for each connected component, and the results multiplied. Order the vertices $v_1,\dots,v_n$ of G in such a way that each vertex except for the first one is adjacent to one of the preceding vertices. Then the brute force algorithm is organized as follows: Assign images to $v_1,\dots,v_n$ in turn. There are $|V(H)|$ possibilities to map v_1, but then if v_i is adjacent to v_j, $j<i$, the image of v_j is fixed, and therefore there are at most c possibilities for the image of v_i. Thus, the algorithm runs in $O^*(c^n)$. This approach also allows H to have a bounded number of vertices of high degree.

▶ **Example 8.** (Graphs of bounded clique width.) Let $\mathcal{C}_k$ denote the class of all graphs of clique width at most k (to be defined in Section 2.2). Then #GraphHom$(-,\mathcal{C}_k)$, GraphHom$(-,\mathcal{C}_k)$ can be solved in time $O^*((2k+1)^{|V(G)|+|V(H)|})$, implying that $\mathcal{C}_k$ is plain exponential [23].

Here we briefly describe the simple algorithm solving #GraphHom$(-,\mathcal{K})$, where $\mathcal{K}$ is the class of cliques. Given a graph G and a number s (or, equivalently, the clique K_s) the solution algorithm maintains an array $N(S,\ell)$ for $S\subseteq V$ and $\ell\le s$, which contains the number of homomorphisms from the subgraph of G induced by S to an ℓ-element clique. To compute each $N(S,\ell)$ we go over all subsets $S'\subseteq S$, consider the vertices from S' to be mapped to the ℓ-th vertex of the ℓ-clique. Then there are $N(S-S',\ell-1)$ ways to map the remaining vertices, and $N(S,\ell)$ is the sum of all numbers like this. It is not hard to see that the running time of this algorithm is $O^*(3^{|V(G)|})$. It can be improved to run in time $O^*(2^{|V(G)|})$ [17], and some further improvements are possible in certain cases [10].

▶ **Example 9.** Often plain exponential classes can be combined to obtain a new plain exponential class. For graphs G,H let $G\times H$ denote their product, the graph with vertex set $V(G)\times V(H)$ and edges $(u_1,v_2)(u_2,v_2)$ whenever $u_1u_2\in E(G)$ and $v_1v_2\in E(H)$. Also, for graph classes $\mathcal{G},\mathcal{H}$, let $\mathcal{G}\times\mathcal{H}$ denote the class $\{G\times H\mid G\in\mathcal{G},H\in\mathcal{H}\}$. If $\mathcal{G},\mathcal{H}$ are plain exponential, then so is $\mathcal{G}\times\mathcal{H}$. Indeed, let π_1,π_2 denote the projection homomorphisms of $G\times H$ onto G and H, respectively; that is, $\pi_1:(u,v)\mapsto u$ and $\pi_2:(u,v)\mapsto v$. For any graph T a mapping $\varphi: V(T)\to V(G)\times V(H)$ is a homomorphism if and only if the mappings $\varphi_1=\pi_1\circ\varphi_1: V(T)\to V(G)$ and $\varphi_2=\pi_2\circ\varphi: V(T)\to V(H)$ are homomorphisms. In this case $\varphi(u,v)=(\varphi_1(u),\varphi_2(v))$. This immediately implies that $\mathsf{hom}(T,G\times H)=\mathsf{hom}(T,G)\cdot\mathsf{hom}(T,\mathcal{H})$, and the result follows.

We will often deal with vertex labeled graphs. It will be convenient to represent labels on vertices of a graph G as a *label function* $\pi\colon V(G)\to[k]$, in which case we say that G is *k-labeled.* The graph $G=(V,E)$ equipped with a label function π will be denoted by $\mathbb{G}=(V,E,\pi)$. The k-labeled graph $\mathbb{G}$ is then called a *k-labeling* of G. Let $\mathbb{G}_1=(V_1,E_1,\pi_1)$ and $\mathbb{G}_2=(V_2,E_2,\pi_2)$ be k-labeled graph. A mapping $\varphi\colon V_1\to V_2$ is a *homomorphism* of k-labeled graph $\mathbb{G}_1$ to k-labeled graph $\mathbb{G}_2$ if it is a homomorphism of graph $G_1=(V_1,E_1)$ to $G_2=(V_2,E_2)$ respecting the labeling, that is, $\pi_2(\varphi(v))=\pi_1(v)$ for every $v\in V_1$.

The following notation will also be useful. Let again $\mathbb{G}_1, \mathbb{G}_2$ be k-labeled graphs, such that V_1, V_2 are disjoint. Then $\mathbb{G}_1 \bigoplus \mathbb{G}_2 = (V_1 \uplus V_2, E_1 \uplus E_2, \pi_1 \uplus \pi_2)$, where $\pi_1 \uplus \pi_2(v) = \pi_1(v)$, if $v \in V_1$, and $\pi_1 \uplus \pi_2(v) = \pi_2(v)$, if $v \in V_2$.

Finally, the subgraph of a graph $G = (V, E)$ induced by a set $S \subseteq V$ is denoted by $G[S]$. For a k-labeled graph $\mathbb{G} = (V, E, \pi)$, by $\mathbb{G}[S]$ we denote the k-labeled subgraph induced by $S \subseteq V$. Note that the label function of $\mathbb{G}[S]$ is $\pi|_S$, i.e., the restriction of π on the set S.

2.2 Clique width and k-expressions

The simplest way to introduce clique width of a graph is through k-expressions.

▶ **Definition 10.** *The following operators are defined on k-labeled graphs.*

- $\cdot_i$: *Construct a graph with one vertex, which is labeled $i \in [k]$.*
- $\rho_{i \to j}(\mathbb{G})$: *Relabel all vertices with label $i \in [k]$ of a k-labeled graph $\mathbb{G}$ to label $j \in [k]$.*
- $\eta_{ij}(\mathbb{G})$, *for $i \neq j$: Add an edge from every vertex labeled i to every vertex labeled j in $\mathbb{G}$, i.e. add edges uv for any vertices u, v where u has label i and v has label j.*
- $\mathbb{G}_1 \bigoplus \mathbb{G}_2$: *The disjoint union of k-labeled graphs $\mathbb{G}_1$ and $\mathbb{G}_2$.*

*A k-*expression *is any (properly formed) formula using the above operators.*

Every k-expression represents a k-labeled graph. We say that a graph $G = (V, E)$ is represented by k-expression Φ, if there exists a k-labeling π of the vertices of G such that Φ represents $\mathbb{G} = (V, E, \pi)$. A graph has clique width *k if k is minimal so that the graph is represented by a k-expression. The class of all graphs of clique width at most k is denoted by $\mathcal{C}_k$.*

Wahlström in [23] used k-expressions of graphs to show that $\mathcal{C}_k$ is plain exponential. However, k-expressions suitable for his plain exponential algorithm must satisfy an extra condition. Let Φ be a k-expression representing a k-labeled graph $\mathbb{G}$. Note that any subexpression of Φ represents a subgraph of $\mathbb{G}$. We say that k-expression Φ is *safe* if for every its subexpression $\Phi_1 \bigoplus \Phi_2$ such that Φ_1, Φ_2 represent graphs $\mathbb{G}_1, \mathbb{G}_2$, respectively, the graph $\mathbb{G}_i$ equals $\mathbb{G}[V(\mathbb{G}_i)]$ for $i = 1, 2$. In other words all edges of $\mathbb{G}$ between vertices of $\mathbb{G}_i$, $i = 1, 2$, are already edges of $\mathbb{G}_i$.

▶ **Lemma 11** ([23]).

(1) *Every graph of clique width k can be represented by a safe k-expression.*

(2) *A safe k-expression for a graph of clique width k can be found in plain exponential time.*

A class $\mathcal{G}$ of graphs has bounded clique width if $\mathcal{G} \subseteq \mathcal{C}_k$ for some k. Classes of bounded clique width include cliques, cographs, and distance-hereditary graphs [12, 4]. We will also be interested in nice graph classes that do not have bounded clique width. These include classes Hypercubes of hypercubes, Grids of rectangular grids, and subdivisions of cliques $\mathcal{K}(\mathcal{H})$ (introduced in Section 1) [20, 3].

3 Extended clique width

3.1 Extended k-expressions

In this section we introduce a more general version of k-expressions, and accordingly a more general version of clique width. New k-expressions require two more operators on k-labeled graphs. The first one does not have analogues in k-expressions. Let $\mathbb{G}$ be a k-labeled graph and r a positive integer parameter. The idea behind the *inflation* operator is the following. For each vertex v of $\mathbb{G}$ we add up to r new copies of v. The set of new copies of v depends only on the label of v, and is given by the vector $\overrightarrow{M}$ defined below. Let $v_{i_1},\dots,v_{i_\ell}$ be the added copies of v, and v itself is considered as v_0. Next, some new edges are introduced: whether or not edge v_iw_j is added depends only on whether $vw \in V(\mathcal{G})$, the labels of v,w, and the numbers i,j. These connections are given by the set $\mathcal{S}$ defined below. Finally, the new copies obtain labels, and the label of v_i only depends on the label of v in $\mathbb{G}$ and i. This step is governed by the vector $\overrightarrow{\sigma}$ in the definition below.

We now proceed to a formal definition. Fix natural k,r. By $\overrightarrow{M}$ we denote a vector $(M_1,\dots,M_k)$ where each M_i is a subset of $[[r]]$ containing 0. For such a vector $\overrightarrow{M}$, let

$$\mathcal{L}(\overrightarrow{M}) = \{(i_1,j_1,i_2,j_2) \mid i_1,i_2 \in [k], j_1 \in M_{i_1}, j_2 \in M_{i_2}\}.$$

▶ **Definition 12.** *Let $\overrightarrow{M} = (M_1,\dots,M_k)$, $\{0\} \subseteq M_i \subseteq [[r]]$ for $i \in [k]$, $\sigma_j : [k] \to [k]$ for $j \le [[r]]$, where σ_0 is the identity mapping, and $\mathcal{S} \subseteq \mathcal{L}(\overrightarrow{M})$. Also, $\mathcal{S}$ is required to be a symmetric set, that is, if $(i_1,j_1,i_2,j_2) \in \mathcal{S}$ then $(i_2,j_2,i_1,j_1) \in \mathcal{S}$. Operator $\beta_{\overrightarrow{M},\overrightarrow{\sigma},\mathcal{S}}$ transforms a k-labeled graph $\mathbb{G} = (V,E,\pi)$ to a k-labeled graph $\mathbb{G}' = (V',E',\pi')$ as follows:*

- *$V' = \bigcup_{i=1}^k C_i$, where $C_i = \{a_j | j \in M_i,\ a \in V$ and $\pi(a) = i\}$. The vertices a_0, $a \in V$, are called original vertices of $\mathbb{G}' = \beta_{\overrightarrow{M},\overrightarrow{\sigma},\mathcal{S}}(\mathbb{G})$ and are identified with their corresponding vertices from V;*
- *$a_jb_{j'} \in E'$ if and only if $ab \in E$, and $(\pi(a),j,\pi(b),j') \in \mathcal{S}$ or $j = j' = 0$;*
- *$\pi'(a_j) = \sigma_j(\pi(a))$*

The second operator combines disjoint union with a sequence of adding edges operators.

▶ **Definition 13.** *Let $\mathcal{T} \subseteq [k] \times [k]$. Operator $\eta_{\mathcal{T}}$ takes two k-labeled graphs as input and produces a k-labeled graph as output. For k-labeled graphs $\mathbb{G}_1 = (V_1,E_1,\pi_1)$, and $\mathbb{G}_2 = (V_2,E_2,\pi_2)$, V_1,V_2 disjoint, the k-labeled graph $\eta_{\mathcal{T}}(\mathbb{G}_1,\mathbb{G}_2) = (V,E,\pi)$, is defined as follows:*

- *$V = V_1 \cup V_2$;*
- *$E = E_1 \cup E_2 \cup \{(a,b) \mid a \in V_1, b \in V_2, \pi_1(a) = i, \pi_2(b) = j,\ (i,j) \in \mathcal{T}\}$;*
- *$\pi(a) = \pi_1(a)$ if $a \in V_1$ and $\pi(a) = \pi_2(a)$ if $a \in V_2$.*

We refer to this operator as the connect operator.

An *extended (k,r)-expression* is a (properly formed) expression that involves operators $\cdot_i$ $(i \in [k])$, $\rho_{i\to j}$ $(i,j \in [k])$, $\beta_{\overrightarrow{M},\overrightarrow{\sigma},\mathcal{S}}$, and $\eta_{\mathcal{T}}$, where $\overrightarrow{M},\overrightarrow{\sigma},\mathcal{S},\mathcal{T}$ are as in Definitions 12, 13. Similar to k-expressions, extended (k,r)-expressions represent k-labeled graphs, as well as usual graphs. For an example of extended (k,r)-expression see the construction of hypercubes in Section 3.2.

Note that if $\mathbb{G}_1$ and $\mathbb{G}_2$ are two isomorphic k-labeled graphs, and $\mathbb{G}_1$ is represented by an extended (k,r)-expression Φ, then Φ is an extended (k,r)-expression representing $\mathbb{G}_2$ as well.

A graph $G = (V,E)$ is said to have *extended clique width* (k,r) if the pair (k,r) is minimal such that there is a k-labeling π of G and an extended (k,r)-expression Φ that represents $\mathbb{G} = (V,E,\pi)$. If such a π exists we also say that Φ represents G. Note that an extended clique width of a graph is not unique, as pairs of numbers can be incomparable. However, for our purposes it will usually be enough to assume that $k = r$: just replace both parameters with $\max(k,r)$. The class of all graphs of extended clique width at most (k,r) is denoted by $\mathcal{X}_{k,r}$. A class $\mathcal{G}$ of graphs has bounded extended clique width if $\mathcal{G} \subseteq \mathcal{X}_{k,k}$ for some k.

The connect operator is clearly a substitute for the operator η_{ij} from Definition 10 of clique width. In particular, graphs of extended clique width $(k,0)$ are very close to graphs of clique width k.

▶ **Proposition 14.** *Any graph G that can be represented by a k-expression, can also be represented by an extended $(k, 0)$-expression. Therefore, $\mathcal{C}_k \subseteq \mathcal{X}_{k,0}$, that is, every graph that has clique width k also has extended clique width at most $(k, 0)$.*

As is easily seen, the connect operator can be expressed through disjoint union and adding edges. However, we will need properties similar to the safety of k-expressions. Unfortunately, the inflation operator does not allow for an equally clean and easy definition of safety, as in the case of k-expressions, and we use the connect operator instead.

Let $\mathbb{G} = \eta_{\mathcal{T}}(\mathbb{G}_1, \mathbb{G}_2)$. It is straightforward from the definition that $\mathbb{G}[V(\mathbb{G}_1)]$ is equal to $\mathbb{G}_1$ and $\mathbb{G}[V(\mathbb{G}_2)]$ is equal to $\mathbb{G}_2$, that is, $\eta_{\mathcal{T}}$ does not add edges inside $\mathbb{G}_1, \mathbb{G}_2$. Also, if $\mathbb{G} = \beta_{\overrightarrow{M},\overrightarrow{\sigma},\mathcal{S}}(\mathbb{G}_1)$, then again $\mathbb{G}[V(\mathbb{G}_1)]$ is equal to $\mathbb{G}_1$. Similar to k-expressions we say that an extended (k, r)-expression Φ is *safe* if for each of its subexpressions $\eta_{\mathcal{T}}(\Phi_1, \Phi_2)$ and $\beta_{\overrightarrow{M},\overrightarrow{\sigma},\mathcal{S}}(\Phi_1)$ such that Φ_1, Φ_2 represent graphs $\mathbb{G}_1, \mathbb{G}_2$, respectively, it holds $\mathbb{G}_i = \mathbb{G}[V(\mathbb{G}_i)]$ for $i = 1, 2$. The following property is straightforward.

▶ **Lemma 15.** *Any extended (k, r)-expression is safe.*

An extended (k, r)-expression representing G (if one exists) can be found in plain exponential time.

▶ **Theorem 16.** *There is an algorithm running in time $O^*((4 \max(k, r) + 4)^{|V(G)|})$ that given a graph G outputs an extended (k, r)-expression for G if one exists, or reports "NO" otherwise.*

Proof (Sketch). One of the ingredients of our algorithm is the problem of deciding whether two k-labeled graphs are isomorphic. k-labeled graphs $\mathbb{G} = (V_1, E_1, \pi_1), \mathbb{H} = (V_2, E_2, \pi_2)$ are isomorphic if there exists an isomorphism φ from the graph $G = (V_1, E_1)$ to $H = (V_2, E_2)$ such that $\pi_1(a) = \pi_2(\varphi(a))$ for $a \in V_1$. We show that this problem can be reduced to the ordinary Graph Isomorphism problem and use the celebrated result by Babai [1] that there is an algorithm that, given graphs G and H, decides whether there exists an isomorphism between G and H in time $O(2^{\log(|V(G)|)^{O(1)}})$.

▶ **Lemma 17.** *There is a polynomial time reduction from the problem of deciding the isomorphism of k-labeled graphs to Graph Isomorphism.*

We now describe the main part of the algorithm. Create an array N of size $(k + 1)^n$ whose entries $N(\mathbb{G}')$ are labeled with a k-labeling $\mathbb{G}'$ of a subgraph G' of G. For every entry $N(\mathbb{G}')$ the k-labeled graph $\mathbb{G}'$ either has an extended (k, r)-expression or it does not. The goal is to set the value of each entry $N(\mathbb{G}')$ to some extended (k, r)-expression for $\mathbb{G}'$ if it has one and to "no" otherwise. Then either for some labeling $\mathbb{G}$ of G the entry $N(\mathbb{G})$ contains a (k, r)-expression for G, or G does not have extended clique width at most (k, r).

Now we consider more detailed possibilities for each $\mathbb{G}'$. There are four cases. Case 1 takes place if $\mathbb{G}'$ has an extended (k, r)-expression that ends with an inflation operator; Case 2 takes place if $\mathbb{G}'$ has an extended (k, r)-expression that ends with a connect operator; Case 3 takes place if $\mathbb{G}'$ has an extended (k, r)-expression that ends with a sequence of relabeling operators; and, finally, Case 4 takes place if $\mathbb{G}'$ does not have an extended (k, r)-expression.

All one-element k-labeled graphs are obviously represented by an extended (k,r)-expression. Let us suppose the values of each entry $N(\mathbb{G}')$, where $\mathbb{G}'$ contains at most $n-1$ vertices is set correctly. Then, we want to set the correct values for entries of the array whose associated k-labeled graph has exactly n vertices. We use the dynamic programming approach that consists of two phases. In Phase 1, for each entry $N(\mathbb{G}')$ such that $\mathbb{G}'$ has n vertices, we check if $\mathbb{G}'$ satisfies the conditions of Case 1. Then for each k-labeled graph like this that does not satisfy the conditions of Case 1 we check if it falls in Case 2. In Phase 2, by relabeling $\mathbb{G}'$ for which $N(\mathbb{G}')$ is assigned a value, we find a new extended (k, r)-expression for $\mathbb{G}'$ that do not satisfy the conditions of Cases 1 and 2, but satisfy the conditions of Case 3. In the end, for each $\mathbb{G}'$ that belongs to none of Cases 1, 2, or 3, we set the value $N(\mathbb{G}')$ to "no" because it does not have an extended (k, r)-expression. In the rest of this proof, for a k-labeled graph $\mathbb{G}'$, we show how to check if it satisfies the conditions of each of Cases 1 and 2.

Let $\mathbb{G}' = (V', E', \pi')$ be a k-labeled graph with $|V'| = n$ and it has an extended (k, r)-expression that ends with an inflation operator. Then there is an induced subgraph $\mathbb{G}'_1$ of $\mathbb{G}'$ such that the result of application of an inflation operator to $\mathbb{G}'_1$ is isomorphic to $\mathbb{G}'$, and $\mathbb{G}'_1$ has an extended (k, r)-expression. Thus, there exist $\sigma_i : [k] \to [k]$, $i \in [[r]]$, $\overrightarrow{M}$, $M_i \subseteq [[r]]$, $i \in [k]$, $\mathcal{S} \subseteq \mathcal{L}(\overrightarrow{M})$, and a set $V'_1 \subset V'$, such that

(A) $\mathbb{G}'_2 = (V'_2, E'_2, \pi'_2) = \beta_{\overrightarrow{M},\overrightarrow{\sigma},\mathcal{S}}(\mathbb{G}'[V'_1])$ is isomorphic to $\mathbb{G}'$, and

(B) $\mathbb{G}'[V'_1]$ has an extended (k, r)-expression.

Conversely, if there exist $V'_1 \subset V'$, $\sigma_i : [k] \to [k]$, $i \in [[r]]$, $\overrightarrow{M}$, $M_i \subseteq [[r]]$, $i \in [k]$, $\mathcal{S} \subseteq \mathcal{L}(\overrightarrow{M})$ satisfying conditions (A),(B), then $\mathbb{G}'_2$ has an extended (k, r)-expression that ends with an inflation operator. As $\mathbb{G}'_2$ and $\mathbb{G}'$ are isomorphic, $\mathbb{G}'$ has an extended k-expression that ends with an inflation operator as well. Thus, the sufficient and necessary conditions for $\mathbb{G}'$ to have an extended (k, r)-expression that ends with an inflation operator, is that there exist $V'_1 \subset V'$, $\sigma_i : [k] \to [k]$, $i \in [[r]]$, $\overrightarrow{M}$, $M_i \subseteq [[r]]$, $i \in [k]$, $\mathcal{S} \subseteq \mathcal{L}(\overrightarrow{M})$ satisfying (A),(B).

The algorithm now searches through all possible selection of V'_1, $\overrightarrow{\sigma}$, $\mathcal{S}$, to check if conditions (A),(B) satisfied for any of them. Let us evaluate the running time of this procedure. Checking condition (A) takes time $O(2^{log((k+2)n)^{O(1)}})$ by Lemma 17, while condition (B) can be verified by looking up the existing entry $N(\mathbb{G}'[V'_1])$ in $O(1)$ time. There are 2^n choices for V'_1 and k^{rk} choices for $\overrightarrow{\sigma}$. Vector $\overrightarrow{M}$ can be chosen in 2^{rk} ways, and so $\mathcal{L}(\overrightarrow{M})$ has at most $2^{2rk}k^2$ elements. Thus, $\mathcal{S}$ can be chosen in at most $2^{2^{2rk}k^2}$ ways. Thus, the total running time of filling up $N(\mathbb{G}')$ in this case is upper bounded by

$$2^{|V(G)|} \times k^{rk} \times 2^{2rk}k^2 \times 2^{2^{2rk}k^2} \times O(2^{log((k+2)n)^{O(1)}}) = O^*(2^{2|V(G)|}).$$

Now let us suppose that $\mathbb{G}' = (V', E', \pi')$ has an extended (k, r)-expression that ends with a connect operator. Then due to the safety of extended (k, r)-expressions there exist two induced subgraphs $\mathbb{G}'_1$ and $\mathbb{G}'_2$ of $\mathbb{G}'$ such that, first, they both are represented by extended (k, r)-expressions, and, second, there is $\mathcal{T} \subseteq [k]^2$, such that $\eta_{\mathcal{T}}(\mathbb{G}'_1, \mathbb{G}'_2)$ is identical to $\mathbb{G}'$. Thus to find an extended (k, r)-expression for $\mathbb{G}'$ it suffices to go through all partitions of V' into sets V'_1 and V'_2 and for each partition check the following two conditions. First, check if $\mathbb{G}'_1 = \mathbb{G}'[V'_1]$ and $\mathbb{G}'_2 = \mathbb{G}'[V'_2]$ have an extended (k, r)-expression by looking up the entries $N(\mathbb{G}'_1), N(\mathbb{G}'_2)$. Second, check if there is $\mathcal{T} \subseteq [k]^2$ such that $\eta_{\mathcal{T}}(\mathbb{G}'_1, \mathbb{G}'_2)$ is identical to $\mathbb{G}$. Since there are at most $2^{|V(G)|}$ ways to partition V' into V'_1 and V'_2, takes time $O(2^{|V(G)|})$ to check if $\mathbb{G}'$ falls into Case 2.

So far we have registered an extended (k, r)-expression for every $\mathbb{G}'$ that satisfies the conditions of Case 1 or Case 2. Now, start Phase 2 and check whether any of the remaining k-labeled graphs $\mathbb{G}'$ satisfies the conditions of Case 3. In order to do that we go through all k-labeled graphs $\mathbb{G}'$ with n vertices and such that $N(\mathbb{G}')$ contains an extended (k, r)-expression Φ, that is initially for all $\mathbb{G}'$ that fall into Cases 1,2. Then we consider every possible relabeling ρ_{ij} in turn. If $\rho_{ij}(\mathbb{G}')$ is a k-labeled graph such that $N(\mathbb{G}')$ does not have an extended (k, r)-expression, then we set $N(\rho_{ij}(\mathbb{G}')) = \rho_{ij}(\Phi)$. We repeat this process for each k-labeled graph $\mathbb{G}'$, until no new entries can be filled. The time required for Phase 2 in total, for all $\mathbb{G}'$, not only those with n vertices is bounded by number of all k-labelings of all subgraphs of G times the number of possible operators ρ_{ij}. As is easily seen, the time required for Phase 2 in total is

$$(k+1)^{|V(G)|} \times k^2 = O^*((k+1)^{|V(G)|})$$

Time complexity: The array we construct has $(k+1)^{|V(G)|}$ entries. The time required to complete Phase 1 for all the entries is bounded by $O^*(4^{|V(G)|} \times (k+1)^{|V(G)|})$. The time to complete Phase 2 for all entries is bounded by $O^*((k+1)^{|V(G)|})$. Thus the total running time is $O^*((4k+4)^{|V(G)|})$. ◀

Next we explore what kind of graphs and k-labeled graphs can be represented by extended (k,r)-expressions.

3.2 Graph classes of bounded extended but not regular clique width

In this section we show that not all graphs of bounded extended clique width also have bounded clique width. Specifically, we consider the classes Hypercubes of hypercubes, Grids of rectangular grids, and $\mathcal{K}(\mathcal{H})$ of sudivisions of cliques by graphs from a class $\mathcal{H}$. All these classes have unbounded clique width, as mentioned in Section 2.2.

▶ **Theorem 18.**

(1) Hypercubes *has extended clique width at most (2,1).*

(2) Grids *has extended clique width at most (6,1).*

(3) *If $\mathcal{H}$ is a class of graphs of extended clique width (k,r), then $\mathcal{K}(\mathcal{H})$ has extended clique width at most $(k+5, \max(r,1))$.*

Proof. We present extended (2,1)-expressions for hypercubes and extended (6,1)-expressions for grids. Extended expressions for subdivide cliques are more involved, and the reader is referred to the full version of the paper [7].

(1) Let HC_n denote an n-dimensional hypercube. An extended (2,1)-expression Φ_n representing HC_n is constructed by induction on the dimensionality of the hypercube. The base cases of induction are HC_0 and HC_1. An extended (2,1)-expression for HC_0 is $\cdot_1$, and an extended (2,1)-expression for HC_1 is $\eta_{\{(1,2)\}}(\cdot_1, \cdot_2)$.

Suppose that for $m \leq n$ the graph HC_m has an extended (2,1)-expression. Let Φ_n be an extended (2,1)-expression for HC_n. Let $\overrightarrow{M} = (\{0,1\},\{0,1\})$, let σ_0 be the identity mapping on [2], let $\sigma_1 : [2] \to [2]$ be given by $\sigma_1(1) = 2, \sigma_1(2) = 1$, and let $\mathcal{S} = \{(1,1,1,1),(2,1,2,1),(1,1,2,1),\ (2,1,1,1),(1,0,2,1),(2,1,1,0),(2,0,1,1),(1,1,2,0)\}$. Then it is not hard to see that $\beta_{\overrightarrow{M},\overrightarrow{\sigma},\mathcal{S}}\Phi_n$ is an extended (2,1)-expression for HC_{n+1}.

(2) Let us denote the vertex set of an $n \times m$-grid by $G_{n,m} = [n] \times [m]$. We proceed by induction on n, m. First, observe that a 2×2-grid labeled in an arbitrary way with 4 labels can be represented by a 4-expression in a straightforward manner. We choose the labeling π_{22} of $G_{2,2}$ given by $\pi_{22}(1,1) = 1, \pi_{22}(2,1) = 2, \pi_{22}(2,1) = 3, \pi_{22}(2,2) = 4$.

Next, we construct a 6-expression (not an extended one) for a $2 \times m$-grid labeled in a specific way. The labeling π'_{2m} of $G_{2,m}$ we achieve is given by $\pi'_{2m}(1,1) = \cdots = \pi'_{2m}(1,m-1) = 1$, $\pi'_{2m}(2,1) = \cdots = \pi'_{2m}(2,m-1) = 2$, $\pi'_{2m}(1,m) = 3$, $\pi'_{2m}(2,m) = 4$. Suppose we have constructed a 6-expression representing a $2 \times (m-1)$-grid labeled this way. Then add vertices $(1,m)$ and $(2,m)$ labeled 5 and 6, respectively, and apply operators $\eta_{53}, \eta_{56}, \eta_{64}$, and $\rho_{3\to1}, \rho_{4\to2}, \rho_{5\to3}, \eta_{6\to4}$. It is straightforward that the resulting labeled graph is a $2 \times m$-grid labeled in the required way.

Now starting with the labeled $2 \times m$-grid constructed in the previous step we show by induction that a $n \times m$-grid with labeling π_{nm} can be represented by an extended (4,1)-expression, where π_{nm} is given by $\pi_{nm}(i,j) = 3$ for $i \leq n-2$ and $j \in [m]$, $\pi_{nm}(n-1,j) = 1, \pi_{nm}(n,j) = 2$ for $j \in [m]$. The base case for induction, the grid $G_{2,m}$ labeled with π_{2m} can be obtained from the labeled grid constructed in the previous paragraph by applying operators $\rho_{3\to 1}$ and $\rho_{4\to 2}$. Suppose that an extended (4,1)-expression representing $G_{n-1,m}$ labeled with π_{n-1m} exists. For the induction step we consider inflation operator with the following parameters: $k = 4$, $r = 1$, $\overrightarrow{M} = (\{0,1\},\{0\},\{0\},\{0\})$, $\mathcal{S} = \{(1,1,2,0),(2,0,1,1),(1,1,1,1)\}$, σ_0 is the identity mapping on [4] and $\sigma_1(i) = i$, except $\sigma_1(1) = 4$. The operator $\beta_{\overrightarrow{M},\overrightarrow{\sigma},\mathcal{S}}$ applied to $G_{n-1,m}$ labeled with π_{n-1m} works as follows: it creates an extra copy of each vertex with label 1, that is, of $n-2$-nd row, and connects each new vertex a_1 to every vertex with label 2, a is connected to. In other words, if $a = (n-2,i)$, then a_1 plays the role of (n,i) and is properly connected to the only vertex with label 2 vertex $(n-2,i)$ is connected to, that is, $(n-1,i)$. Also, $\beta_{\overrightarrow{M},\sigma,\mathcal{S}}$ connects vertices $(n,i),(n,i+1)$. Finally, the vertices of the form (n,i) are assigned label 4. In order to obtain a grid labeled with π_{nm} it suffices to apply operators $\rho_{1\to 3}$, $\rho_{2\to 1}$ and $\rho_{4\to 2}$. ◀

4 Counting homomorphism to labeled graphs given an extended *k*-expression

In this section we prove our main result.

▶ **Theorem 19.** *Let G and H be two graphs, and let k-labeled graph $\mathbb{H}$ be a k-labeling of graph H. Given an extended (k,r)-expression Φ for $\mathbb{H}$, $\mathsf{hom}(G,H)$ can be found in time* $O^*((2(\max(k,r)+1))^{2|V(G)|})$

The following notation and terminology will be used throughout this section. Let $\mathsf{HOM}(G,H)$ denote the set of all homomorphisms from G to H. Let $X \subseteq V(G)$, and let $\chi : X \to [k]$ be a label function. A mapping φ from X to k-labeled graph $\mathbb{H} = (V,E,\pi)$ is said to be consistent with χ if for every $x \in X$ it holds $\pi(\varphi(x)) = \chi(x)$. Let $\mathsf{hom}_\chi(G[X],\mathbb{H})$, $\mathsf{HOM}_\chi(G[X],\mathbb{H})$, $\mathsf{map}_\chi(G[X],\mathbb{H})$, and $\mathsf{MAP}_\chi(G[X],\mathbb{H})$, denote the number of homomorphisms from $G[X]$ to $\mathbb{H}$ consistent with χ, the set of all homomorphisms from $G[X]$ to $\mathbb{H}$ consistent with χ, the number of all mappings from $G[X]$ to $\mathbb{H}$ consistent with χ, and the set of all mappings from $G[X]$ to $\mathbb{H}$ consistent with χ, respectively.

Observing that an extended (k,r)-expression can be naturally viewed as an extended $(\max(k,r),\max(k,r))$-expression, in what follows we assume $k = r$. Let Φ be an extended (k,k)-expression for a k-labeling $\mathbb{H}$ of the graph H. We proceed by induction on the structure of Φ. More precisely, our algorithm will compute entries of an array $\mathsf{hom}(\mathbb{G}[X],\mathbb{H}')$, where $X \subseteq V(G)$ and $\mathbb{G}[X]$ is a k-labeling of $G[X]$, and $\mathbb{H}'$ is the k-labeled graph represented by a subexpression of Φ. Since the labeling of $\mathbb{H}'$ is important in this inductive process, we also cannot avoid labeling the graph $\mathbb{G}$. Operator $\cdot_i$ creating a graph $\mathbb{H}'$ with a single vertex labeled i gives the base case of induction. In this case $\mathsf{hom}(\mathbb{G}[X],\mathbb{H}') = 1$ if all vertices of X are labeled i and $\mathbb{G}[X]$ has no edges; otherwise $\mathsf{hom}(\mathbb{G}[X],\mathbb{H}') = 0$. Finally, after computing the numbers $\mathsf{hom}(\mathbb{G},\mathbb{H})$ for all the k-labelings $\mathbb{G}$ of G, we complete using the following observation.

▶ **Observation 20.** *Let G and H be graphs, and let k-labeled graph $\mathbb{H}$ be a k-labeling of H. Then*

$$\mathsf{hom}(G,H) = \sum_{\chi:V(G)\to[k]} \mathsf{hom}_\chi(G,\mathbb{H})$$

It therefore suffices to show how to compute $\mathsf{hom}(\mathbb{G}[X],\mathbb{H}')$, where $\mathbb{G}[X]$ is an arbitrary k-labeling of $G[X]$, $X \subseteq V(G)$, and $\mathbb{H}'$ is represented by a subexpression Φ' of Φ, provided $\mathsf{hom}(\mathbb{G}[Y],\mathbb{H}'')$ is known for all $Y \subset X$, all labelings $\mathbb{G}[Y]$ of $G[Y]$, and $\mathbb{H}''$ represented by a subexpression Φ'' of Φ' with $\Phi'' \neq \Phi'$. We consider 3 cases depending on the last operator of Φ'. In the cases of the relabeling and connect operators the argument is similar to that for k-expressions. Here we only consider the inflation operator.

4.1 Inflation operator

In this part, we show how to make a recursive step in the case when the last operator of Φ' is an inflation operator. Before explaining this step, we need several definitions.

A *retraction* is a homomorphism ψ from a graph G_2 to its subgraph G_1 such that $\psi(v) = v$ for each vertex v of G_1. In this case the subgraph G_1 is called a *retract* of G_2. A retraction from a k-labeled graph $\mathbb{G}_2 = (V_2, E_2, \pi_2)$ to a k-labeled graph $\mathbb{G}_1 = (V_1, E_1, \pi_1)$ is defined to be a retraction from $G_2 = (V_2, E_2)$ to $G_1 = (V_1, E_1)$ preserving the label function π_2, that is, $\pi_2(v) = \pi_1(\psi(v))$ for all $v \in V_2$.

It will be convenient for us to subdivide operator $\beta_{\overrightarrow{M},\overrightarrow{\sigma},\mathcal{S}}$ into two steps: the first one is expansion of the original graph using $\overrightarrow{M}$ and $\mathcal{S}$, and the second is relabeling of some vertices of the resulting graph using $\overrightarrow{\sigma}$. More specifically, let $\mathbb{H} = (V, E, \pi)$ be a k-labeled graph, $\overrightarrow{M}$, $M_i \subseteq [[k]]$ for $i \in [k]$ (recall that we assume $k = r$), $\mathcal{S} \subseteq \mathcal{L}(\overrightarrow{M})$, and $\sigma_i : [k] \to [k]$, $i \in [[k]]$. Then $\mathbb{H}' = (V', E', \pi') = \alpha_{\overrightarrow{M},\mathcal{S}}(\mathbb{H})$ is given by

- $V' = \bigcup_{i=1}^{k} C_i$, where $C_i = \{a_j | j \in M_i,\ a \in V \text{ and } \pi_1(a) = i\}$. The vertices a_0, $a \in V$, are called original vertices of $\mathbb{H}' = \alpha_{\overrightarrow{M},\mathcal{S}}(\mathbb{H})$ and are identified with their corresponding vertices from V;
- $(a_j, b_{j'}) \in E'$ if and only if $(a, b) \in E$, and $(\pi(a), j, \pi(b), j') \in \mathcal{S}$ or $j = j' = 0$;
- $\pi'(a_j) = \pi(a)$.

Then, $\mathbb{H}'' = (V'', E'', \pi'') = \beta_{\overrightarrow{M},\overrightarrow{\sigma},\mathcal{S}}(\mathbb{H})$, that is, $V'' = V'$, $E'' = E'$, and $\pi''(a_j) = \sigma_j(\pi(a))$ for $a \in V''$ and $j \in M_{\pi(a)}$.

As is easily seen, $\mathbb{H}$ is an induced subgraph of $\mathbb{H}'$, and a retract. Indeed, the mapping μ that maps every $a_j \in V(\mathbb{H}')$ to a (recall that a_j is a "copy" of some $a \in V(\mathbb{H})$) is a retraction.

The objective is to find a method to express the number of homomorphisms from induced subgraphs of G to $\mathbb{H}''$ given those from induced subgraphs of G to $\mathbb{H}$.

▶ **Lemma 21.** *Let $Y \subseteq V(G)$ and let γ be a function $Y \to [k]$. There is an algorithm that given $\mathsf{hom}_\zeta(G[X],\mathbb{H})$ for all functions ζ from a subset $X \subset Y$ to $[k]$ as input, finds $\mathsf{hom}_\gamma(G[Y],\mathbb{H}'')$ in time $O((2(k+1))^{|V(G)|})$.*

We break this down into two steps. The main result of Step I, which is summarized in Lemma 22, finds an equality for the number of homomorphisms from G to $\mathbb{H}'$. Then, the result for Step II analogous to Lemma 22 finds the number of homomorphisms from induced subgraphs of G to $\mathbb{H}''$ given those for G and $\mathbb{H}'$. As Step II is substantially simpler than Step I, we omit it here.

Step I

Let $\mathbb{H}' = (V', E', \pi') = \alpha_{\overrightarrow{M},\mathcal{S}}(\mathbb{H})$ and $\mathbb{H} = (V, E, \pi)$. Let Y be a subset of $V(G)$ and γ a function $Y \to [k]$. Also, set

$$\mathbb{W}(\gamma) = \{\omega |\ \omega : Y \to [[k]] \text{ and } \forall a \in Y,\ \omega(a) \in M_{\gamma(a)}\}.$$

For $\omega \in \mathbb{W}(\gamma)$, let

$$\mathsf{HOM}_\gamma(G[Y], \mathbb{H}', \omega) = \{\varphi \mid \ \varphi \in \mathsf{HOM}_\gamma(G[Y], \mathbb{H}') \text{ and } \forall a \in Y, \exists b \in V(\mathbb{H}) \ \text{ s. t. } \varphi(a) = b_{\omega(a)}\}.$$

For the rest of Step I, let X' and X'' be two disjoint subsets of $V(G)$ and let $\chi' : X' \to [k]$ and $\chi'' : X'' \to [k]$ be arbitrary functions. Also let $X = X' \uplus X''$ and let $\chi = \chi' \uplus \chi''$.

Let $\mathsf{HOM}_{\chi',\chi''}(G[X], \mathbb{H}')$ denote the set of all elements of $\mathsf{HOM}_\chi(G[X], \mathbb{H}')$ that map a vertex a from X to an original vertex of $\mathbb{H}'$ (recall that any vertex of $\mathbb{H}$ is called an original vertex of $\mathbb{H}'$) if and only if $a \in X'$.

For any $\varphi \in \mathsf{HOM}_{\chi',\chi''}(G[X], \mathbb{H}')$, there is a unique $\omega \in \mathbb{W}(\chi)$ such that φ is also an element of $\mathsf{HOM}_\chi(G[X], \mathbb{H}', \omega)$. Let us call ω the *consistent function* of φ. We then partition $\mathsf{HOM}_{\chi',\chi''}(G[X], \mathbb{H}')$ into smaller subsets and count the elements in each smaller subset. The partition splits $\mathsf{HOM}_{\chi',\chi''}(G[X], \mathbb{H}')$ into sets of homomorphisms that all share the same consistent function $\omega \in \mathbb{W}(\chi)$. As is easily seen, $\mathsf{HOM}_\chi(G[X], \mathbb{H}', \omega) \cap \mathsf{HOM}_{\chi',\chi''}(G[X], \mathbb{H}')$ is such a subset.

Let $\mathcal{B}(\chi', \chi'')$ be the set of all $\omega \in \mathbb{W}(\chi)$ such that ω satisfies the following properties:

(b.1) $\omega \in \mathbb{W}(\chi)$ and $\omega(x) = 0$ if and only if $x \in X'$.

(b.2) For every $a, b \in X$ such that at least one of them is not an element of X', and $ab \in E(G)$ it holds that $(\chi(a), \omega(a), \chi(b), \omega(b)) \in \mathcal{S}$.

Now, as the set of homomorphisms is subdivided into sufficiently small fragments, it is possible to show that the number of elements in $HOM_\chi(G[X], \mathbb{H}')$ such that ω is their consistent function is the same for any $\omega \in \mathcal{B}(\chi', \chi'')$ and it is zero otherwise.

▶ **Lemma 22.** *Let G, $\mathbb{H}$, $\mathbb{H}'$, X', X'', $X = X' \uplus X''$, χ', χ'', and $\chi = \chi' \uplus \chi''$ be defined as above, then*

$$|\mathsf{HOM}_{\chi',\chi''}(G[X], \mathbb{H}')| = |\mathcal{B}(\chi', \chi'')| \times \mathsf{hom}_\chi(G[X], \mathbb{H}).$$

To evaluate the running time of this procedure, note that the algorithm has to enumerate all possible partitions X', X'' of X, and all mappings that can be in $\mathbb{W}(\chi)$. Overall, it amounts to the number of mappings from X to a $k+1$ element set. The number of choices of X is $2^{|V(G)|}$. Thus the running time is bounded by $O((2(k+1))^{|V(G)|})$. Lemma 21 now follows from Lemma 22 and a similar result for Step II.

4.2 Putting pieces together

We are now in a position to prove Theorem 19.

Proof of Theorem 19. By Observation 20, $\mathsf{hom}(G, H)$ equals the sum of $\mathsf{hom}_\chi(G, \mathbb{H})$ over all k-label functions $\chi : V(G) \to [k]$. For each χ we need to compute $\mathsf{hom}_\chi(G, \mathbb{H})$. This computation is done through dynamic programming and requires finding all the numbers of the form $\mathsf{hom}_\chi(G[X], \mathbb{H}')$, where $X \subseteq V(G)$ and $\mathbb{H}'$ is a graph represented by a subexpression of Φ. By Lemma 21 and similar results for relabeling and connect operators computing each such value from the previous values takes $O((2k+1)^{|V(G)|})$ time. There are $k^{|V(G)|}$ label functions χ, and $2^{|V(G)|}$ subsets of $V(G)$. As the number of subexpressions of Φ introduces only a polynomial factor, the running time of the algorithm is $O^*((2(k+1)^{2|V(G)|})$. ◀

5 Beyond bounded extended clique width

In this section we study plain exponential classes that do not have bounded extended clique width. We start with showing that the class of all graphs with degrees less than a constant does not have bounded extended clique width, and how it can be combined with any plain exponential class to produce a new plain exponential class. Then we present two more plain exponential classes of graphs that so far not representable as derivatives of graph with bounded degree and/or bounded extended clique width.

5.1 Bounded degrees

To prove that some graph class does not have bounded extended clique width we first identify two nontrivial properties of graphs whose extended clique width is at most (k, k). Let G be a graph and $\mathcal{N}(v)$ denote the neighborhood of $v \in V(G)$. Also, let H be an induced subgraph of G, and $\mathcal{N}_H(v) = \mathcal{N}(v) \cap H$ for $v \in V(G)$.

▶ **Lemma 23.** *Let G be a connected graph and $|V(G)| = n$. If G has extended clique width at most (k, k) then the following two conditions hold:*

(1) *For any $\frac{k^2}{n} < \alpha \leq \frac{1}{2}$, there exists a subset $W \subseteq V(G)$ with $\frac{\alpha n}{k+1} \leq |W| \leq \alpha n$ such that there are at most 2^k subsets $U_1, \ldots, U_\ell$ of W with the following property: for every $v \in V(G) - W$ either $\mathcal{N}_H(v) = U_i$ for some $i \in [\ell]$, or $\mathcal{N}_H(v) = \mathcal{N}_H(w) \cap U_i$ for some $w \in V(H)$ and $i \in [\ell]$.*

(2) *If in addition the maximal degree of G is d, there is a constant $\delta(d, k)$ that only depends on d and k, such that for any β, $\delta(d, k) < \beta < \frac{1}{2}$, there are subsets $U \subseteq W \subseteq V$ such that $|W| \geq d|U|$, and $\frac{\beta n}{k+1} < |W| \leq \beta n$. Also, there is a partition Π of W into $|U|$ classes such that every vertex from $W - U$ only has neighbors in at most d blocks of Π.*

Then to prove that the class of all graphs whose degrees are bounded by a constant, does not have bounded extended clique width, we prove that a random d-regular graph does not satisfy the property from Lemma 23(2) with high probability, concluding that $\mathcal{D}_d$ does not have bounded extended clique width.

▶ **Lemma 24.** *Let $d > 3$. The probability that a random d-regular graph with n vertices satisfies the condition of Lemma 23(2) is $o(1)$.*

Classes of bounded degree can be combined with any plain exponential class to form another plain exponential class, as the following theorem shows. Let G_1 and G_2 be graphs. The *Cartesian product* of G_1 and G_2, denoted by $G_1 \square G_2$, is the graph whose vertex set is $V(G_1) \times V(G_2)$, and vertices (u_1, v_1) and (u_2, v_2) of $G_1 \square G_2$ are connected with an edge if and only if $u_1 = u_2$ and $v_1 v_2 \in E(G_2)$, or $v_1 = v_2$ and $u_1 u_2 \in E(G_1)$. For classes $\mathcal{G}, \mathcal{H}$ of graphs $\mathcal{G} \square \mathcal{H}$ denotes the class $\{G \square H \mid G \in \mathcal{G}, H \in \mathcal{H}\}$.

▶ **Theorem 25.** *If $\mathcal{D}$ is in plain exponential class of graphs and $\mathcal{B}$ has bounded degree, then $\mathcal{B} \square \mathcal{D}$ is also plain exponential.*

Proof. Let d be a bound on the degree of graphs from $\mathcal{B}$. Let $H = B \square D$, $B \in \mathcal{B}$, $D \in \mathcal{D}$, and let G be a graph. We are concerned with the number $\mathsf{hom}(G, H)$. Without loss of generality assume $V(B) = [r]$. Let P be a r-partition of $V(G)$. Then a mapping $h : V(G) \to V(H)$ is said to be consistent with P if for every $v \in V(G)$ such that $v \in P_i$ it holds $h(v) \in \{(i, e) | e \in V(D)\}$. In other words, h maps vertices of each set $P_i \in V(G)$ to vertices of the same copy of D in the Cartesian product.

Our algorithm will find a set $\mathcal{P}$ of r-partitions of $V(G)$ such that if a homomorphism $h : G \to H$ is consistent with an r-partition P, then $P \in \mathcal{P}$. Every r-partition can be viewed as a mapping from $V(G)$ to $V(B)$. Since B has degree at most d, partitions from $\mathcal{P}$ can be enumerated using a process similar to that in Example 7. Order the vertices $v_1, \dots, v_n$ of G in such a way that each vertex except for the first one is adjacent to one of the preceding vertices. Then a brute force algorithm is organized as follows: Assign images from $V(B)$ to $v_1, \dots, v_n$ in turn. Clearly, there are $V(B)$ possibilities to map v_1. Suppose that images from $V(B)$ are assigned to $v_1, \dots, v_{j-1}$ by a mapping $\pi : \{v_1, \dots, v_{j-1}\} \to V(B)$. We claim that there are just $d + 1$ possibilities to extend π on v_j if we want to keep the possibility that a homomorphism consistent with the obtained r-partition exists. By the choice of the order $v_1, \dots, v_n$, there is v_i, $i < j$, adjacent with v_j. It is possible to assing $\pi(v_j) = \pi(v_i)$. In this case a consistent homomorphism may map the edge v_iv_j to an edge of the form $(\pi(v_i), e_1)(\pi(v_i), e_2)$ for some $e_1e_2 \in E(D)$. Otherwise v_iv_j should be mapped to an edge of the form $(\pi(v_i), e)(\pi(v_j), e)$. In this case there are at most d possibilities for $\pi(v_j)$. Thus, the algorithm enumerates all the required r-partitions in time $O^*((d + 1)^n)$.

Now, let P be one of the r-partitions of $V(G)$ generated in the previous step. Let G'_P be a graph that is obtained by contracting every edge of $V(G)$ whose ends are in different blocks of P. We claim that $\mathsf{hom}(G'_P, D)$ is equal to the number of homomorphisms of G to $B\Box D$ that are consistent with P.

Let $x \in V(G)$ and let $y \in V(G'_P)$. We use the notation $x \in y$, if y is the result of contraction of x with 0 or more other vertices of G. Also, the set of all homomorphism from G to $H = B\Box D$ that are consistent with P is denoted by $HOM(P, G, H)$. We define a mapping φ from elements of $\mathsf{HOM}(G'_P, D)$ to elements of $\mathsf{HOM}(P, G, H)$ as follows: for every $h' \in \mathsf{HOM}(G'_P, D)$ set $\varphi(h') = h$, where h is given by $h(x) = (i, e)$, for $x \in P_i$ and $e = h'(y)$, $x \in y$.

We show that φ is bijective. First, we show that it is injective. If $h'_1, h'_2 \in \mathsf{HOM}(G'_P, D)$ are two different mappings, there is an element $y \in V(G'_P)$ such that $h'_1(y) \neq h'_2(y)$. Therefore, for every $x \in V(G)$ with $x \in y$, $\varphi(h'_1)(x) \neq \varphi(h'_2)(x)$.

Next we prove that φ is also surjective. We define a function $\varphi^{-1} : \mathsf{HOM}(P, G, H) \to \mathsf{MAP}(G'_P, E)$ such that $(\varphi^{-1} \circ \varphi)$ is the identity mapping. Then to complete the proof of surjectivity it only remains to show that the range of φ^{-1} is $\mathsf{HOM}(G'_P, D)$.

Note that for any homomorphism from G to H consistent with P, any $y \in V(G'_P)$, and any $w_1, w_2 \in y$, if $h(w_1) = (i, e)$, then $h(w_2) = (j, e)$ for some $j \in [r]$. We define φ^{-1} as follows. For $h \in \mathsf{HOM}(P, G, H)$ set $\varphi^{-1}(h) = h'$ such that h' is given by $h'(y) = e$, where y is such that for every $x \in y$, $h(x) = (j, e)$ for some $j \in [r]$.

It is straightforward that $(\varphi^{-1} \circ \varphi)$ is the identity function. Now observe that for any $y_1, y_2 \in V(G'_P)$ with $y_1y_2 \in E(G'_P)$ there are $x_1 \in y_1$ and $x_2 \in y_2$ such that $x_1x_2 \in E(G)$. Since y_1 and y_2 are not contracted in G'_P, x_1, x_2 are in the same block P_j of P for some $j \in [r]$. Therefore, $h(x_1) = (j, e_1)$, $h(x_2) = (j, e_2)$, and $e_1e_2 \in E(D)$. Hence $h'(y_1)h'(y_2) = e_1e_2$ is an edge of E. Thus, h' maps an edge of G to an edge of H, and it is a homomorphism. The surjectivity of φ follows.

Finally, since all the r-partitions of G for which there may exist a consistent homomorphism can be enumerated in plain exponential time, and $\mathsf{hom}(G'_P, D)$ can also be found in plain-exponential time for each such partition P, the overall algorithm runs in plain exponential time. ◀

5.2 Subdivided Cliques

Recall that the *subdivision* of an edge uv by a graph H is a graph with vertex set $V(H) \cup \{u, v\}$ and edge set $E(H) \cup \bigcup_{t \in V(H)} \{ut, vt\}$. The subdivision of a graph G by a graph H is the graph obtained by replacing every edge uv of G with its subdivision by a copy of H (a disjoint copy for each edge). Let $\mathcal{K}(\mathcal{H})$ denote the class of subdivisions of cliques by graphs from a class $\mathcal{H}$.

The following theorem is the main result of this section.

▶ **Theorem 26.** *Let $\mathcal{H}$ be a plain exponential class of graphs. Then $\mathcal{K}(\mathcal{H})$ is also plain exponential.*

More precisely, if $\#\textsc{GraphHom}(-, \mathcal{H})$ can be solved in time $O^*(c^{|V(G)|+|V(H)|})$, c constant, for any given graphs G and $H \in \mathcal{H}$, then $\#\textsc{GraphHom}(-, \mathcal{K}(\mathcal{H}))$ can be solved in time $O^*(c_1^{2(|V(G)|+|V(H)|})$, where $c_1 = \max(c, 2)$.

Theorem 18(3) claims that if $\mathcal{H}$ is of bounded extended clique width, then so is $\mathcal{K}(\mathcal{H})$, and then that $\mathcal{K}(\mathcal{H})$ is plain exponential follows from Theorem 19. However, in Theorem 26 $\mathcal{H}$ does not have to be of bounded extended clique width.

5.3 Kneser Graphs

Kneser graphs give another example of a plain exponential class of graphs.

The *Kneser graph* $\mathrm{KG}_{n,k}$ is the graph whose vertex set is the set of k-element subsets of a set with n elements, and two vertices are adjacent if and only if the two corresponding sets are disjoint. By Kneser_k we denote the class of all Kneser graphs for a fixed k. The class Kneser_k is plain exponential, as it follows from the results of [2, 21]. Here we give an alternative algorithm for $\textsc{GraphHom}(-, \mathsf{Kneser}_k)$.

Let G be a graph, and $G^{(k)}$ denote the graph obtained by replacing each of its vertices with a clique of size k and replacing each of its edges with a complete bipartite graph on $k + k$ vertices. For $a \in V(G)$ let $\psi(a)$ denote the set of vertices of the clique replacing v in $G^{(k)}$.

First, we introduce a many to one correspondence between elements of $\mathsf{HOM}(G^{(k)}, K_n)$ and $\mathsf{HOM}(G, \mathrm{KG}_{n,k})$. Let $\tau : \mathsf{HOM}(G^{(k)}, K_n) \to \mathsf{HOM}(G, \mathrm{KG}_{n,k})$ be defined by setting $\tau(\varphi) : V(G) \to \mathrm{KG}_{n,k}$ to be the mapping $v \mapsto \{\varphi(u) | u \in \psi(v)\}$. Notice that the cardinality of $\{\varphi(u) | u \in \psi(v)\}$ equals k because $G^{(k)}[\psi(v)]$ is a k-clique and φ is a homomorphism from $G^{(k)}$ to K_n. Therefore $\tau(\varphi)(v)$ is always a vertex of $\mathrm{KG}_{n,k}$.

It can be shown that for $\varphi \in \mathsf{HOM}(G^{(k)}, K_n)$, $\tau(\varphi)$ is a homomorphism, and moreover for any element σ of $\mathsf{HOM}(G, \mathrm{KG}_{n,k})$, $\tau(\varphi) = \sigma$ for exactly $(k!)^{|V(G)|}$ homomorphisms $\varphi \in \mathsf{HOM}(G^{(k)}, K_n)$. Therefore

$$|\mathsf{HOM}(G, \mathrm{KG}_{n,k})| = \frac{\mathsf{HOM}(G^{(k)}, K_n)}{(k!)^{|V(G)|}}.$$

Since there is an algorithm that computes $\mathsf{hom}(G^{(k)}, K_n)$ in time $O^*(2^{k|V(G)|})$, there is an algorithm that computes $\mathsf{hom}(G, \mathrm{KG}_{n,k})$ in the same time.

▶ Remark 27. The running time of the algorithm above is not plain exponential if k is not a constant. Bonamy et al. [2] proved that the class $\mathsf{Kneser} = \bigcup_{k \in \mathbb{N}} \mathsf{Kneser}_k$ is not plain exponential unless the ETH fails.

Interestingly, the class of Kneser graphs does not have bounded extended clique width.

▶ **Theorem 28.** *The class* Kneser$_2$ *does not have bounded extended clique width.*

To prove this result we use the property of graphs with bounded extended clique width from Lemma 23 (1).

References

1 László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697, 2016.

2 Marthe Bonamy, Lukasz Kowalik, Michal Pilipczuk, Arkadiusz Socala, and Marcin Wrochna. Tight lower bounds for the complexity of multicoloring. *TOCT*, 11(3):13:1–13:19, 2019.

3 Flavia Bonomo, Luciano N. Grippo, Martin Milanic, and Martín Darío Safe. Graph classes with and without powers of bounded clique-width. *Discrete Applied Mathematics*, 199:3–15, 2016.

4 Andreas Brandstädt, Feodor F. Dragan, Hoàng-Oanh Le, and Raffaele Mosca. New graph classes of bounded clique-width. *Theory Comput. Syst.*, 38(5):623–645, 2005.

5 Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006.

6 Marek Cygan, Fedor V. Fomin, Alexander Golovnev, Alexander S. Kulikov, Ivan Mihajlin, Jakub Pachocki, and Arkadiusz Socala. Tight bounds for graph homomorphism and subgraph isomorphism. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1643–1649, 2016.

7 Amineh Dadsetan and Andrei A. Bulatov. Counting homomorphisms in plain exponential time. *CoRR*, abs/1810.03087, 2018. arXiv:1810.03087.

8 Víctor Dalmau and Peter Jonsson. The complexity of counting homomorphisms seen from the other side. *Theor. Comput. Sci.*, 329(1-3):315–323, 2004.

9 Martin E. Dyer and Catherine S. Greenhill. The complexity of counting graph homomorphisms. *Random Struct. Algorithms*, 17(3-4):260–289, 2000.

10 Fedor V. Fomin, Serge Gaspers, and Saket Saurabh. Improved exact algorithms for counting 3- and 4-colorings. In *Computing and Combinatorics, 13th Annual International Conference, COCOON 2007, Banff, Canada, July 16-19, 2007, Proceedings*, pages 65–74, 2007.

11 Fedor V. Fomin, Pinar Heggernes, and Dieter Kratsch. Exact algorithms for graph homomorphisms. *Theory Comput. Syst.*, 41(2):381–393, 2007.

12 Martin Charles Golumbic and Udi Rotics. On the clique-width of some perfect graph classes. *Int. J. Found. Comput. Sci.*, 11(3):423–443, 2000.

13 Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM*, 54(1):1:1–1:24, 2007.

14 P. Hell and Nešetřil. *Graphs and homomorphisms*, volume 28 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, 2004.

15 P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Ser.B*, 48:92–110, 1990.

16 Russell Impagliazzo and Ramamohan Paturi. Complexity of k-SAT. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity, Atlanta, Georgia, USA, May 4-6, 1999*, pages 237–240, 1999.

17 Mikko Koivisto. An $\mathrm{O}^*(2^n)$ algorithm for graph coloring and other partitioning problems via inclusion–exclusion. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 583–590, 2006.

18 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011.

19 László Lovász. Kneser's conjecture, chromatic number, and homotopy. *J. Comb. Theory, Ser. A*, 25(3):319–324, 1978.

20 Johann A. Makowsky and Udi Rotics. On the clique-width of graphs with few P4's. *International Journal of Foundations of Computer Science*, 10(03):329–348, 1999.

21 Jesper Nederlof. Inclusion exclusion for hard problems. Master's thesis, Department of Information and Computer Science, Utrecht University, 2008. URL: `http://www.staff.science.uu.nl/~neder003/MScThesis.pdf`.

22 Patrick Traxler. The time complexity of constraint satisfaction. In *Parameterized and Exact Computation, Third International Workshop, IWPEC 2008, Victoria, Canada, May 14-16, 2008. Proceedings*, pages 190–201, 2008.

23 Magnus Wahlström. New plain-exponential time classes for graph homomorphism. In *Computer Science - Theory and Applications, Fourth International Computer Science Symposium in Russia, CSR 2009, Novosibirsk, Russia, August 18-23, 2009. Proceedings*, pages 346–355, 2009.