On the Complexity of Counting Homomorphisms under Surjectivity Constraints

Jacob Focke

New College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2020

To my grandpa.
From the early days on, you helped me to embark on this journey,
you always supported me on the way,
I wish you were here to see this.

Abstract

Given two graphs G and H, a function h that maps the vertices of G to vertices of H is a (graph) homomorphism if h preserves the edges of G, i.e., whenever two vertices u, v of G share an edge then $h(u)$ and $h(v)$ must share an edge in H. For different H, such homomorphisms represent different structures in G, thereby providing a framework that captures some of the most fundamental combinatorial problems studied in computer science and mathematics. One prominent example is the fact that homomorphisms from a graph G to a triangle (three vertices pairwise connected by edges) correspond to proper 3-colourings of G.

Furthermore, counting homomorphisms has close ties to statistical physics and the computation of partition functions. The complexity of computing and approximating homomorphism counts has therefore drawn a lot of attention over the last four decades. One of the most intriguing open problems in this line of research is determining, for every graph H, the complexity of approximately counting homomorphisms to H. In this thesis, we investigate the complexity of several closely related problems.

Our main objective is to establish complete complexity classifications for large classes of graph homomorphism counting problems. The focus is on counting homomorphisms that satisfy additional properties related to surjectivity. Apart from (unrestricted) homomorphisms, we mainly concentrate on:

- (vertex-)surjective homomorphisms,
- compactions (vertex- and non-loop-edge-surjective homomorphisms),
- retractions (also known as pre-colouring extensions or one-or-all list homomorphisms).

The main contributions of this thesis are as follows:

- **We give a complete complexity dichotomy for exactly counting surjective homomorphisms.** This dichotomy is the same classification that also holds for exactly counting homomorphisms, list homomorphisms and retractions.
- **We give a complete complexity dichotomy for exactly counting compactions.** Notably, this dichotomy is different from the dichotomy for surjective homomorphisms and shows that exactly counting compactions is actually the "hardest" of the aforementioned exact counting problems.

- **We present a collection of approximation-preserving reductions between different homomorphism counting problems.** This includes reductions that establish that approximately counting retractions is at least as hard as approximately counting surjective homomorphisms and also at least as hard as approximately counting compactions. In these reductions we use a Monte Carlo sampling algorithm and this appears to be the first time such an approach has been used to obtain approximation-preserving reductions.
- **We formalise a framework that can be used to generate #BIS-easiness results for approximately counting homomorphisms and retractions.** This framework is based on reduction techniques introduced by Dyer, Goldberg, Greenhill and Jerrum.
- **We give a complete complexity trichotomy for approximately counting retractions to all square-free graphs.** As a consequence, we present separations between the problem of approximately counting homomorphisms and that of approximately counting retractions, as well as between the problem of approximately counting retractions and that of approximately counting list homomorphisms.
- **We present new #BIS-easiness results for approximately counting homomorphisms** and thereby settle the complexity of this problem for a rich class of graphs for which it was previously unresolved.
- **We give a complete complexity dichotomy for counting homomorphisms modulo 2 to all K_4-minor-free graphs.** This confirms a conjecture by Faben and Jerrum for this class of graphs. We use novel global hardness gadgets, which can be used to subsume all previously known classifications. We strengthen the hardness results by ruling out subexponential-time algorithms, assuming rETH.

Acknowledgements

I could not have asked for a more admirable, supportive, dedicated and knowledgable duo of supervisors than I had with Leslie and Standa. Thanks for taking me on as a student, for sharing your time, expertise, and thoughts with me (about research and beyond), and for allowing me to have this extraordinary experience. Also, thanks for introducing me to this great topic, which provided many interesting and fruitful questions to pursue.

Great thanks to all my office mates — without exception they were super pleasant and great fun to share a 10 m^2 office with. Thanks to the "old guard" with John, Dave, Miriam, and Andreas for always offering a friendly ear and good advice, and for the many witty chats, jokes and laughs. I thank the "new guard" with Kuan, James, Andrés, and Marc for a very uplifting office environment and for great times outside the department as well. Especially, I thank Marc, who is co-author to the work on modular counting presented in this thesis, for being such a wholesome person to work, chat, hang out and play board games with.

I thank Steven Kelk and Georgios Stamoulis for an interesting and very enjoyable week of research. I thank Varun Kanade and Rahul Santhanam for taking the time to run my DPhil project assessments — and I thank Rahul Santhanam and Pavol Hell for agreeing to review this (admittedly quite long) thesis.

On several levels I owe thanks to my buddy Lars. His attitude, energy and moral compass are always inspiring — and without Lars' insistence and support I would not have had the idea (or courage) to apply for this DPhil in the first place. Big thanks also to Flo for always taking the time to discuss and figure out bugs in programs (or programming languages, or editors, or...). Furthermore, I thank Flo, Lars and Marc for valuable feedback regarding the introduction of this thesis.

Finally, I am so fortunate to have the never-ending support, love and understanding of my parents, Jana and Thomas, my sister Theresa, and my brother Julius (Thanks for frequently distracting me from work!). So viel verdanke ich auch meinen Großeltern Hannelore und Eberhard, die mich mein Leben lang begleiten und mir Hilfe und Vorbild sind. Last but not least, I thank Wiebke so much for her sacrifices, endurance and encouragement, and for all the great experiences we share.

Declaration

If not explicitly stated otherwise, all contributions (in form of lemmas, theorems, corollaries etc.) presented in this thesis are new results. Parts of this thesis have been published in peer-reviewed academic journals and conference proceedings. Some parts are available as preprints and are currently submitted to conferences/journals. Chapters 2, 3 and 4 are based on the following papers, which are co-authored with my supervisors Leslie Ann Goldberg and Standa Živný.

[57] Jacob Focke, Leslie Ann Goldberg, and Stanislav Živný. The complexity of counting surjective homomorphisms and compactions. *SIAM Journal on Discrete Mathematics*, 33(2):1006–1043, 2019.
   - A preliminary version of this work appeared in the Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, pp. 1772-1781.

[58] Jacob Focke, Leslie Ann Goldberg, and Stanislav Živný. The complexity of approximately counting retractions. *ACM Transactions of Computation Theory*, 12(3):Art. 15, 43, 2020.
   - A preliminary version of this work appeared in the Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, pp. 2205-2215.

[59] Jacob Focke, Leslie Ann Goldberg, and Stanislav Živný. The complexity of approximately counting retractions to square-free graphs. *arXiv preprint arXiv:1907.02319*, 2019.

Chapter 5 is based on the following preprint, which is co-authored with Leslie Ann Goldberg, Marc Roth, and Standa Živný.

[56] Jacob Focke, Leslie Ann Goldberg, Marc Roth, and Stanislav Živný. Counting homomorphisms to K_4-minor-free graphs, modulo 2. *arXiv preprint arXiv:2006.16632*, 2020.
   - A preliminary version of this work will appear in the Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2021.

Contents

List of Symbols and Notation

Chapter 1

(ε, δ)-approximation	*ε-close approximation with error probability δ*	p. 17
$A \leq B$	*A polynomial-time Turing reduces to B*	p. 10
$A \leq_{\mathrm{AP}} B$	*A AP-reduces to B*	p. 17
AP-reduction	*Approximation-preserving reduction*	p. 17
#BIS	*Counting independent sets in a bipartite graph*	p. 18
#COMP(H)	*Compaction counting problem*	p. 13
COMP(H)	*Compaction decision problem*	p. 10
CSP($\mathcal{L}$)	*Constraint satisfaction decision problem with template $\mathcal{L}$*	p. 12
Cactus graph	*Every edge belongs to at most* 1 *cycle*	p. 27
Caterpillars	*Distinguished class of irreflexive trees*	p. 19
Compaction		p. 10
Configuration	*Function that assigns spins to particles of a spin system*	p. 2
CSP	*Constraint satisfaction problem*	p. 12
FP	*Class of polynomial-time computable functions*	p. 14
FPAS	*Fully polynomial approximate sampler*	p. 20
FPRAS	*Fully polynomial RAS*	p. 17
Girth	*Length of shortest cycle*	p. 21
Graph	*Undirected, might have loops but no parallel edges*	p. 9
$\mathcal{H}_{\mathrm{BIS}}$	*Distinguished class of* #BIS*-easy graphs*	p. 22
HOM($H, \mathcal{L}$)	*Homomorphism decision problem with template of lists*	p. 9
HOM(H)	*Homomorphism decision problem*	p. 9
#HOM($H, \mathcal{L}$)	*Homomorphism counting problem with template of lists*	p. 85
#HOM(H)	*Homomorphism counting problem*	p. 13
$\oplus$HOM(H)	*Homomorphism parity problem*	p. 25
H-colouring	*Homomorphism to H*	p. 5
$H(\sigma)$	*Hamiltonian of configuration σ*	p. 2
H^*	*Involution-free reduction of H*	p. 26
Hamiltonian	*Overall energy of a configuration*	p. 2
Homomorphism	*Function that preserves edges*	p. 3
Imp	*Binary Boolean relation "Implies"*	p. 12
Independent set	*Pairwise non-adjacent set of vertices*	p. 5
Involution	*Automorphism of order ≤ 2*	p. 25
Involution-free graph	*Graph without non-trivial involutions*	p. 25

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 1

Introduction and Contributions

But ever since the dawn of civilization, people have not been content to see events as unconnected and inexplicable. They have craved an understanding of the underlying order in the world. Today we still yearn to know why we are here and where we came from. Humanity's deepest desire for knowledge is justification enough for our continuing quest.

– Stephen Hawking, *A Brief History of Time (1988)*

Contents

Organisation of this Chapter

As a warm-up, Section 1.1 is targeted at a more general audience and is written for readers with STEM subject background that are possibly unfamiliar with (some aspects of) the topic of this thesis, or even complexity theory and theoretical computer science. We make an effort to provide some explanation for why researchers invest years of study into concepts such as homomorphisms. Afterwards, in Sections 1.2 to 1.5, we introduce the topics more formally, give an overview of relevant previous work, and state the new results that we contribute to the existing body of work. Section 1.6 explains how this thesis is structured and which tools have been introduced to improve its readability.

1.1 Context and Motivation

1.1.1 From Statistical Physics to Graph Homomorphisms

In statistical physics, materials such as ferromagnets, spin glasses, crystals, or gases can be modelled using a graph G and an integer $q \geq 2$. A graph G consists of a set of nodes/vertices $V(G)$ and a set of edges $E(G)$ connecting these vertices. Particles or sites of a material are represented by the vertices of G. The edges of G represent interactions between particles. Each particle can be in one of q states — these states correspond, for instance, to the moments or *spins* of atoms, or the occupation of sites by gas molecules. A *configuration* of such a spin system is a function $\sigma\colon V(G) \to \{1, \dots q\}$ that assigns a spin to each particle. Well-known examples are the Ising or the Potts model [102, 130] for spin systems with $q = 2$ or $q \geq 3$ spins, respectively.

The *interaction energies* between spins can be represented by a symmetric $q \times q$ matrix K, where an entry $K_{i,j}$ measures the interaction energy between the spins i and j. The overall energy of a configuration σ is then given by the so-called *Hamiltonian* $H(\sigma) = \sum_{\{u,v\} \in E(G)} K_{\sigma(u),\sigma(v)}$, which is simply the sum of all the individual interaction energies. It is assumed that the probability of the system being in a particular configuration σ is $\exp(-H(\sigma)/cT)/Z_K(G)$ according to the Gibbs distribution. Here, T is the temperature of the system, c is the Boltzmann constant, and $Z_K(G)$ is the *partition function*, which we will define momentarily. Note that the higher the energy $H(\sigma)$ the lower the probability of the corresponding configuration σ. So, the system tends to be in low-energy configurations. (For more information about the Gibbs distribution and its ties to statistical physics we refer to the book by Georgii [67].) Here is the definition of the partition function, it uses Ω to denote the space of all configurations:

$$Z_K(G) = \sum_{\sigma \in \Omega} \exp(-H(\sigma)/cT).$$

The partition function is the normalising factor of the Gibbs distribution. Knowing the value of the partition function, one can compute the probability of the system being in a certain configuration. Furthermore, a number of thermodynamic variables (for instance the entropy or the total energy) can be expressed in terms of the partition function. Thus, the partition function forms the link between the microscopic interactions of particles or sites and the macroscopic thermodynamic properties of a system. Being able to compute the partition function of a system is therefore desirable.

We will use the following definition to slightly rearrange the expression of $Z_K(G)$. For a $q \times q$ matrix A, let $Z_A(G) = \sum_{\sigma\in\Omega} \prod_{\{u,v\}\in E(G)} A_{\sigma(u),\sigma(v)}$. Now consider the matrix A with $A_{i,j} = \exp(-K_{i,j}/cT)$. Using the definition of $H(\sigma)$, by factoring out a term for each summand of $H(\sigma)$, note that

$$Z_K(G) = \sum_{\sigma\in\Omega} \prod_{\{u,v\}\in E(G)} \exp(-K_{\sigma(u),\sigma(v)}/cT) = \sum_{\sigma\in\Omega} \prod_{\{u,v\}\in E(G)} A_{\sigma(u),\sigma(v)} = Z_A(G).$$

This is the reason why partition functions are also sometimes referred to as "sums of products".

In the model we just described, the entries of A are positive values. This corresponds to what are called "soft" constraints — for each pair of interacting particles u, v and each pair of spins i and j there is a non-zero probability that u has spin i and v has spin j. This probability depends on the value $A_{i,j}$ (the smaller $A_{i,j}$ the lower the probability).

However, there are also models that work with so-called "hard" constraints, which means that certain configurations or certain interactions are entirely forbidden. For example, in the *hard-core* gas model [35] it is impossible that neighbouring sites are both occupied by a gas particle (as otherwise these two gas particles would overlap). Therefore, configurations with such an interaction are forbidden. Other examples for the use of hard constraints are the Widom-Rowlinson gas model [150], and the Beach model [19]. Forbidden interactions can be modelled by setting the corresponding matrix entry of A to zero (e.g., in the hard-core gas model the forbidden interaction would be "occupied, occupied"). Each configuration that uses a forbidden interaction has a zero-factor in the corresponding product and therefore does not contribute to the partition function. If, in addition, we suppose that all allowed configurations are equally likely, i.e., they are uniformly distributed, then we can suppose that all entries of A are either 0 (interaction forbidden) or 1 (interaction allowed). Consequently, forbidden configurations do not contribute to the partition function and each allowed configuration contributes a term of 1 to the sum, which means that $Z_A(G)$ is precisely the number of allowed configurations of G.

In this setting, A is a symmetric $q \times q$ matrix with entries in $\{0, 1\}$; it can therefore be interpreted as the adjacency matrix of a graph H, i.e., H has q vertices and an edge $\{i, j\}$ if and only if $A_{i,j} = 1$. A *homomorphism* from G to H is a function from the vertices of G to the vertices of H that preserves edges, i.e., $\{u, v\} \in E(G)$ implies $\{\sigma(u), \sigma(v)\} \in E(H)$. Since $\{\sigma(u), \sigma(v)\} \in E(H)$ is equivalent to $A_{\sigma(u),\sigma(v)} = 1$, a configuration σ contributes a term of 1 to $Z_A(G)$ if and only if σ is a homomorphism from G to H. So, if A is the adjacency matrix of a graph H, $Z_A(G)$ is precisely the number of homomorphisms from G to H. Inspired by these connections to both physics and graph homomorphisms, $Z_A(G)$ is also named *partition function* or *homomorphism function*.

$$\begin{array}{c} \\ g_1 \\ g_2 \\ g_3 \\ u \end{array}\begin{array}{c} \begin{array}{cccc} g_1 & g_2 & g_3 & u \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

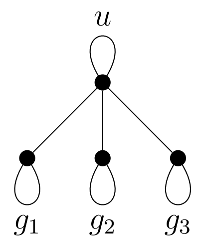


Figure 1.1: The matrix A_3 (on the left) and the graph WR_3 (on the right).

Let us consider an example. In the q-particle Widom-Rowlinson gas model it is assumed that space is divided into microscopic cells, so-called *sites*. There are q different gases that "try" to occupy these sites. So, the different sites can be represented by vertices of a graph G for which two vertices are connected by an edge whenever the corresponding sites are adjacent. Each site can be in one of $q+1$ states — it is either occupied by particles of one of the q gases, or otherwise it is unoccupied. It is assumed that particles of different gases never occupy adjacent sites. For the case $q = 3$ with three gases g_1, g_2, g_3, and an "unoccupied" state u, the corresponding matrix A_3 of forbidden or allowed interactions is given in Figure 1.1 on the left. Note that, for instance, g_1 is allowed to interact with itself since $A_{g_1,g_1} = 1$, which means that g_1 can occupy adjacent sites, whereas $A_{g_1,g_2} = 0$, which means that a site that is occupied by g_1 cannot be adjacent to a site that is occupied by g_2. Interpreted as an adjacency matrix, A_3 defines the graph WR_3 illustrated in Figure 1.1. So, homomorphisms from the graph G to the graph WR_3 are nothing more or less than configurations of the 3-particle Widom-Rowlinson gas model.

Coinciding phase transitions are another intriguing aspect of the connections between particle models from physics and the computational complexity of computing or approximating partition functions. A phase transition is a jump discontinuity of some function (or one of its derivatives), i.e., a sudden change of behaviour when changing control parameters such as temperature or pressure. The previously mentioned physical systems showcase such phase-transition behaviour. The spins of particles in ferromagnetic metals at high temperatures do not tend to form a magnetic field. However, below a certain threshold temperature spontaneous magnetisation occurs. Similarly, the liquid-vapour phase transition in gases is a sudden change in density at a certain threshold pressure. In the complexity theory of computational problems, phase transitions are typically transitions from "easily solvable" to "hard" computational problems at a certain threshold value of a parameter of the problem. Remarkably, there are examples for which these physical phase transitions align with the jumps in complexity of related computational problems. A prominent example are the works of Weitz [149], Sly [136], Sly, Sun [137], and Galanis, Štefankovič, Vigoda [65], who show that for a certain variant of the hard-core gas model, the liquid-vapour phase transition occurs precisely for the same parameter values as the phase transition of the complexity of approximating the corresponding partition function.

1.1.2 Graph Homomorphisms in Computer Science

In this section we motivate the study of homomorphisms from a computer science point of view. If we ask the question "Is there a homomorphism from a given graph G to a graph H?", and think about how hard it would be to answer this question for different graphs, then we enter the realm of computer science. Consider the following framework. For a fixed graph H, Hom(H) is the problem that takes as input some graph G and asks whether there exists a homomorphism from G to H. Depending on the graph H, Hom(H) asks a different question about its input graph.

Figure 1.2: The graphs K_3 (on the left), IO (in the middle) and P_4^* (on the right).

For example, consider the graph K_3 given in Figure 1.2. If we think of the vertices of K_3 as different colours red (r), blue (b) and green (g), then a homomorphism from a graph G to K_3 assigns a colour to each vertex of G in a way that vertices of G that share an edge are mapped to colours that share an edge in K_3 — this means that adjacent vertices in G never have the same colour. Such a colouring is known as *proper 3-colouring* of the vertices of G. Hence the question "Is there a homomorphism from G to K_3?" is exactly the same question as "Is there a proper 3-colouring of the vertices of G?". More generally, for each positive integer k, homomorphisms from a graph G to the complete graph on k vertices correspond to proper k-colourings of the vertices of G. Determining the *chromatic number* of a graph G, i.e., the smallest integer k such that G is proper k-colourable, is one of the most fundamental computational problems, and as such it is also one of Karp's original 21 NP-complete problems [106]. It has far-reaching applications and appears as a sub-problem in many other computational challenges, such as scheduling and timetabling, assigning (radio) frequencies, allocating compiler registers, and even solving sudoku puzzles [115, 120]. Since vertex colouring is such a prominent problem, Hom(H) is also known as the *H-colouring* problem.

However, other similarly important structures are covered by the homomorphism framework as well. Consider the graph IO depicted in Figure 1.2. It has two vertices labelled "In" and "Out". As $\{\mathrm{In}, \mathrm{In}\}$ is not an edge of IO, no two adjacent vertices in G can both be mapped to "In" by a homomorphism from G to IO. Thus, the set of vertices that is mapped to "In" forms an *independent set* of G (also known as *stable set*), that is, a set in which no two vertices are adjacent. Moreover, the homomorphisms from G to IO are in one-to-one correspondence with the independent sets of G. Independent sets appear in a vast number of computational problems, for instance in scheduling, coding theory, social network analysis [135], or chemistry [125, pp. 35-36]. Independent sets are also the complement of cliques (and they are in that sense equivalent) — the corresponding problem of deciding whether a graph has a clique larger than a certain size is another one of Karp's 21 NP-complete problems.

As our last example, say each vertex of a graph G has two labels. The first label is either positive (+) or negative (−), and the second label is either privileged (♕) or unprivileged (♙). We want to label the vertices in such a way that a positive vertex can only be adjacent to a negative vertex if both of these vertices are privileged. The problem we have described is precisely the task of finding a homomorpism from G to the graph P_4^* from Figure 1.2 (with the corresponding vertex labels). So we see that various combinatorial structures and problems can be modelled using the homomorphism framework. Homomorphisms to the graph P_4^* actually also have a meaning in statistical physics as they correspond to configurations of the so-called Beach model [19, 89] (in the same way as homomorphisms to the graph WR_3 correspond to configurations of the 3-particle Widom-Rowlinson model, as pointed out in Section 1.1.1).

Rather than analysing the complexity of $\textsc{Hom}(H)$ for each H individually, we are interested in a complexity classification for all H. Such a classification was achieved in a seminal work by Hell and Nešetřil [96]. In order to state this result we need the notion of the complexity classes P and NP. Informally, P is the class of decision problems (problems that can be stated as a yes-no question) for which a solution can be easily found — whereas NP is the class of decision problems for which a given solution can be easily verified. The question, whether one can easily find a solution for every problem for which a given solution is easily verifiable, is the famous P versus NP problem, and it is widely believed that this is not the case and therefore $\mathrm{P} \neq \mathrm{NP}$. A problem in NP is said to be NP-*complete* if it is "at least as hard to solve" as every problem in NP. Under the assumption that $\mathrm{P} \neq \mathrm{NP}$, an NP-complete problem cannot be in P. We can now state the result by Hell and Nešetřil.

Theorem 1.1 ([96, Theorem 1]). *Let H be a graph. If H is either bipartite or has a loop then* $\textsc{Hom}(H)$ *is in* P*. Otherwise it is* NP*-complete.*

Such a partition into two complementing complexity classes is a so-called *complexity dichotomy*. Such dichotomies (or, more generally, classifications with a finite number of complexity classes) are interesting since Ladner's Theorem [112] states that given $\mathrm{P} \neq \mathrm{NP}$ there exists an infinite hierarchy of disjoint intermediate complexity classes within NP. So this is another point of view from which the homomorphism framework is remarkable: It is an example of a large and expressive class of problems within NP that does not contain such an infinite complexity hierarchy.

Over the years, homomorphisms have been studied in various settings, there are several books on the topic (e.g. [98, 118]), and they can be encountered in many areas of complexity theory and beyond, for instance in extremal graph theory [10, 11], parameterised complexity [29, 55], database theory [23, 86, 110], statistical physics [13, 14], constraint satisfaction problems [53,99], and holant problems with links to quantum computing [1, 22]. Homomorphisms have been studied in the context of more general notions of graphs, e.g., directed graphs [40, 88, 93], signed graphs [8, 124] or weighted graphs [21, 72]. There is also a lot of research on homomorphisms that fulfil additional properties, such as different local [54] or global constraints [6, 18], which leads us to the next section.

1.1.3 Homomorphisms under Surjectivity Constraints

We have described that the homomorphism problem covers questions like

> Can we colour the vertices of a graph with the colours red, blue, and green in a way that no two adjacent vertices have the same colour?

A very natural, closely related question is

> Can we colour the vertices of a graph with the colours red, blue, and green in a way that no two adjacent vertices have the same colour *and we actually use each colour at least once*?

In this particular case, it is clear that the two questions are equivalent: If G has at least 3 vertices and can be properly 3-coloured, then it can also be properly 3-coloured using all of the 3 colours (and vice versa). However, for other H-colouring problems the question is more interesting. The structure we are looking for is a *surjective homomorphism.* For graphs G and H, a function from $V(G)$ to $V(H)$ is *surjective* if every element of $V(H)$ is the image of at least one element of $V(G)$. So, intuitively, a surjective homomorphism from G to H is a homomorphism that "uses" all of the colours/vertices of H. In general, it is not true that the existence of a homomorphism from G to H implies the existence of a surjective homomorphism, as we demonstrate with the following example. Consider the graphs depicted in Figure 1.3. Suppose that G is a 5-vertex star and H is a looped square. A homomorphism from G to H has to map the center v of the 5-vertex star to some "corner" of the looped square H, say a. Since edges have to be preserved by a homomorphism, all remaining vertices of G have to be mapped to one of a, b, or d. However, c can never be "used" as it does not share an edge with a. So, there are homomorphisms from the 5-vertex star to the looped square, but no such homomorphism is surjective (i.e., uses all vertices of the looped square).

Figure 1.3: 5-vertex star (on the left) and looped square (on the right).

A *compaction* from G to H is a surjective homomorphism that also "uses" all of the non-loop edges of H. Compactions are the structure we are looking for when we ask the question

> Can we colour the vertices of a graph with the colours red, blue, and green in a way that no two adjacent vertices have the same colour, and there

> is at least one red-blue edge, at least one red-green edge and at least one blue-green edge?

Another interesting related question is the task of finding what is called a *pre-colouring extension*:

> Given a graph G for which some of the vertices are already coloured red, blue, or green, can we colour the remaining vertices in a way that no two adjacent vertices have the same colour?

This might remind the reader of sudoku puzzles, for which some numbers are already filled in... It is not immediately obvious how such a pre-colouring extension relates to surjectivity. To see this, we introduce list homomorphisms and retractions.

If for each vertex v of G we specify a so-called *list* S_v of vertices of H, then a *list homomorphism* maps every vertex v of G to a vertex from the corresponding list S_v. For example, one could consider list homomorphisms from the Petersen graph (see Figure 1.4 on the left) to the graph K_3 (Figure 1.2). These homomorphisms then correspond to proper 3-colourings of the Petersen graph, where some vertices have a restricted set of allowed colours.

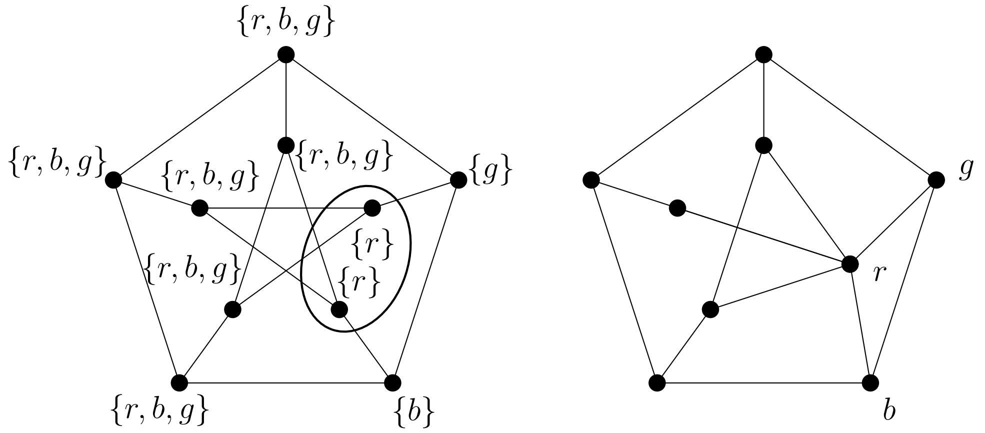


Figure 1.4: Petersen graph with lists (on the left) and Petersen graph with vertices identified according to single-vertex lists (on the right).

If every list contains either just a single vertex or otherwise all of the vertices of H, then such a list homomorphism is called *retraction*. We can observe that a retraction is nothing more or less than a pre-colouring extension since vertices with a single-vertex list can only be coloured with one particular colour and therefore can be considered as already coloured, and vertices with a list of the form $V(H)$ correspond to the vertices that have yet to be coloured using any of the colours.

The connection between retractions/pre-colouring extensions and surjective homomorphisms becomes clear when we use a third equivalent definition. A retraction can also be defined as a homomorphism from a graph G to an induced subgraph H of G that is the identity on H, i.e., that maps each vertex of H to itself. This subgraph version of the definition is equivalent to the aforementioned list version since one can identify all of the vertices in G that have the same single-vertex list $\{u\}$ with each other to form a single vertex u of the subgraph H. Homomorphisms from this new graph to H are retractions in the sense of the subgraph version of the definition. For the Petersen graph in our example, the modified graph is displayed in Figure 1.4 on the right. Note that it contains K_3 as an induced subgraph. Formally, this equivalence was shown by Feder and Hell [49]. The details are not important at this point, however, it shows that retractions can be interpreted as homomorphisms that map *surjectively* from a graph G onto a subgraph H of G.

In Section 1.2, we will outline the substantial body of research on such homomorphisms under surjectivity constraints, and on the problems of deciding when such homomorphisms exist. The goal of this thesis is to determine the computational complexity of counting homomorphisms, surjective homomorphisms, compactions, and (most prominently) retractions. We will study these problems under different counting models, namely exact counting, approximate counting, and modular counting. We introduce these models in Sections 1.3, 1.4, and 1.5, respectively.

1.2 Preliminaries — Graphs, Homomorphisms, and Decision Problems

If not stated otherwise, a *graph* is assumed to be undirected and to have no parallel edges. A *(self-)loop* is an edge from a vertex to itself. Two classes of graphs are of particular interest. A graph is *irreflexive* if it does not have any self-loops, and it is *reflexive* if every vertex has a self-loop.

We will now formally define the homomorphism decision problems that we introduced informally in Section 1.1.3. Given graphs G and H together with a set of *lists* $\mathbf{S} = \{S_v \subseteq V(H) \mid v \in V(G)\}$, a homomorphism from $(G, \mathbf{S})$ to H is a homomorphism h from G to H such that for each $v \in V(G)$ we have $h(v) \in S_v$. Given a set S, let $\mathcal{P}(S)$ denote its power set. We consider the following problem, which takes as parameter a graph H and a set of lists $\mathcal{L} \subseteq \mathcal{P}(V(H))$.

Name: $\textsc{Hom}(H, \mathcal{L})$.
Input: An irreflexive graph G and a collection of lists $\mathbf{S} = \{S_v \in \mathcal{L} \mid v \in V(G)\}$.
Output: Is there a homomorphism from $(G, \mathbf{S})$ to H?

The graph G in the input is assumed to be irreflexive because this is standard in the field, and because it makes hardness results stronger. Some cases are of special interest and we introduce additional notation for these.

- If $\mathcal{L} = \{V(H)\}$ then $\textsc{Hom}(H, \mathcal{L})$ is the *Homomorphism decision problem* and we also write $\textsc{Hom}(H)$ to denote this problem.

- If $\mathcal{L} = \{S \subseteq V(H) \mid |S| \in \{1, |V(H)|\}\}$ then $\textsc{Hom}(H, \mathcal{L})$ is the *Retraction decision problem* and we also write $\textsc{Ret}(H)$.

- If $\mathcal{L} = \mathcal{P}(V(H))$ then $\textsc{Hom}(H, \mathcal{L})$ is the *List homomorphism decision problem* and we also write $\textsc{LHom}(H)$.

So, the list homomorphism problem is a natural generalisation of the retraction problem, which in turn is a generalisation of the homomorphism problem.

A homomorphism h from G to H is said to *use a vertex* $v \in V(H)$ if there is a vertex $u \in V(G)$ such that $h(u) = v$. Then, h is *surjective* if it uses every vertex of H. Similarly, a homomorphism h is said to *use an edge* $\{v_1, v_2\} \in E(H)$ if there is an edge $\{u_1, u_2\} \in E(G)$ such that $h(u_1) = v_1$ and $h(u_2) = v_2$, and h is a *compaction* if it uses every vertex of H and every non-loop edge of H. Here are the corresponding decision problems that take as parameter a graph H.

Name: $\textsc{SHom}(H)$.
Input: An irreflexive graph G.
Output: Is there a surjective homomorphism from G to H?

Name: $\textsc{Comp}(H)$.
Input: An irreflexive graph G.
Output: Is there a compaction from G to H?

We have already seen that the complexity of $\textsc{Hom}(H)$ is classified for all H (Theorem 1.1). Similarly, a complete classification is known for $\textsc{LHom}(H)$ due to Feder, Hell and Huang [50]. The complexity of $\textsc{Ret}(H)$ has only recently been resolved as a consequence of two independent proofs of the CSP dichotomy by Zhuk [152] and Bulatov [16]. (More information about the CSP dichotomy will be given in the subsequent paragraph on "Constraint Satisfaction Problems".) Intriguingly, the complexities of the problem classes $\textsc{SHom}(H)$ and $\textsc{Comp}(H)$ are still far from being resolved completely — despite a lot of research on these problems [80–82, 114, 119, 147]. For irreflexive graphs, compactions coincide with the notion of a "homomorphic image", as used, e.g., in [90, 92]. It appears that NP-completeness results for surjective homomorphism and compaction problems are hard to come by and even resolving their complexity for all graphs of size at most four was far from swift [119, 144, 146].

Decision Complexity Landscape In their survey on surjective homomorphisms, Bodirsky, Kára and Martin [6] show a relationship in form of a hierarchy between the different homomorphism problems. Given computational problems A and B, we write $A \leq B$ if there is a polynomial-time Turing reduction from A to B. We write $A \equiv B$ if both $A \leq B$ and $B \leq A$.

Proposition 1.2 ([6, Proposition 1])**.** *Let H be a graph. Then*

$$\textsc{Hom}(H) \leq \textsc{SHom}(H) \leq \textsc{Comp}(H) \leq \textsc{Ret}(H) \leq \textsc{LHom}(H).$$

We will refer back to Proposition 1.2 when we establish complexity hierarchies for the corresponding counting problems. It is a long-standing unresolved conjecture that $\textsc{Comp}(H) \equiv \textsc{Ret}(H)$ for each graph H (attributed to Peter Winkler, see, e.g., [6, Conjecture 2]). From the current knowledge it is even possible that $\mathrm{SH}\textsc{om}(H) \equiv \textsc{Comp}(H) \equiv \textsc{Ret}(H)$ for each graph H.

Proposition 1.2 shows that, for each graph H, $\mathrm{SH}\textsc{om}(H)$ is at least as hard as $\textsc{Hom}(H)$. However, there is evidence that, for some graphs H, $\mathrm{SH}\textsc{om}(H)$ is actually harder than $\textsc{Hom}(H)$. Suppose the graph H consists of an irreflexive clique on 3 vertices (a triangle) together with a single isolated looped vertex. $\textsc{Hom}(H)$ is trivially solvable because a homomorphism can map all vertices of an instance G to the looped vertex in H. However, this argument does not hold for surjective homomorphisms, and it turns out that $\mathrm{SH}\textsc{om}(H)$ is actually NP-complete in this case [6]. (Intuitively, the 3-vertex clique in H can be used to model proper vertex 3-colouring.) Thus, $\textsc{Hom}(H)$ and $\mathrm{SH}\textsc{om}(H)$ are known to be separated under the assumption that $P \neq$ NP. (We say that two homomorphism problems A and B are *separated* if there exists a parameter H for which A and B have different complexity, subject to some complexity theory assumptions.) Similarly, $\textsc{Ret}(H)$ and $\mathrm{LH}\textsc{om}(H)$ are separated given $P \neq$ NP [6].

Retractions Retractions will accompany us throughout this thesis, we will encounter them in every chapter, and they are the main focus of Chapters 3 and 4. In the context of graphs and relational structures, retractions have been studied over a long period of time [53,94,95,100,129], and consequently the corresponding decision problem $\textsc{Ret}(H)$ is also well-studied [52,98,145,146,148]. In the context of topological spaces, retractions have been originally studied by Borsuk as early as the 1930s [12]. Retractions are also known under the names *one-or-all list homomorphisms* (e.g. [49,50]) and *pre-colouring extensions* (e.g. [5,7,51,103,111,121,140]). Hell and Nešetřil's review article [99] gives an even more extensive list of related work.

The definition of $\textsc{Ret}(H)$ given at the beginning of Section 1.2 describes retractions in terms of list homomorphisms. In some works, a retraction is alternatively defined as a homomorphism h from G to an induced subgraph H of G such that, for each $v \in V(H)$, $h(v) = v$. For us, the definition in terms of lists is more convenient as it does not restrict the class of input graphs and, for instance, it allows us to consider connected graphs G even when H has multiple connected components. The two decision problems that correspond to these two different definitions of retractions are known to be interreducible by polynomial-time Turing reductions due to Feder and Hell [49, Theorem 4.1]. We note that the reductions presented in their work are *parsimonious* (which means that they preserve the number of solutions) and therefore this interreducibility will also extend to the retraction counting problem which we define in Section 1.3.

Even though the complexity of $\textsc{Ret}(H)$ is completely classified using algebraic criteria, a graph theoretical classification that would allow us to easily determine the complexity for each particular parameter graph H is not known. However, Feder, Hell, Jonsson, Krokhin and Nordh [52, Corollary 4.2, Theorem 5.1] give the following graph theoretical result, which we will use later. A *pseudotree* is a graph with at most one cycle. A graph H is called *loop-connected* if, for every connected component C of H, the looped vertices in C induce a connected subgraph of C.

Theorem 1.3 ([52]). *Let H be a pseudotree. Then* $\textsc{Ret}(H)$ *is* NP*-complete if any of the following hold:*

- *H is not loop-connected,*
- *H contains a cycle of size at least* 5*,*
- *H contains a reflexive cycle of size* 4 *or*
- *H contains an irreflexive cycle of size* 3*.*

Otherwise $\textsc{Ret}(H)$ *is in* P*.*

Constraint Satisfaction Problems Some of our techniques will make use of the framework of constraint satisfaction problems, which is closely related to the class of homomorphism problems.

Let $\mathcal{L}$ be a relational structure, i.e., a set of relations on a finite *domain* D ($\mathcal{L}$ is also called *constraint language* or *template*). For a set of variables X, a *constraint* on X from $\mathcal{L}$ is of the form $R(Y)$, where Y is a tuple of variables from X and R is a $|Y|$-ary relation from $\mathcal{L}$. An assignment $\sigma\colon X \to D$ *satisfies* the constraint $R(Y)$ if $\sigma(Y) \in R$ (where $\sigma(Y)$ is evaluated componentwise). The *constraint satisfaction problem* (CSP) with parameter $\mathcal{L}$ is defined as follows.

Name: CSP($\mathcal{L}$).
Input: A set of variables X and a set C of constraints on X from $\mathcal{L}$.
Output: Is there an assignment $\sigma\colon X \to D$ that satisfies all constraints in C?

For example, consider the Boolean domain $D = \{0, 1\}$ and suppose that $\mathcal{L}$ contains only the binary Boolean relation $\mathrm{Imp} = \{(0, 0), (0, 1), (1, 1)\}$. Then a constraint is of the form $\mathrm{Imp}(x, y)$ and ensures that, in any satisfying assignment σ, we have $\sigma(x) \implies \sigma(y)$, that is, if $\sigma(x) = 1$ then $\sigma(y) = 1$.

As a graph can be expressed as a relational structure with a single binary, symmetric relation, constraint satisfaction problems generalise homomorphism problems. In more detail, for a graph H, let $\mathcal{L}_H$ be the constraint language with domain $D = V(H)$ that contains only the relation $R_E = \{(u, v) \mid \{u, v\} \in E(H)\}$. Then CSP($\mathcal{L}_H$) takes as input a set X together with a set of constraints C of the form $R_E(a, b)$ where (a, b) is a pair of elements from X. Thus, an instance of CSP($\mathcal{L}_H$) is also a relational structure with a single binary relation (defined by the sets (a, b) that are subject to a constraint from C)[1], and can be interpreted as a graph G. So, a satisfying assignment of CSP($\mathcal{L}_H$) is nothing more or less than a homomorphism from the corresponding graph G to the graph H. Therefore, the problems CSP($\mathcal{L}_H$) and $\textsc{Hom}(H)$ are equivalent in the sense that they are interreducible by parsimonious polynomial-time Turing reductions.

In general, solutions of the constraint satisfaction problem are essentially homomorphisms between relational structures and many of the insights about graph homomorphisms extend to homomorphisms between relational structures. CSPs can

[1]Since R_E is symmetric we can assume without loss of generality that the binary relation defined by the constraints in C is also symmetric.

also be used to model retraction and list homomorphism problems (by including in $\mathcal{L}$ a unary relation for each allowed list), see e.g. [99] for further exposition.

In a famous conjecture, Feder and Vardi [53] hypothesised that for every constraint language $\mathcal{L}$ the problem $\mathrm{CSP}(\mathcal{L})$ is either in P or, otherwise, is NP-complete. They also established that, in a sense, this class would then be maximal with the property of exhibiting such a complexity dichotomy. After being open for many years, the conjecture was finally confirmed in 2017 independently by Bulatov [16] and Zhuk [152]. This dichotomy is especially intriguing in light of the well-known fact that, given $\mathrm{P} \neq \mathrm{NP}$, there exist problems in NP that are neither in P nor NP-complete [112].

Summarising the bulk of research on CSPs is beyond the scope of this thesis. A good overview can be found in the survey by Hell and Nešetřil [99]. However, a small selection of related more recent work shall be mentioned. Larose [113] surveys the work on CSPs with templates that represent directed graphs (digraphs). The survey has a special focus on CSPs with additional unary constraints, such as the digraph retraction problem. Another line of work explores surjective CSPs, that is, a version of constraint satisfaction problems that asks for surjective satisfying assignments [24, 26, 151]. Recently, surjectivity has also been investigated in the context of *valued* constraint satisfaction optimisation problems [60, 122].

1.3 Counting Homomorphisms Exactly

Valiant [141] introduced the complexity class #P when investigating the computational complexity of determining the permanent of a matrix. Subsequently, a lot of research has been invested into the complexity of counting problems. One important branch of this research concentrates on counting homomorphisms.

#P can be seen as the counting analogue to NP, that is, it contains all functions f that compute the (exact) number of accepting paths of a nondeterminstic polynomial-time Turing machine. Intuitively, if a decision problem A in NP asks to determine whether a solution exists, then the corresponding (exact) counting problem — we usually write #A — asks to determine the exact number of solutions. For example consider the homomorphism counting problem.

Name: $\#\textsc{Hom}(H)$.
Input: An irreflexive graph G.
Output: The number of homomorphisms from G to H.

Analogously, we define counting versions of other homomorphism problems, such as $\#\textsc{SHom}(H)$, $\#\textsc{Comp}(H)$, $\#\textsc{Ret}(H)$ and $\#\textsc{LHom}(H)$.

A problem #A in #P is #P-*complete* if every problem in #P reduces to #A by a polynomial-time Turing reduction. Toda's theorem [139] states that the entire polynomial-time hierarchy (PH) is contained in $P^{\#\mathrm{P}}$, i.e., every problem in the polynomial-time hierarchy can be solved with a #P oracle. Since P^{NP} is "only" the second level of the polynomial-time hierarchy, under standard complexity theory assumptions (specifically, if PH does not collapse to the second level), #P-complete counting problems are substantially harder than every decision problem in NP.

It appears to be a good intuition that the counting version of an NP-complete decision problem is #P-complete — this relation seems to be true for all natural problems, in fact, there are no known exceptions. However, there is currently no known proof for this statement, so it is not clear whether it actually is a theorem. On the contrary, an interesting aspect of #P-complete counting problems is that the underlying decision problem might be polynomial-time solvable. For example, determining whether a graph has an independent set is trivial, whereas counting all independent sets is known to be #P-complete [43].

FP is the class of functions that can be computed in polynomial time. Due to Toda's theorem, FP = #P would imply P = NP = PH (among other things) and it is therefore extremely unlikely that a #P-complete problem is in FP.

An important result by Dyer and Greenhill [42, 44] establishes a complexity dichotomy for counting homomorphisms and gives an explicit graph-theoretic criterion that separates tractable (FP) cases from intractable (#P-complete) cases: The complexity of counting the homomorphisms from an input graph G to a fixed graph H is polynomial-time solvable if every connected component of H is either an irreflexive complete bipartite graph or a reflexive complete graph. For all remaining graphs H, they show that #Hom(H) is #P-complete.

There are some interesting aspects to this classification. We can see that classes of counting problems might also exhibit nice dichotomies. This is remarkable since Ladner's theorem was extended by Schöning [133] to the class #P. Schöning's theorem shows that given FP $\neq$ #P there are problems in #P that are neither in FP nor #P-complete — in fact it shows that there exists an infinite complexity hierarchy of intermediate problems within #P.

Another interesting aspect of the Dyer and Greenhill dichotomy is the fact that the homomorphism counting problem is tractable only for a very restricted class of graphs H. (It turns out that the only tractable cases are those for which the number of homomorphisms can be computed almost trivially.) This inherent hardness of counting problems drives the study of relaxations of the problem formulation.

- Instead of asking for the exact number of homomorphisms we might be content with a close approximation of this number. This leads to the field of approximate counting, which will be introduced in Section 1.4.

- Another option is to make use of deeper knowledge about the problem instances. For example, one might only be interested to solve a counting problem on graph instances with relatively small maximum degree, bounded treewidth [33, 97], or without certain induced subgraphs [85]. This approach also leads to the field of parameterised complexity, which explores a relaxed notion of tractability called *fixed-parameter tractability* [55].

- Alternatively, if we only need to know whether there is an even or an odd number of homomorphisms, then we aim to determine the number of homomorphisms modulo 2. The branch of modular counting is introduced in Section 1.5.

In this thesis, apart from exact counting, we investigate homomorphism problems from the realms of approximate counting and modular counting. We will see that, indeed, exact counting is hard in all but a few restricted cases, also when adding surjectivity constraints. We will witness that, in comparison, the complexity landscapes of the homomorphism framework are more nuanced in the approximate and modular counting settings — coincidentally, this means that analysing and classifying these problem versions also becomes more challenging and involved.

1.3.1 Previous Results

We now formally state the previously mentioned classification by Dyer and Greenhill. We also include an observation [32, 97] that the same classification holds for #LHom(H).

Theorem 1.4 ([42, Theorem 1]). *Let H be a graph. If every connected component of H is a reflexive clique or an irreflexive complete bipartite graph, then the problems* #Hom(H) *and* #LHom(H) *are in* FP*. Otherwise,* #Hom(H) *and* #LHom(H) *are* #P*-complete.*

Borgs, Chayes, Lovász, Sós and Vesztergombi [11] survey the work on counting homomorphisms with a focus on extremal graph theory and weighted homomorphisms. They investigate the convergence of sequences of graphs with respect to homomorphism numbers (as well as other things). For a collection of algebraic results on counting homomorphisms we can draw for instance from the textbooks by Hell and Nešetřil [98] as well as Lovász [118]. Further fundamental results also go back to Lovász [116].

Succeeding the Dyer and Greenhill dichotomy result, different generalisations of #Hom(H) have been classified. In a seminal work, Bulatov [15, Theroem 2.22] gives a complete complexity dichotomy for the counting constraint satisfaction problem using techniques from universal algebra. Subsequently, Dyer and Richerby [45] give a simpler proof of this dichotomy by more elementary means. They also introduce a new criterion for the classification and show that this criterion is decidable.

Another fruitful line of work explores the complexity of counting weighted homomorphisms, i.e., the complexity of computing partition functions [87, 138]. Bulatov and Grohe [14] give a complexity dichotomy for counting homomorphisms with non-negative real weights. This result is then extended to arbitrary real weights by Goldberg, Grohe, Jerrum and Thurley [72], and further extended to complex weights by Cai, Chen and Lu [21]. Other variants consider, for example, bounded-degree input graphs [84], or directed target graphs [20, 40].

1.3.2 Our Results

As outlined in Section 1.3, the complexities of the surjective homomorphism decision problem and the compaction decision problem are still unresolved. Despite this fact, in Chapter 2 we prove a complete complexity dichotomy for counting surjective homomorphisms to a fixed graph H, and we also prove a complete classification for counting compactions to a fixed graph H. We slightly extend these classifications by also including the problems of counting surjective list homomorphisms and list compactions (#LSHom(H) and #LComp(H)). These problems will be formally defined at the beginning of Chapter 2.

The main contribution is the following classification, which shows that the complexity of counting compactions is different from the complexity of counting homomorphisms (as classified in Theorem 1.4).

Theorem 1.5. *Let H be a graph. If every connected component of H is an irreflexive star or a reflexive clique of size at most* 2 *then* #Comp(H) *and* #LComp(H) *are in* FP*. Otherwise,* #Comp(H) *and* #LComp(H) *are* #P*-complete.*

There is also evidence that surjective homomorphism problems are harder to solve than (unrestricted) homomorphism problems. As we have mentioned in Section 1.3, there are graphs H for which the decision problem Hom(H) is in P, but SHom(H) is NP-complete. We present further evidence in that direction by showing, in Section 2.3.3, that in a *uniform* setting (both G and H are part of the input) there is a class of problems for which counting all homomorphisms is in FP, whereas counting only the surjective homomorphisms is #P-complete. In contrast, we show (Theorem 1.6) that the problem of counting surjective homomorphisms to a fixed graph H has the same complexity characterisation as the problem of counting all homomorphisms to H.

Theorem 1.6. *Let H be a graph. If every connected component of H is a reflexive clique or an irreflexive complete bipartite graph, then* #SHom(H) *and* #LSHom(H) *are in* FP*. Otherwise,* #SHom(H) *and* #LSHom(H) *are* #P*-complete.*

We will also show the following relationships that follow from the given classifications.

Corollary 1.7. *Let H be a graph. Then*

$$\#\textsc{Hom}(H) \equiv \#\textsc{LHom}(H) \equiv \#\textsc{SHom}(H) \equiv \#\textsc{LSHom}(H) \equiv \#\textsc{Ret}(H) \leq \#\textsc{Comp}(H) \equiv \#\textsc{LComp}(H).$$

Furthermore, there is a graph H for which #Comp(H) *and* #LComp(H) *are* #P*-complete, but* #Hom(H)*,* #LHom(H)*,* #SHom(H)*,* #LSHom(H) *and* #Ret(H) *are in* FP*.*

Note that, for every graph H, #Hom(H) and #Ret(H) are interreducible (details are given in Section 2.4). Therefore, by Theorems 1.4 and 1.5, there are graphs H for which #Ret(H) is in FP and #Comp(H) is #P-complete — one example is a reflexive clique on 3 vertices. This shows that an analogue of the Winkler conjecture [6, Conjecture 2] does not hold in the counting setting, unless FP = #P.

1.3.3 Subsequent Results

Our dichotomy results from Section 1.3.2 have been succeeded by some interesting subsequent work [25, 27, 30].

In the proof of Theorem 1.5 (presented in Chapter 2) we express compaction counts as linear combinations of homomorphism counts. We then use polynomial interpolation to prove hardness. The perspective of linear combinations has frequently proved useful when analysing homomorphisms, e.g., in [10], [118] and [42]. Interpolation is also a standard tool in the field, as noted in [42, Section 2].

In a note, Dell [30] initially points out that a simpler and more elegant version of the interpolation can be used to prove a weaker version of Theorems 1.5 and 1.6. In these weaker versions, the graph in the input is allowed to have loops. This approach uses ideas from an impactful paper by Curticapean, Dell and Marx [29] from 2017. The same idea, written more generally, was also discovered by Chen [25].

Refining these results, Chen, Curticapean and Dell [27] show a classification for counting homomorphisms to so-called *quantum graphs* (linear combinations of homomorphism counts). Apart from a shorter proof of the Dyer and Greenhill dichotomy, this work offers a nice presentation of a generalisation of our Theorems 1.5 and 1.6. It also short-cuts some of our more technical proofs. It makes use of the tensor product of graphs and the bipartite double cover to resolve the difficulties caused by self-loops in the previous versions. It also gives a matching conditional lower bound from the perspective of fine-grained complexity theory.

1.4 Counting Homomorphisms Approximately

In a seminal paper, Dyer, Goldberg, Greenhill and Jerrum [37] propose to systematically study the complexity of approximate counting by means of approximation-preserving reductions. Rather than formulating approximate counting problem versions (and investigating their relative complexity with respect to polynomial-time Turing reductions), Dyer et al. introduce the concept of approximation-preserving reductions (AP-reductions). Intuitively, an *approximation-preserving reduction* (AP-reduction) from a problem A to a problem B is an algorithm that is a good approximation algorithm for A if it has oracle access to a good approximation algorithm for B. In this case we write $A \leq_{\mathrm{AP}} B$.

In order to define AP-reductions formally, we restate some standard definitions taken from [123, Definitions 11.1, 11.2, Exercise 11.3]. A randomised algorithm gives an (ε, δ)-*approximation* for the value V if the output X of the algorithm satisfies $\Pr(|X - V| \leq \varepsilon V) \geq 1 - \delta$. Slightly overloading the notation, an (ε, δ)-*approximation* for a problem V is a randomised algorithm which, given an input x and parameters $\varepsilon,\ \delta \in (0, 1)$, outputs an (ε, δ)-approximation for $V(x)$. A *randomised approximation scheme* (RAS) for a problem V is an $(\varepsilon, 1/4)$-approximation of V. A RAS is called *fully polynomial* (FPRAS) if it runs in time that is polynomial in $1/\varepsilon$ and the size of the input $|x|$.

Now we can state the technical definition from [37]. An *approximation-preserving reduction* from a problem A to a problem B is a probabilistic oracle Turing machine M which takes as input an instance x of A and a parameter $\varepsilon \in (0, 1)$, and satisfies the following three properties:

1) Every oracle call made by M is of the form (y, δ), where y is an instance of B

and $\delta \in (0,1)$ with $1/\delta \in \text{poly}(|x|, 1/\varepsilon)$ specifies the precision of approximation.
2) The Turing machine M is a RAS for A whenever the oracle is a RAS for B.
3) The runtime of M is polynomial in $|x|$ and $1/\varepsilon$.

Problems that admit an FPRAS are considered to be efficiently approximable. #SAT is the problem of counting satisfying assignments of a Boolean formula, and Zuckerman [153] shows that there is no FPRAS for #SAT, unless NP = RP, which is considered to be extremely unlikely. Dyer et al. use the fact that #SAT is #P-complete under AP-reductions to show, more generally, that if #A is a counting version of an NP-complete underlying decision problem A, then #A is #P-complete under AP-reductions [37, Theorem 1]. (Recall that the exact counting analogue of this statement is supported only by empirical evidence.)

With an eye to Zuckerman's result, also consider a work by Bordewich [9], which states that if any problem in #P does not have an FPRAS then there is an infinite complexity hierarchy with respect to AP-reductions within #P. So there is an intricate complexity landscape of approximate counting problems and, once again, it is an interesting goal to find large problem classes that do not contain such an infinite approximation hierarchy.

In their groundwork, Dyer et al. [37] identify another important class of AP-interreducible problems. The class is sometimes named after its main representative #BIS, the problem of counting independent sets in a bipartite graph. Some examples of problems that are AP-interreducible with #BIS are counting downsets in a partial order, or counting configurations of the 2-particle Widom-Rowlinson model (which is the same as counting homomorphisms to a looped path on 3 vertices). At this point, #BIS is believed to be of intermediate complexity, i.e., it is not believed to have an FPRAS and there is also no known AP-reduction from #SAT to #BIS. Dyer et al. also show that #BIS is complete in $\#\text{RH}\Pi_1$, a logically defined subclass of #P. Many approximate counting problems have since been found to exhibit a complexity trichotomy using the three classes of 1) FPRAS-able problems, 2) problems AP-interreducible with #BIS, and 3) problems AP-interreducible with #SAT. Some examples appear in Section 1.4.1.

Approximate counting homomorphisms to a fixed graph H has been studied intensively in the past [37,38,62,63,76,78,108], and much progress has been made towards determining its complexity. However, despite a number of very interesting partial results, the complexity of approximately counting (unrestricted) homomorphisms is still one of the biggest (wide) open problems in counting complexity. To this day, there are still graphs with as few as four vertices for which its complexity is unresolved (see [108]). These examples are depicted in Figure 1.5. It is also not clear whether the class of problems of the form $\#\text{Hom}(H)$ contains an infinite approximation hierarchy. An intriguing candidate subclass that might exhibit such an infinite hierarchy is counting proper q-colourings of a bipartite graph, which for every integer q can be expressed by the $\#\text{Hom}(H)$ framework [37, Section 6]. Another possible candidate subclass, parameterised by an integer q, is approximating the partition function of the q-state ferromagnetic Potts model [130]. This problem is equivalent to approximately counting homomorphisms to a graph J_q that is the edge-subdivision of a q-leaf star [76].

This shows that there are major open problems, surprisingly, even if H is an irreflexive tree, or of very small size.

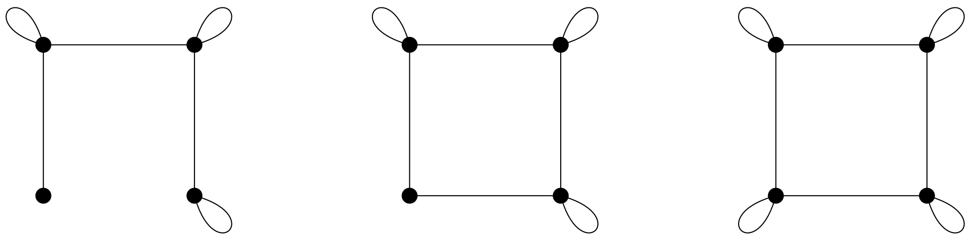


Figure 1.5: Graphs on at most 4 vertices for which the relative complexity of #Hom(H) with respect to AP-reductions is unresolved.

Another notable consequence of the proof of [37, Theorem 1] is that for every problem in #P one can design an FPRAS if given access to an NP oracle. In this sense, the complexity of approximate counting is closer to the complexity of decision problems than exact counting problems. We will also see this intuition reflected in the complexity landscape we present in Section 1.4.2.

1.4.1 Previous Results

A first classification for approximately counting homomorphisms is stated already in the initial work of Dyer et al. [37, Section 3]. It covers all connected graphs H with at most three vertices. In his thesis, Kelk [108] continues the work on approximately counting and sampling homomorphisms. Besides introducing many useful gadgets and tools, Kelk gives an almost complete complexity characterisation for connected graphs with at most four vertices, including more #BIS-equivalent problems.

Goldberg and Jerrum [76] investigate the complexity of approximately counting H-colourings for trees H. First they give a complete classification in form of a trichotomy for a weighted version of the problem. It turns out that counting *weighted* homomorphisms to trees H is in FP if H is a star, and it is AP-interreducible with #BIS if H is a caterpillar but not a star. (A *caterpillar* is an irreflexive tree that contains a path P such that all vertices outside of P have degree 1.) Otherwise, the problem is as hard to approximate as #SAT. A similar result holds for the unweighted problem #Hom(H). However, if H is an irreflexive tree other than a caterpillar then it is open whether approximately counting homomorphisms to H is #BIS-equivalent, #SAT-hard, or even none of the two. In this case, it is only known that #BIS AP-reduces to #Hom(H). However, there are trees to which approximately counting homomorphisms is #SAT-equivalent with respect to AP-reductions (see [76, Section 5]).

Theorem 1.8 ([76])**.** *Let H be an irreflexive tree.*

i) If H is a star, then #Hom(H) is in FP.

ii) *Otherwise, if* H *is a caterpillar, then* #Hom(H) *is* #BIS*-equivalent under AP-reductions.*
iii) *Otherwise,* #Hom(H) *is* #BIS*-hard under AP-reductions.*

Galanis, Goldberg and Jerrum [62] then show that #Hom(H) is #BIS-hard to approximate for almost every graph H. (The exceptions are only the tractable cases that are trivially solvable even for exact counting.)

Theorem 1.9 ([62])**.** *Let* H *be a connected graph. If* H *is a reflexive complete graph or an irreflexive complete bipartite graph, then* #Hom(H) *is in* FP*. Otherwise,* #BIS $\leq_{\mathrm{AP}}$ #Hom(H)*.*

Despite these results, the complexity of approximately counting (unrestricted) homomorphisms is still a major open problem. Intriguingly, lists make a crucial difference, and Galanis, Goldberg and Jerrum [63] give a complete complexity trichotomy for approximately counting list homomorphisms. For a graph H, the complexity of #LHom(H) is determined by the maximum complexity #LHom(C) for a connected component C of H (see [63]). In the connected case the complexity is determined by the following theorem.

Theorem 1.10 ([63])**.** *Let* H *be a connected graph.*

(i) *If* H *is a reflexive complete graph or an irreflexive complete bipartite graph, then* #LHom(H) *is in* FP*.*
(ii) *Otherwise, if* H *is an irreflexive bipartite permutation graph or a reflexive proper interval graph, then* #LHom(H) *is* #BIS*-equivalent under AP-reductions.*
(iii) *Otherwise,* #LHom(H) *is* #SAT*-equivalent under AP-reductions.*

The connection between approximately counting and approximately sampling homomorphisms to a fixed graph H is studied in a number of works (see, e.g., [38, 41, 78, 104, 105, 108]). Dyer, Goldberg and Jerrum [38] show that there are problems in #P that have an FPRAS but do not have a fully polynomial approximate sampler (FPAS), unless some plausible complexity theory assumptions are violated. They also show that within the framework of counting (weighted) H-colourings there is a reduction from approximate counting to approximate sampling. Another interesting result along these lines is the fact that approximate counting and approximate sampling are of equivalent complexity for so-called *self-reducible* problems — the technical definition of this property is not important at the moment. This result was presented by Jerrum, Valiant and Vazirani [105], and then reformulated and generalised by Dyer and Greenhill [41]. In Chapter 3, we will use the fact that the counting retraction problem is self-reducible to design the AP-reductions we present in Theorem 1.19.

On a more general note, there is a wide range of results on approximate counting problems that are broadly related to homomorphism counting, such as approximating partition functions (e.g. [4, 61, 64, 65, 73–75, 77, 128, 134]), approximate counting CSPs (e.g. [2, 17, 28, 39]), or the approximability of Holant problems (e.g. [3]).

1.4.2 Our Results

One aspect that motivates the study of approximately counting surjective homomorphisms, compactions and retractions is the hope of gaining further insights into the tantalising open questions about approximately counting (unrestricted) homomorphisms. Another aspect is the long history of research on these structures and the corresponding decision problems. We will show that in contrast to the exact counting setting (but similar to the decision setting) approximately counting retractions is the hardest of these homomorphism problems. Therefore, retractions are the main focus of Chapters 3 and 4. The main result is a classification for all *square-free* graphs (graphs without 4-cycles, whether induced or not), and it took two long papers [58, 59] to work our way from the class of trees to the class of all graphs without triangles (cycles of length 3) and squares (cycles of length 4), and finally to all square-free graphs. As we had hoped for, en route we also obtain new #BIS-easiness results for approximately counting homomorphisms.

Classification Results The *girth* of a graph is the length of a shortest cycle in H. The first contribution of Chapter 3 is a complexity trichotomy for approximately counting retractions to all graphs of girth at least 5. In order to state this result, we need a few definitions. A *partially bristled reflexive path* is a tree consisting of a reflexive path P, together with a (possibly empty) set of unlooped "bristle" vertices U and a matching connecting all of the vertices of U to "internal" vertices of P (vertices of P that are not endpoints of the path). A more formal definition, along with an example, is given in Section 3.1.2.

Theorem 1.11. *Let H be a graph of girth at least* 5.

i) If every connected component of H is an irreflexive star or a reflexive clique of size at most 2*, then* #$\textsc{Ret}(H)$ *is in* FP.
ii) Otherwise, if every connected component of H is an irreflexive caterpillar or a partially bristled reflexive path, then #$\textsc{Ret}(H)$ *is approximation-equivalent to* #BIS.
iii) Otherwise, #$\textsc{Ret}(H)$ *is approximation-equivalent to* #SAT.

The proof of Theorem 1.11 is presented in Section 3.2.3.

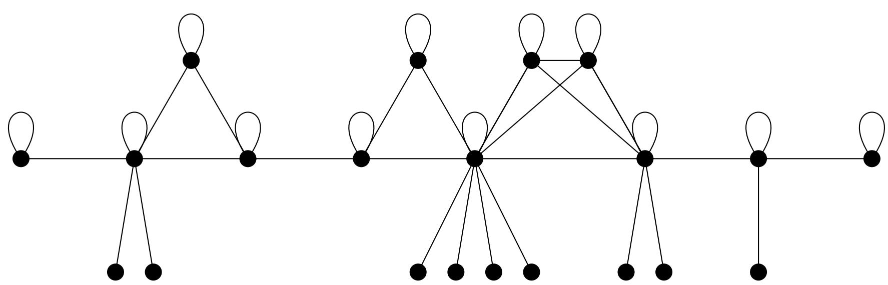


Figure 1.6: Example graph which turns out to be #BIS-easy.

The definition of partially bristled reflexive paths is refined and generalised by the class $\mathcal{H}_{\mathrm{BIS}}$ (which we formally define in Definition 4.2). Informally, a graph in $\mathcal{H}_{\mathrm{BIS}}$ is a path P of looped vertices with some attached bristles, depicted below the path in Figure 1.6 and some attached looped vertices forming cliques with two consecutive vertices of P (depicted above the path in Figure 1.6). For each vertex p of the path P the number of attached bristles satisfies the following properties:

- If p is an endpoint of P then it does not have a bristle.
- If p is not an endpoint of P then it is part of exactly two reflexive cliques K_L ("to the left" of p) and K_R ("to the right" of p). Then p has at most $(|K_L| - 1) \cdot (|K_R| - 1)$ bristles.

$\mathcal{H}_{\mathrm{BIS}}$ also contains graphs with squares and is more general than what we need to classify all square-free graphs. It is a fairly broad class of graphs but it turns out that approximately counting retractions to any member of $\mathcal{H}_{\mathrm{BIS}}$ is #BIS-easy (and #BIS-hardness follows from previous results).

Theorem 1.12. *Let H be a graph in $\mathcal{H}_{\mathrm{BIS}}$. Then approximately counting retractions to H is* #BIS*-equivalent under approximation-preserving reductions.*

For the class of square-free graphs, $\mathcal{H}_{\mathrm{BIS}}$ completely captures the truth — together with the class of non-trivial irreflexive caterpillars (defined momentarily) they form precisely the class of #BIS-equivalent square-free graphs. We prove Theorem 1.12 in Chapter 4, and then use it together with Theorem 1.11 and a number of other pieces to show the following more general classification for all square-free graphs. To shorten notation, we say that a graph is *trivial* if it is a reflexive clique or an irreflexive complete bipartite graph.

Theorem 1.13. *Let H be a square-free graph.*

i) If every connected component of H is trivial then approximately counting retractions to H is in FP.
ii) Otherwise, if every connected component of H is

- *trivial,*
- *in the class $\mathcal{H}_{\mathrm{BIS}}$, or*
- *is an irreflexive caterpillar*

then approximately counting retractions to H is #BIS*-equivalent.*
iii) Otherwise, approximately counting retractions to H is #SAT*-equivalent.*

Approximate Counting Landscape The second contribution of Chapter 3 is a map of the complexity landscape of approximate counting homomorphism problems. Before stating our results, we give an overview of the approximate counting complexity landscape in Figure 1.7.

The results summarised in this figure are consistent with the results that are known for the corresponding decision problems, as given in Proposition 1.2, and interestingly

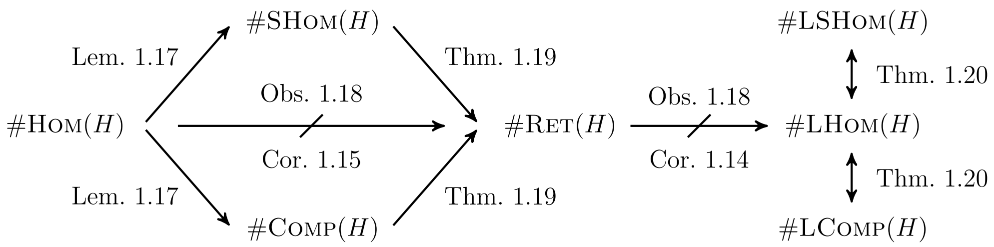


Figure 1.7: *Approximate counting complexity landscape.* An arrow from a problem A to a problem B means that there exists an AP-reduction from A to B. Struck through arrows correspond to a reductions with a separation. The references for the reduction and the separation are given above and below the arrow, respectively.

they are in contrast to the situation in the exact counting world. Note that for exact counting (see Corollary 1.7 and Figure 1.8), #Hom(H), #Ret(H) and #LHom(H) are interreducible. Also, all of the exact counting problems that we have mentioned reduce to #Comp(H) and #LComp(H). Moreover, #Comp(H) and #LComp(H) are separated from the remaining problems.

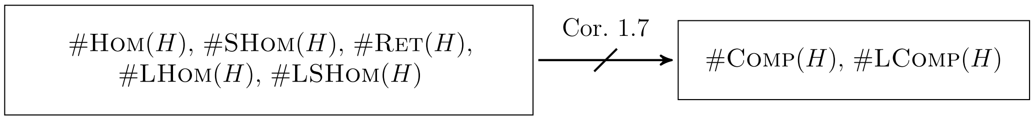


Figure 1.8: *Exact counting complexity landscape.* All problems in the same box are interreducible with respect to polynomial-time Turing reductions. The arrow means that each problem in the box on the left-hand side reduces to each problem on the right-hand side using a polynomial-time Turing reduction. The arrow is struck through as there exists a separation between each problem on the left and each problem on the right.

An interesting consequence of Theorem 1.11 is a separation between #Ret(H) and #LHom(H).

Corollary 1.14. #Ret(H) *and* #LHom(H) *are separated subject to the assumption that* #BIS *and* #SAT *are not AP-interreducible. In particular, if H is a partially bristled reflexive path with at least one unlooped vertex, then* #Ret(H) $\equiv_{\mathrm{AP}}$ #BIS*, whereas* #LHom(H) $\equiv_{\mathrm{AP}}$ #SAT*.*

The fact that #Ret(H) $\equiv_{\mathrm{AP}}$ #BIS for partially bristled reflexive paths follows from Theorem 1.11. The fact that #LHom(H) $\equiv_{\mathrm{AP}}$ #SAT is from [63], see Theorem 1.10. As another consequence, Theorem 1.11 separates #Ret(H) from #Hom(H), but in a different sense. Recall, that for $q \geq 3$, #Hom(J_q) is AP-interreducible with the task of computing the partition function of the q-state ferromagnetic Potts model [76] (where J_q is the irreflexive graph obtained from the q-leaf star by subdividing each edge). Despite extensive work on this problem [66, 74, 76], it is only known to be #BIS-hard but is not known to be #BIS-easy or to be #SAT-hard (with respect to AP-reductions).

Corollary 1.15. *Let q be an integer with $q \geq 3$. $\#\text{Hom}(H)$ and $\#\text{Ret}(H)$ are separated subject to the assumption that approximately computing the partition function of the q-state ferromagnetic Potts model is not #SAT-hard. In particular, it follows from Theorem 1.11 that $\#\text{SAT} \leq_{\text{AP}} \#\text{Ret}(J_q)$.*

In addition to these separations, we show that approximately counting retractions is at least as hard as approximately counting surjective homomorphisms and also at least as hard as approximately counting compactions (formally stated in Theorem 1.19). The latter is surprising as it is in contrast to known results for the corresponding exact counting problems (see Figure 1.8). Our proof uses an interesting Monte Carlo approach to AP-reductions and it appears to be the first time that Monte Carlo algorithms have been used in this context. More details on this method are given in Section 3.1.1. The approach gives analogous reductions for the list versions of these problems for free.

Since approximately counting homomorphisms reduces to approximately counting retractions, our #BIS-easiness results from Theorem 1.12 carry over to (unrestricted) homomorphism counting. Thus, we are able to resolve the complexity of approximately counting homomorphisms to graphs in $\mathcal{H}_{\text{BIS}}$, thereby adding to the very fragmented knowledge of #BIS-easiness results for approximately counting homomorphisms.

Corollary 1.16. *Let H be a graph in $\mathcal{H}_{\text{BIS}}$. Then approximately counting homomorphisms to H is #BIS-equivalent under approximation-preserving reductions.*

Some additional consequences of our reductions, when combined with previously known classification results, are mentioned at the end of this section.

Formal Reduction Statements This paragraph merely states the reductions from Figure 1.7 formally, so we can later refer back to them. We trace Figure 1.7 vaguely from left to right.

- We prove Lemma 1.17 in Section 3.4 in form of Lemmas 3.28 and 3.29.
  Lemma 1.17. *Let H be a graph. Then $\#\text{Hom}(H) \leq_{\text{AP}} \#\text{SHom}(H)$ and, if H is connected, then $\#\text{Hom}(H) \leq_{\text{AP}} \#\text{Comp}(H)$.*

- The following simple observation is immediate from the problem definitions.
  Observation 1.18. *Let H be a graph. Then $\#\text{Hom}(H) \leq_{\text{AP}} \#\text{Ret}(H) \leq_{\text{AP}} \#\text{LHom}(H)$.*

- Theorem 1.19 is proved in Section 3.3.1 in the form of Corollaries 3.24 and 3.26.
  Theorem 1.19. *Let H be a graph. Then $\#\text{SHom}(H) \leq_{\text{AP}} \#\text{Ret}(H)$ and $\#\text{Comp}(H) \leq_{\text{AP}} \#\text{Ret}(H)$.*

- The proof of Theorem 1.20 is in Section 3.4.
  Theorem 1.20. *Let H be a graph. Then $\#\text{LSHom}(H) \equiv_{\text{AP}} \#\text{LHom}(H)$ and $\#\text{LComp}(H) \equiv_{\text{AP}} \#\text{LHom}(H)$.*

Some additional Consequences We briefly mention some implications of our reductions when combined with previously known classification results. We recover an analogue of the Galanis, Goldberg and Jerrum dichotomy (Theorem 1.9) in terms of #BIS, cf. Section 3.4.

Theorem 1.21. *Let H be a connected graph. If H is a reflexive clique or an irreflexive biclique, then* #SHom(H)*,* #Ret(H) *and* #Comp(H) *are in* FP*. Otherwise, each of these problems is* #BIS*-hard under approximation-preserving reductions.*

This also allows us to state new #BIS-equivalence results, which are not limited to square-free graphs.

Corollary 1.22. *Let H be one of the following:*

- *A reflexive proper interval graph but not a complete graph.*
- *An irreflexive bipartite permutation graph but not a complete bipartite graph.*

Then #SHom(H)*,* #Comp(H) *and* #Ret(H) *are* #BIS*-equivalent.*

Remark 1.23. Since both approximately counting homomorphisms (Corollary 1.16) and approximately counting retractions (Theorem 1.13) are #BIS-equivalent for graphs in $\mathcal{H}_{\mathrm{BIS}}$ this result extends to the problems of approximately counting surjective homomorphisms and approximately counting compactions (by the reductions given in Lemma 1.17 and Theorem 1.19).

1.5 Counting Homomorphisms Modulo 2

In Chapter 5 we work on a conjecture by Faben and Jerrum [48] about the complexity of counting homomorphisms to a fixed graph H, modulo 2. In order to state this conjecture we need some definitions. Here is the formal definition of the *homomorphism parity problem*.

Name: $\oplus$Hom(H).
Input: An irreflexive graph G.
Output: The number of homomorphisms from G to H, modulo 2.

Analogously, we define $\oplus$Ret(H). These problems are members of the complexity class $\oplus$P, which contains all functions that compute the parity of a function in #P. Intuitively, it covers problems that ask to compute a number of solutions, modulo 2 (or alternatively, whether this number is even or odd).

An *involution* is an automorphism σ that is its own inverse, i.e., $\sigma \circ \sigma$ is the identity function. An involution is *non-trivial* if it is not the identity function, and a graph H is *involution-free* if it has no non-trivial involutions. For a graph H and an automorphism σ of H, H^σ is the subgraph of H induced by the fixed points of σ (the vertices v with $\sigma(v) = v$).

In order to study the parity of homomorphism numbers, Faben and Jerrum [48] introduced the concept of an *involution-free reduction*. They use two binary relations

$\to$ and $\to^*$ over the set of graphs. $H \to K$ if and only if $K = H^\sigma$ for some non-trivial involution σ, and $\to^*$ is the reflexive-transitive closure of $\to$. For each graph H, there is a uniquely defined involution-free graph H^* (up to isomorphism) with $H \to^* H^*$ [48, Theorem 3.7]. The graph H^* is the *involution-free reduction* of H.

Theorem 1.24 ([48, Theorem 3.4]). *For all graphs G and H, the number of homomorphisms from G to H has the same parity as the number of homomorphisms from G to H^*.*

Now we can state the Faben-Jerrum conjecture.

Conjecture 1.25 ([48, Section 3.1]). *Let H be a graph. If its involution-free reduction H^* has at most one vertex, then $\oplus$Hom(H) can be solved in polynomial time. Otherwise, $\oplus$Hom(H) is $\oplus$P-complete.*

Since $\oplus$Hom(H) and $\oplus$Hom(H^*) are interreducible (Theorem 1.24) it suffices to investigate involution-free graphs. An important insight by Göbel, Goldberg and Richerby [69, Theorem 3.1] shows that work on the parity homomorphism problem smoothly fits into the theme of this thesis. Their result implies that for involution-free graphs H, the problems $\oplus$Hom(H) and $\oplus$Ret(H) are interreducible. As a matter of fact, since the challenging part of the Faben-Jerrum conjecture is proving hardness results, we actually work primarily with the retraction problem and make ample use of the corresponding single-vertex lists.

Some early works on $\oplus$P are the result by Valiant [141] about computing the permanent modulo k, a result by Papadimitriou and Zachos [127] showing that $\oplus P^{\oplus P} = \oplus P$, and works by Goldschlager and Parberry [79] as well as Valiant and Vazirani [143]. Similarly to #P, Toda [139] shows that $\oplus$P is "above" the polynomial-time hierarchy. In particular, from every problem in PH there is a randomised polynomial-time reduction to some problem in $\oplus$P.

The class $\oplus$P has some peculiar properties. For one, $\oplus$P contains trivial parity problems with NP-complete underlying decision versions. As mentioned previously, we expect an NP-hard decision problem to come with a #P-hard exact counting version, and for approximate counting such connection is a known theorem (see [37, Theorem 1]). However, there are natural problems for which solutions appear in pairs. Consequently, counting solutions modulo 2 is trivial for such problems, despite the fact that determining whether there is any solution is NP-hard. One such example is counting proper 3-colourings of a graph. Since we can re-define the colours in six different ways, the overall number of 3-colourings is a multiple of 6, which makes the parity problem trivial. In contrast, the corresponding decision and exact counting problems are well-known to be NP-complete and #P-complete, respectively.

Another peculiarity about modular counting was discovered by Valiant [142] who gives an example problem for which counting solutions modulo 7 is polynomial-time solvable, but counting solutions modulo 2 is $\oplus$P-complete.

1.5.1 Previous Results

In their initial investigation, Faben and Jerrum [48, Theorems 3.8 and 6.1] confirm their conjecture (Conjecture 1.25) for all irreflexive forests. Göbel, Goldberg and

Richerby extend these results to broader classes of irreflexive graphs. They first consider connected graphs for which each edge belongs to at most one cycle — so-called *cactus graphs* [68]. They show the conjecture for all irreflexive graphs whose connected components are cactus graphs. Subsequently, they prove the conjecture for all graphs whose involution-free reduction is square-free [69].

Göbel, Lagodzinski and Seidel [70] as well as Kazeminia and Bulatov [107] study the more general setting of counting modulo p, where p is any prime number. They extend the classifications of the parity problem for trees and square-free graphs, respectively, to the setting of counting modulo p. More information on modular counting is presented in the theses of both Faben [47] and Göbel [71].

1.5.2 Our Results

In previous works, the presence of squares in the graph H caused significant difficulties for the hardness proofs (see [68, Section 5] and [69, Section 1.3]). In Chapter 5, we move away from this restriction and consider forbidden graph minors instead.

The concept of graph minors is covered by many text books (see, for example, [34]). In short, a graph H is J-minor-free if the graph J cannot be obtained from H by a sequence of vertex deletions, edge deletions, and edge contractions (removing any self-loops and parallel edges that are formed by the contraction). Graph classes based on excluded minors form the basis of the graph structure theory of Robertson and Seymour (see [117]).

The first (and main) contribution of Chapter 5 is to prove the Faben-Jerrum conjecture for every irreflexive graph H that does not have a K_4-minor, where K_4 is the complete graph on four vertices. The class of K_4-minor-free graphs is a rich and well-studied class. It is equivalent to the class of graphs with treewidth at most 2 and it includes all outerplanar and series-parallel graphs [36].

Both trees and cactus graphs are K_4-minor free, so our result subsumes the tree result of Faben and Jerrum [48] and also the cactus-graph result of Göbel et al. [68]. K_4-minor-free graphs can contain a 4-cycle and, going the other way, square-free graphs can have a K_4-minor. Thus, our result is incomparable with the result of [69]. However, as a minor contribution, our techniques also give a shorter proof of the classification for square-free graphs, see Remark 5.15.

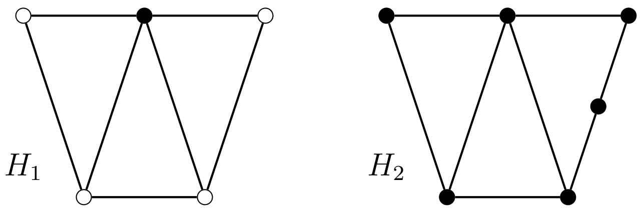
Our second contribution is to extend $\oplus$P-hardness, using the randomised version of the Exponential Time Hypothesis of Impagliazzo and Paturi (rETH) to rule out subexponential algorithms. In order to state our result, we first state the hypothesis.

Conjecture 1.26 (rETH, [101]). *There is a positive constant c such that no algorithm, deterministic or randomised, can decide the satisfiability of an n-variable* 3-SAT *instance in time* $\exp(c \cdot n)$.

Using the rETH, we can now state our main result.

Theorem 1.27. *Let H be a simple graph whose involution-free reduction H^* is K_4-minor free. If H^* contains at most one vertex, then $\oplus\textsc{Hom}(H)$ can be solved in polynomial time. Otherwise, $\oplus\textsc{Hom}(H)$ is $\oplus$P-complete and, assuming the randomised Exponential Time Hypothesis, it cannot be solved in time* $\exp(o(|V(G)| + |E(G)|))$.

As an example of an application of Theorem 1.27, consider the following K_4-minor-free graphs H_1 and H_2.



The graph H_1 has a non-trivial involution whose only fixed-point is the solid vertex, so H_1^* has one vertex. By Theorem 1.27, $\oplus$Hom(H_1) can be solved in polynomial time. The graph H_2 does not have any non-trivial involutions, so $H_2^* = H_2$. By Theorem 1.27, $\oplus$Hom(H_2) is $\oplus$P-complete and it cannot be solved in time $\exp(o(|V(G)| + |E(G)|))$, unless the rETH fails.

Both an overview of the techniques and the details of the proof of Theorem 1.27 are given in Chapter 5. Additionally, we pick up a couple of other classifications, namely

- a proof of the Faben-Jerrum conjecture for graphs whose involution-free reduction has degree at most 3 (Theorem 5.83), and
- a complete complexity classification for counting list homomorphisms, modulo 2 (Theorem 5.86). This answers a question from Göbel's PhD thesis [71, Chapter 9].

1.6 Organisation of this Thesis

This thesis is divided into three main parts. Part I contains our work on exactly counting homomorphisms under surjectivity constraints. In Part II, we proceed to investigate the relaxed model of approximate counting. The main focus of the chapters in Part II is the complexity of approximately counting retractions, relative to #BIS, relative to #SAT, and relative to other homomorphism counting problems. Finally, in Part III, we explore the complexity of determining the parity of homomorphism counts, that is, counting homomorphisms (and retractions), modulo 2.

Together with the basic definitions given in this introduction (Chapter 1) the different chapters are basically self-contained. We might sometimes point to results from another chapter, but we will only use definitions from the current chapter and the introduction. Consequently, each chapter has its own "preliminaries" section, which introduces notation used in that chapter. In particular, Parts I, II and III can be read and understood independently of each other.

For this reason, the list of symbols and notation, which is provided on page x, is arranged chapter-wise. Appendices use the notation and definitions from the chapter they belong to.

Part I

Exact Counting

Chapter 2

Exactly Counting Surjective Homomorphisms and Compactions

Well, some mathematics problems look simple, and you try them for a year or so, and then you try them for a hundred years, and it turns out that they're extremely hard to solve. There's no reason why these problems shouldn't be easy, and yet they turn out to be extremely intricate.

–Andrew Wiles, *Interview (2000)*

This chapter is based on the following paper:

[57] Jacob Focke, Leslie Ann Goldberg, and Stanislav Živný.
The complexity of counting surjective homomorphisms and compactions.
SIAM Journal on Discrete Mathematics, 33(2):1006–1043, 2019.

- A preliminary version of this work appeared in the Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, pp. 1772-1781.

Contents

Organisation of this Chapter

The structure of this chapter is straight-forward. In Section 2.2, we prove the dichotomy for exactly counting compactions, which is the main result of this chapter (Theorem 1.5). In Section 2.3, we first show the dichotomy for exactly counting surjective homomorphisms (Theorem 1.6). Afterwards, in Section 2.3.3, we investigate homomorphism counting in a uniform setting. Finally, in Section 2.4, we briefly address the classification for exactly counting retractions and establish the relationships between the different counting problems stated in Corollary 1.7.

2.1 Preliminaries

Graphs and Homomorphism Counts In this chapter, graphs are undirected and may contain loops but no parallel edges. For the scope of this chapter we use $N\big((G,\mathbf{S}) \to H\big)$ to denote the number of homomorphisms from $(G,\mathbf{S})$ to H. In the special case without lists, we use $N\big(G \to H\big)$ to denote the number of homomorphisms from a graph G to a graph H. We use $N^{\text{sur}}\big(G \to H\big)$ and $N^{\text{comp}}\big(G \to H\big)$ to denote the number of surjective homomorphisms and compactions, respectively.

In the introduction we have already defined the problems #Hom(H), #SHom(H), #Comp(H), #Ret(H) and #LHom(H). We will also use the following modifications of the list homomorphism counting problem.

Name: #LComp(H).
Input: Irreflexive graph G and a collection of lists $\mathbf{S} = \{S_v \subseteq V(H) : v \in V(G)\}$.
Output: $N^{\text{comp}}\big((G,\mathbf{S}) \to H\big)$.

Name: #LSHom(H).
Input: Irreflexive graph G and a collection of lists $\mathbf{S} = \{S_v \subseteq V(H) : v \in V(G)\}$.
Output: $N^{\text{sur}}\big((G,\mathbf{S}) \to H\big)$.

Some special classes of graphs appear frequently in this chapter. A graph H is a *biclique* if it is bipartite (disregarding any loops) and there is a partition of $V(H)$ into two disjoint sets U and V such that, for every $u \in U$ and $v \in V$, $E(H)$ contains the edge $\{u, v\}$. A biclique is a *star* if $|U| = 1$ or $|V| = 1$ (or both). Note that a star may have only one vertex since, for example, we could have $|U| = 1$ and $|V| = 0$. We sometimes use the notation $K_{a,b}$ to denote an irreflexive biclique whose vertices can be partitioned into U and V with $|U| = a$ and $|V| = b$. The *size* of a graph is the number of its vertices.

A subgraph H' of H is said to be *loop-hereditary* with respect to H if for every $v \in V(H')$ that is contained in a loop in $E(H)$, v is also contained in a loop in $E(H')$.

We indicate that two graphs G_1 and G_2 are isomorphic by writing $G_1 \cong G_2$.

Connected Graphs It will often be technically convenient to restrict the problems that we study by requiring the input graph G to be connected. In each case, we do this by adding a superscript "C" to the name of the problem. For example, the problem $\#\text{HOM}^{\text{C}}(\text{H})$ is defined as follows.

Name: $\#\text{HOM}^{\text{C}}(\text{H})$.
Input: A *connected* irreflexive graph G.
Output: $N\big(G \to H\big)$.

It is well known and easy to see (See, e.g., [118, (5.28)]) that if G is an irreflexive graph with components $G_1, \ldots, G_t$ then $N\big(G \to H\big) = \prod_{i \in [t]} N\big(G_i \to H\big)$. Similarly, given $\mathbf{S} = \{S_v \subseteq V(H) : v \in V(G)\}$ let $\mathbf{S}_i = \{S_v : v \in V(G_i)\}$. Then $N\big((G, \mathbf{S}) \to H\big) = \prod_{i \in [t]} N\big((G_i, \mathbf{S}_i) \to H\big)$. Thus, Dyer and Greenhill's theorem (Theorem 1.4) can be re-stated in the following convenient form.

Theorem 2.1 (Dyer, Greenhill)**.** *Let H be a graph. If every connected component of H is a reflexive clique or an irreflexive biclique, then* $\#\text{HOM}^{C}(\text{H})$*,* $\#\text{HOM}(H)$*,* $\#\text{LHOM}^{C}(\text{H})$ *and* $\#\text{LHOM}(H)$ *are all in* FP*. Otherwise,* $\#\text{HOM}^{C}(\text{H})$*,* $\#\text{HOM}(H)$*,* $\#\text{LHOM}^{C}(\text{H})$ *and* $\#\text{LHOM}(H)$ *are all* #P*-complete.*

Miscellanea Finally, we introduce some frequently used notation. For every positive integer n, we define $[n] = \{1, \ldots, n\}$.

Given sets S_1 and S_2, we write $S_1 \oplus S_2$ for the disjoint union of S_1 and S_2. Given graphs G_1 and G_2, we write $G_1 \oplus G_2$ for the graph $(V(G_1) \oplus V(G_2), E(G_1) \oplus E(G_2))$. If V is a set of vertices then we write $G_1 \oplus V$ as shorthand for the graph $G_1 \oplus (V, \emptyset)$. Similarly, if M is a matching (a set of disjoint edges) with vertex set V, then we write $G_1 \oplus M$ as shorthand for the graph $G_1 \oplus (V, M)$.

2.2 Counting Compactions

The section is divided into a short subsection on tractable cases and the main subsection on hardness results which also contains the proof of the final dichotomy result, Theorem 1.5.

2.2.1 Tractability Results

The tractability result in Lemma 2.2 follows from the fact (see Theorem 2.9) that the number of compactions from G to H can be expressed as a linear combination of the number of homomorphisms from G to certain subgraphs J of H. While we need the full details of our particular linear expansion to derive our hardness results, the following simpler version suffices for tractability.

Lemma 2.2. *Let H be a graph such that every connected component is an irreflexive star or a reflexive clique of size at most* 2*. Then* $\#\text{COMP}(H)$ *and* $\#\text{LCOMP}(H)$ *are in* FP*.*

Proof. First we deal with the case that H is the empty graph. Suppose that H is the empty graph and let $(G,\mathbf{S})$ be an instance of $\#\text{LCOMP}(H)$. If G is empty then $N^{\text{comp}}\big((G,\mathbf{S}) \to H\big) = 1$. Otherwise, $N^{\text{comp}}\big((G,\mathbf{S}) \to H\big) = 0$. Thus, if H is empty, then $\#\text{LCOMP}(H)$ is in FP. Obviously, this also implies that $\#\text{COMP}(H)$ is in FP.

Let $\mathcal{H}$ be the set of all non-empty graphs in which every connected component is an irreflexive star or a reflexive clique of size at most 2. We will show that for every $H \in \mathcal{H}$, $\#\text{LCOMP}(H)$ is in FP. To do this, we need the following notation. Given a graph H, let $m(H)$ denote the sum of $|V(H)|$ and the number of non-loop edges of H. We will use induction on $m(H)$.

The base case is $m(H) = 1$. In this case, H has only one vertex w. If G is non-empty and has $w \in S_v$ for every vertex $v \in V(G)$ then $N^{\text{comp}}\big((G,\mathbf{S}) \to H\big) = 1$. Otherwise, $N^{\text{comp}}\big((G,\mathbf{S}) \to H\big)$ is 0. So $\#\text{LCOMP}(H)$ is in FP.

For the inductive step, consider some $H \in \mathcal{H}$ with $m(H) > 1$. Let $(G,\mathbf{S})$ be an instance of $\#\text{LCOMP}(H)$. If G is empty then $N^{\text{comp}}\big((G,\mathbf{S}) \to H\big) = 0$, so suppose that G is non-empty. For every subgraph H' of H let $\mathbf{S}_{H'}$ denote the set of lists $\mathbf{S}_{H'} = \{S_v \cap V(H') : v \in V(G)\}$. It is easy to see that $N\big((G,\mathbf{S}) \to H\big) = \sum_{H'} N^{\text{comp}}\big((G,\mathbf{S}_{H'}) \to H'\big)$, where the sum is over all loop-hereditary subgraphs H' of H. This observation is well known and is implicit, e.g., in the proof of a lemma of Borgs, Chayes, Kahn and Lovász [10, Lemma 4.2] (in a context without lists or loops).

A subgraph H' of H is said to be a *proper* subgraph of H if either $V(H')$ is a strict subset of $V(H)$ or $E(H')$ is a strict subset of $E(H)$ (or both). For every graph H, let $\mathit{Sub}^{<}(H)$ denote the set of non-empty proper subgraphs of H that are loop-hereditary with respect to H. Note that if $H \in \mathcal{H}$ and $H' \in \mathit{Sub}^{<}(H)$ then $H' \in \mathcal{H}$ and $m(H') < m(H)$. This property holds explicitly for graphs in $\mathcal{H}$. In contrast, every reflexive clique and every irreflexive biclique outside of $\mathcal{H}$ contain some loop-hereditary subgraph that is neither clique nor biclique. We can refine the summation as follows.

$$N\big((G,\mathbf{S}) \to H\big) = N^{\text{comp}}\big((G,\mathbf{S}) \to H\big) + \sum_{H' \in \mathit{Sub}^{<}(H)} N^{\text{comp}}\big((G,\mathbf{S}_{H'}) \to H'\big).$$

Since $H \in \mathcal{H}$, every component of H is a reflexive clique or an irreflexive biclique, so Theorem 1.4 shows that the quantity $N\big((G, \mathbf{S}) \to H\big)$ on the left-hand side can be computed in polynomial time. By induction, we see that every term of the form $N^{\text{comp}}\big((G, \mathbf{S}_{H'}) \to H'\big)$ can also be computed in polynomial time. Subtracting this from the left-hand side, we obtain $N^{\text{comp}}\big((G, \mathbf{S}) \to H\big)$, as desired.

Thus, we have proved that #LCOMP(H) is in FP. The problem #COMP(H) is a restriction of #LCOMP(H), so it is also in FP.

□

2.2.2 Hardness Results

This is the key section of this chapter. In this section, we consider a graph H that has a connected component that is not an irreflexive star or a reflexive clique of size at most 2. The objective is to show that #COMP(H) and #LCOMP(H) are #P-hard (this is the hardness content of Theorem 1.5).

We start with a brief proof sketch. The easy case is when H contains a component that is not a reflexive clique or an irreflexive biclique. In this case, Dyer and Greenhill's Theorem 1.4 shows that #HOM(H) is #P-hard. We obtain the desired hardness by giving (in Theorem 2.5) a polynomial-time Turing reduction from #HOM(H) to #COMP(H). The result is finished off with a trivial reduction from #COMP(H) to #LCOMP(H). The proof of Theorem 2.5 is not difficult — given an input G to #HOM(H), we add isolated vertices and edges to G and recover the desired quantity $N\big(G \to H\big)$ using an oracle for #COMP(H) and polynomial interpolation. There are small technical issues related to size-1 components in H, and these are dealt with in Lemma 2.3.

The more interesting case is when every component of H is a reflexive clique or an irreflexive biclique, but some component is either a reflexive clique of size at least 3 or an irreflexive biclique that is not a star. The first milestone is Lemma 2.15, which shows #P-hardness in the special case where H is connected. We prove Lemma 2.15 in a slightly stronger setting where the input graph G is connected. This allows us, in the remainder of the section, to generalise the connected case to the case in which H is not connected.

The main difficulty, then, is Lemma 2.15. The goal is to show that #COMP(H) is #P-hard when H is a reflexive clique of size at least 3 or an irreflexive biclique that is not a star. Our main method for solving this problem is a technique (Theorem 2.9) that lets us compute the number of compactions from a connected graph G to a connected graph H using a weighted sum of homomorphism counts, say $N\big(G \to J_1\big), \ldots, N\big(G \to J_k\big)$. An important feature is that some of the weights might be negative.

Our basic approach will be to find a constituent J_i such that #HOM$^{\text{C}}(\text{J}_\text{i})$ is #P-hard and to reduce #HOM$^{\text{C}}(\text{J}_\text{i})$ to the problem of computing the weighted sum. Of course, if computing $N\big(G \to J_1\big)$ is #P-hard and computing $N\big(G \to J_2\big)$ is #P-hard, it does not follow that computing a weighted sum of these is #P-hard.

In order to solve this problem, in Lemmas 2.11 and 2.12 we use an argument similar to that of Lovász [116, Theorem 3.6] to prove the existence of input instances that help us to distinguish between the problems #HOM$^{\text{C}}(\text{J}_1), \ldots,$ #HOM$^{\text{C}}(\text{J}_\text{k})$. Theorem 2.13

then provides the desired reduction from a chosen #$\text{Hom}^{\text{C}}(\text{J}_\text{i})$ to the problem of computing the weighted sum. Theorem 2.13 is proved by a more complicated interpolation construction, in which we use the instances from Lemma 2.12 to modify the input.

Having sketched the proof at a high level, we are now ready to begin. We start by working towards the proof of Theorem 2.5. The first step is to show that deleting size-1 components from H does not add any complexity to #Comp(H).

Lemma 2.3. *Let H be a graph that has exactly q size-1 components. Let H' be the graph constructed from H by removing all size-1 components. Then #Comp(H') $\leq$ #Comp(H).*

Proof. Let $W = \{w_1, \dots, w_q\}$ be the vertices of H that are contained in size-1 components. We can assume $q \geq 1$, otherwise $H' = H$. Let G' be an input to #Comp(H') and note that G' might contain isolated vertices. For any non-negative integer t, let V_t be a set of t isolated vertices, distinct from the vertices of G', and let $G_t = G' \oplus V_t$. For all $i \in \{0, \dots, t\}$, we define $S^i(G')$ to be the number of homomorphisms σ from G' to H with the following properties:

1. σ uses all non-loop edges of H'.
2. $|\sigma(V(G')) \cap \{w_1, \dots, w_q\}| = i$,

where $\sigma(V(G'))$ is the image of $V(G')$ under the map σ. We define $N^i(V_t)$ as the number of homomorphisms τ from V_t to H such that $\{w_1, \dots, w_i\} \subseteq \tau(V(V_t))$. Intuitively, $N^i(V_t)$ is the number of homomorphisms from V_t to H that use at least a set of i arbitrary but fixed vertices of H, as the particular choice of vertices $\{w_1, \dots, w_i\}$ is not important when counting homomorphisms from a set of isolated vertices. For any compaction $\gamma \colon V(G_t) \to V(H)$, the restriction $\gamma|_{V(G)}$ has to use all non-loop edges in H'. As H' does not have size-1 components, this implies that all vertices other than $w_1, \dots, w_q$ are used by $\gamma|_{V(G)}$. Say, additionally, that γ uses $q - i$ vertices from W, for some $i \in \{0, \dots, q\}$. Then, $\gamma|_{V_t}$ has to use the remaining i vertices. Thus, for each fixed $t \geq 0$, we obtain a linear equation:

$$\underbrace{N^{\text{comp}}\big(G_t \to H\big)}_{b_t} = \sum_{i=0}^{q} \underbrace{S^{q-i}(G')}_{x_i} \underbrace{N^i(V_t)}_{a_{t,i}}.$$

By choosing $q + 1$ different values for the parameter t we obtain a system of linear equations. Here, we choose $t = 0, \dots, q$. Then the system is of the form $\mathbf{b} = \mathbf{A}\mathbf{x}$ for

$$\mathbf{b} = \begin{pmatrix} b_0 \\ \vdots \\ b_q \end{pmatrix}, \qquad \mathbf{A} = \begin{pmatrix} a_{0,0} & \dots & a_{0,q} \\ \vdots & \ddots & \vdots \\ a_{q,0} & \dots & a_{q,q} \end{pmatrix}, \qquad \text{and} \qquad \mathbf{x} = \begin{pmatrix} x_0 \\ \vdots \\ x_q \end{pmatrix}.$$

Note, that the vector $\mathbf{b}$ can be computed using $q+1$ #Comp(H) oracle calls. Further,

$$x_q = S^0(G') = N^{\text{comp}}\big(G' \to H'\big).$$

Thus, determining $\mathbf{x}$ is sufficient for computing the sought-for $N^{\text{comp}}(G' \to H')$. It remains to show that the matrix $\mathbf{A}$ is of full rank and is therefore invertible.

If $t < i$, we observe that $a_{t,i} = 0$ as we cannot use at least i vertices of H when we have fewer than i vertices in the domain. For the diagonal elements with $t \in \{0, \ldots, q\}$ we have that $a_{t,t} = N^t(V_t) = t!$ (note that $0! = 1$). Hence,

$$\mathbf{A} = \begin{pmatrix} 0! & 0 & \cdots & 0 \\ * & 1! & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ * & \cdots & * & q! \end{pmatrix}$$

is a triangular matrix with non-zero diagonal entries, which completes the proof. □

Lemma 2.4. *Let H be a graph without any size-1 components. Then $\#\text{Hom}(H) \leq \#\text{Comp}(H)$.*

Proof. The proof is by interpolation and is somewhat similar to the proof of Lemma 2.3. Let G be an input to $\#\text{Hom}(H)$. We design a graph $G_t = G \oplus I_t$ as an input to the problem $\#\text{Comp}(H)$ by adding a set I_t of t disjoint new edges to the graph G.

We introduce some notation. Let $E^0(H)$ be the set of non-loop edges of H and let $r = |E^0(H)|$. Let $S^k(G)$ be the number of homomorphisms σ from G to H that use exactly k of the non-loop edges of H (additionally, σ might use any number of loops). Let $\{e_1, \ldots, e_k\}$ be a set of k arbitrary but fixed non-loop edges from H. We define $N^k(I_t)$ as the number of homomorphisms τ from I_t to H such that $\{e_1, \ldots, e_k\}$ are amongst the edges used by τ. Note that the particular choice of edges $\{e_1, \ldots, e_k\}$ is not important when counting homomorphisms from an independent set of edges to H—$N^k(I_t)$ only depends on the numbers k and t.

We observe that, for each compaction $\gamma \colon V(G_t) \to V(H)$, the restriction $\gamma|_{V(G)}$ uses some set $F \subseteq E^0(H)$ of non-loop edges and does not use any other non-loop edges of H. Suppose that F has cardinality $|F| = r - k$ for some $k \in \{0, \ldots, r\}$. Then $\gamma|_{V(I_t)}$ uses at least the remaining k fixed non-loop edges of H. As H does not have any size-1 components, this ensures at the same time that γ is surjective.

Therefore, we obtain the following linear equation for a fixed $t \geq 0$:

$$\underbrace{N^{\text{comp}}(G_t \to H)}_{b_t} = \sum_{k=0}^{r} \underbrace{S^{r-k}(G)}_{x_k} \underbrace{N^k(I_t)}_{a_{t,k}}.$$

As in the proof of Lemma 2.3, we choose $t = 0, \ldots, r$ to obtain a system of linear equations with

$$\mathbf{b} = \begin{pmatrix} b_0 \\ \vdots \\ b_r \end{pmatrix}, \qquad \mathbf{A} = \begin{pmatrix} a_{0,0} & \cdots & a_{0,r} \\ \vdots & \ddots & \vdots \\ a_{r,0} & \cdots & a_{r,r} \end{pmatrix}, \qquad \text{and} \qquad \mathbf{x} = \begin{pmatrix} x_0 \\ \vdots \\ x_r \end{pmatrix}.$$

We can compute $\mathbf{b}$ using a $\#\text{Comp}(H)$ oracle. Further,

$$\sum_{k=0}^{r} x_k = \sum_{k=0}^{r} S^{r-k}(G) = \sum_{k=0}^{r} S^k(G) = N(G \to H).$$

Thus, determining the vector $\mathbf{x}$ is sufficient for computing the sought-for number of homomorphisms $N\big(G \to H\big)$.

Finally, we show that $\mathbf{A}$ is invertible. If $t < k$, we observe that $a_{t,k} = N^k(I_t) = 0$, as clearly it is impossible to use more than t edges of H when there are only t edges in I_t. Further, for the diagonal elements it holds that for $t \in [r]$ we have $a_{t,t} = N^t(I_t) = 2^t t!$ as there are $t!$ possibilities for assigning the edges in I_t to the fixed set of t edges of H and there are 2^t vertex mappings for each such assignment of edges, also $N^0(I_0) = 1$. Hence,

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ * & 2^1 1! & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ * & \cdots & * & 2^r r! \end{pmatrix}$$

is a triangular matrix with non-zero diagonal entries and is therefore invertible. □

Theorem 2.5. *Let H be a graph. Then* $\#\textsc{Hom}(H) \leq \#\textsc{Comp}(H)$.

Proof. Let H' be the graph constructed from H by removing all size-1 components. By Lemma 2.3 we obtain $\#\textsc{Comp}(H') \leq \#\textsc{Comp}(H)$. Then Lemma 2.4 can be applied to the graph H' and thus we obtain $\#\textsc{Hom}(H') \leq \#\textsc{Comp}(H') \leq \#\textsc{Comp}(H)$. Finally, it follows from Theorem 1.4 that $\#\textsc{Hom}(H') \equiv \#\textsc{Hom}(H)$, which gives $\#\textsc{Hom}(H) \equiv \#\textsc{Hom}(H') \leq \#\textsc{Comp}(H') \leq \#\textsc{Comp}(H)$. □

Theorem 2.5 shows that hardness results from Theorem 1.4 will carry over from $\#\textsc{Hom}(H)$ to $\#\textsc{Comp}(H)$. We also know some cases where $\#\textsc{Comp}(H)$ is tractable from Lemma 2.2. The complexity of $\#\textsc{Comp}(H)$ is still unresolved if every component of H is a reflexive clique or an irreflexive biclique, but some reflexive clique has size greater than 2, or some irreflexive biclique is not a star. This is the case described at length at the beginning of the section. Recall that the first step is to specify a technique (Theorem 2.9) that lets us compute the number of compactions from a connected graph G to a connected graph H using a weighted sum of homomorphism counts, say $N\big(G \to J_1\big), \ldots, N\big(G \to J_k\big)$. Towards this end, we introduce some definitions which we will use repeatedly in the remainder of this section.

Definition 2.6. A *weighted graph set* is a tuple $(\mathcal{H}, \lambda)$, where $\mathcal{H}$ is a set of *non-empty, pairwise non-isomorphic, connected* graphs and λ is a function $\lambda\colon \mathcal{H} \to \mathbb{Z}$.

Definition 2.7. Let H be a connected graph. By *Sub*(H) we denote the set of non-empty, loop-hereditary, connected subgraphs of H. Let $\mathcal{S}_H$ be a set which contains exactly one representative of each isomorphism class of the graphs in *Sub*(H). Finally, for $H' \in \mathcal{S}_H$, we define $\mu_H(H')$ to be the number of graphs in *Sub*(H) that are isomorphic to H'.

Note that for a connected graph H, we have $\mu_H(H) = 1$.

Definition 2.8. For each non-empty connected graph H, we define a weight function λ_H which assigns an integer weight to each non-empty connected graph J.

- If J is not isomorphic to any graph in $\mathcal{S}_H$, then $\lambda_H(J) = 0$.
- If $J \cong H$, then $\lambda_H(J) = 1$.
- Finally, if J is isomorphic to some graph in $\mathcal{S}_H$ but $J \ncong H$, we define $\lambda_H(J)$ inductively as follows.

$$\lambda_H(J) = -\sum_{\substack{H' \in \mathcal{S}_H \\ \text{s.t. } H' \ncong H}} \mu_H(H') \lambda_{H'}(J).$$

Note that λ_H is well-defined as all graphs $H' \in \mathcal{S}_H$ with $H' \ncong H$ are smaller than H either in the sense of having fewer vertices or in the sense of having the same number of vertices but fewer edges.

The following theorem is the key to our approach for computing the number of compactions from a connected graph G to a connected graph H using a weighted sum of homomorphism counts. In Appendix A, we give an illustrative example where we verify the theorem for the case $H = K_{2,3}$ and we give the intuition behind the definitions. Here we go on to give the formal statement and proof.

Theorem 2.9. *Let H be a non-empty connected graph. Then for every non-empty, irreflexive and connected graph G we have*

$$N^{\text{comp}}\big(G \to H\big) = \sum_{J \in \mathcal{S}_H} \lambda_H(J) N\big(G \to J\big).$$

Proof. Let $H_1, H_2, \ldots$ be the set of non-empty connected graphs sorted by some fixed ordering that ensures that if H_i is isomorphic to a subgraph of H_j, then $i \leq j$. We verify the statement of the theorem by induction over the graph index with respect to this ordering. Let G be non-empty, irreflexive and connected.

For the base case, H_1 is K_1, which is the graph with one vertex and no edges. In this case, $\mathcal{S}_{H_1} = \{K_1\}$ and $\lambda_{K_1}(K_1) = 1$. Also

$$N^{\text{comp}}\big(G \to K_1\big) = N\big(G \to K_1\big).$$

So the theorem holds in this case.

Now assume that the statement holds for all graphs up to index i and consider the graph H_{i+1}. For ease of notation we set $H = H_{i+1}$. We use the fact that every homomorphism from a connected graph G to H_{i+1} is a compaction onto some non-empty, loop-hereditary and connected subgraph of H_{i+1} and vice versa. Thus, it holds that

$$\begin{aligned} N\big(G \to H\big) &= \sum_{H' \in \mathcal{S}_H} \mu_H(H') \cdot N^{\text{comp}}\big(G \to H'\big) \\ &= N^{\text{comp}}\big(G \to H\big) + \sum_{\substack{H' \in \mathcal{S}_H \\ \text{s.t. } H' \ncong H}} \mu_H(H') \cdot N^{\text{comp}}\big(G \to H'\big). \end{aligned}$$

Thus, we can rearrange and use the induction hypothesis to obtain

$$\begin{aligned} N^{\mathrm{comp}}\big(G \to H\big) &= N\big(G \to H\big) - \sum_{\substack{H' \in \mathcal{S}_H \\ \text{s.t. } H' \not\cong H}} \mu_H(H') \cdot N^{\mathrm{comp}}\big(G \to H'\big) \\ &= N\big(G \to H\big) - \sum_{\substack{H' \in \mathcal{S}_H \\ \text{s.t. } H' \not\cong H}} \mu_H(H') \cdot \sum_{J \in \mathcal{S}_{H'}} \lambda_{H'}(J) N\big(G \to J\big). \end{aligned}$$

Then we change the order of summation and use that $\lambda_{H'}(J) = 0$ if J is not isomorphic to any graph in $\mathcal{S}_{H'}$ to collect all coefficients that belong to a particular term $N\big(G \to J\big)$. We obtain

$$\begin{aligned} N^{\mathrm{comp}}\big(G \to H\big) &= N\big(G \to H\big) - \sum_{\substack{J \in \mathcal{S}_H \\ \text{s.t. } J \not\cong H}} \Big(\sum_{\substack{H' \in \mathcal{S}_H \\ \text{s.t. } H' \not\cong H}} \mu_H(H') \lambda_{H'}(J) \Big) N\big(G \to J\big) \\ &= \sum_{J \in \mathcal{S}_H} \lambda_H(J) N\big(G \to J\big). \end{aligned}$$

□

We remark that Theorem 2.9 can be generalised to graphs H and G with multiple connected components by looking at all subgraphs of H, rather than just at the connected ones. However, within this chapter, the version for connected graphs suffices.

Let $(\mathcal{H}, \lambda)$ be a weighted graph set. The following parameterised problem is not natural in its own right, but it helps us to analyse the complexity of #COMP$^{\mathrm{C}}$(H):

Name: #GRAPHSETHOM$^{\mathrm{C}}((\mathcal{H}, \lambda))$.
Input: An irreflexive, connected graph G.
Output: $Z_{\mathcal{H},\lambda}(G) = \begin{cases} 0 & \text{if } G \text{ is empty} \\ \sum_{J \in \mathcal{H}} \lambda(J) N\big(G \to J\big) & \text{otherwise.} \end{cases}$

Corollary 2.10. *Let H be a non-empty connected graph. Then*

$$\#\text{COMP}^{C}(\mathrm{H}) \equiv \#\text{GRAPHSETHOM}^{C}((\mathcal{S}_{\mathrm{H}}, \lambda_{\mathrm{H}})).$$

Proof. The corollary follows directly from Theorem 2.9. □

Corollary 2.10 gives us the desired connection between weighted graph sets and compactions. We will use this later in the proof of Lemma 2.15 to establish the #P-hardness of #COMP$^{\mathrm{C}}$(H) when H is either a reflexive clique of size at least 3 or an irreflexive biclique that is not a star.

Our next goal is to prove Theorem 2.13, which states that, for certain weighted graph sets $(\mathcal{H}, \lambda)$, determining $Z_{\mathcal{H},\lambda}(G)$ is at least as hard as computing $N\big(G \to J\big)$

for some graph J from the set $\mathcal{H}$ with $\lambda(J) \neq 0$. To this end, we first introduce two lemmas that help us to distinguish between different graphs J in the interpolation that we will later use to prove Theorem 2.13.

For the following lemmas, we introduce some new notation. For a graph G with distinguished vertex $v \in V(G)$ and a graph H with distinguished vertex $w \in V(H)$, the quantity $N\big((G,v) \to (H,w)\big)$ denotes the number of homomorphisms h from G to H with $h(v) = w$. Analogously, $N^{\text{inj}}\big((G,v) \to (H,w)\big)$ denotes the number of injective homomorphisms h from G to H with $h(v) = w$. If there exists an isomorphism from G to H that maps v onto w, we write $(G,v) \cong (H,w)$, otherwise we write $(G,v) \ncong (H,w)$. In the following lemma, we show that for two such target entities (H_1, w_1) and (H_2, w_2) that are non-isomorphic, there exists an input which separates them. To this end, we use an argument very similar to that presented in [69, Lemma 3.6] and in the textbook by Hell and Nešetřil [98, Theorem 2.11], which goes back to the works of Lovász [116, Theorem 3.6].

Lemma 2.11. *Let H_1 and H_2 be connected graphs with distinguished vertices $w_1 \in V(H_1)$ and $w_2 \in V(H_2)$ such that $(H_1, w_1) \ncong (H_2, w_2)$. Suppose that one of the following cases holds:*

Case 1. H_1 and H_2 are reflexive graphs.

Case 2. H_1 and H_2 are irreflexive bipartite graphs, each of which contains at least one edge.

Then

i) There exists a connected irreflexive graph G with distinguished vertex $v \in V(G)$ for which $N\big((G,v) \to (H_1,w_1)\big) \neq N\big((G,v) \to (H_2,w_2)\big)$.

ii) In Case 2 we can assume that G contains at least one edge and is bipartite.

Proof. In order to shorten the proof, we define some notation that depends on which case holds. In Case 1, we say that a tuple (G,v) consisting of a graph G with distinguished vertex v is *relevant* if G is connected and reflexive. In Case 2, we say that it is relevant if G is connected, irreflexive and bipartite and contains at least one edge. We start with a claim that applies in either case.

Claim: There exists a relevant (G,v) such that

$$N\big((G,v) \to (H_1,w_1)\big) \neq N\big((G,v) \to (H_2,w_2)\big).$$

Proof of the claim: To prove the claim, assume for a contradiction that for all relevant (G,v) we have

$$N\big((G,v) \to (H_1,w_1)\big) = N\big((G,v) \to (H_2,w_2)\big). \tag{2.1}$$

The contradiction will follow from the following subclaim:

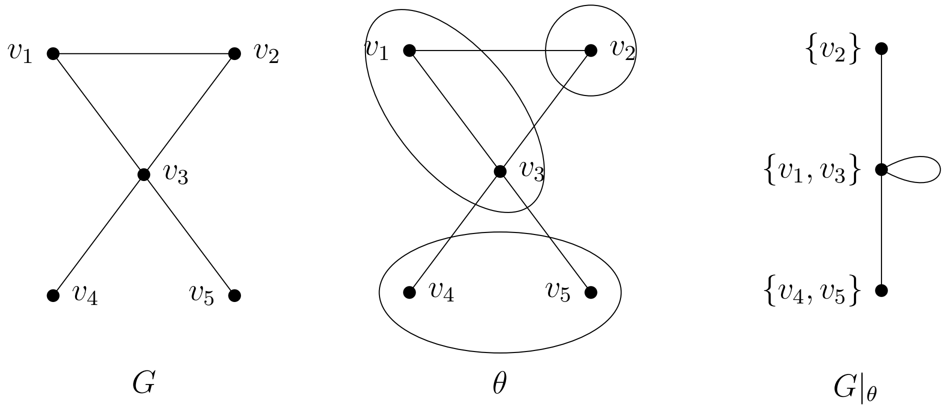


Figure 2.1: Graph G and the corresponding quotient graph $G|_\theta$ for $\theta = \{\{v_2\}, \{v_1, v_3\}, \{v_4, v_5\}\}$.

Subclaim: For every relevant (G, v),

$$N^{\text{inj}}\big((G, v) \to (H_1, w_1)\big) = N^{\text{inj}}\big((G, v) \to (H_2, w_2)\big).$$

Proof of the subclaim: The proof of the subclaim is by induction on the number of vertices of G. For the base case of the induction we treat the two cases separately.

In Case 1, the base case of the induction is $|V(G)| = 1$. The relevant (G, v) is the graph consisting of the single (looped) vertex v. For every reflexive graph H and vertex $w \in V(H)$ we have that $N\big((G, v) \to (H, w)\big) = N^{\text{inj}}\big((G, v) \to (H, w)\big)$. Therefore, (2.1) implies that the subclaim is true for this (G, v).

In Case 2, the base case of the induction is $|V(G)| = 2$. (There are no relevant (G, v) with $|V(G)| < 2$ since G has to contain an edge.) Consider a relevant (H, w). Since H is irreflexive and the two vertices of G are connected by an edge (so cannot be mapped by a homomorphism to the same vertex of H) we have $N\big((G, v) \to (H, w)\big) = N^{\text{inj}}\big((G, v) \to (H, w)\big)$. Once again, (2.1) implies that the subclaim is true for this (G, v).

For the inductive step, suppose that the subclaim holds for all relevant (G, v) in which G has up to $k - 1$ vertices. Consider a relevant (G, v) with $|V(G)| = k$. Let Θ be the set of partitions of $V(G)$ — that is, each $\theta \in \Theta$ is a set $\{U_1, \ldots, U_j\}$ for some integer j such that the elements of θ are non-empty and pairwise disjoint subsets of $V(G)$ with $\bigcup_{i=1}^{j} U_i = V(G)$. For $\theta \in \Theta$ with $\theta = \{U_1, \ldots, U_j\}$, by $G|_\theta$ we denote the corresponding *quotient graph*, i.e. let $G|_\theta$ be the graph with vertices $\{U_1, \ldots, U_j\}$ that has an edge $\{U_i, U_{i'}\}$ if and only if there exist $v \in U_i$ and $u \in U_{i'}$ with $\{v, u\} \in E(G)$. Therefore, $G|_\theta$ might have loops but no multi-edges, see Figure 2.1. Let v_θ denote the vertex of $G|_\theta$ which corresponds to the equivalence class of θ that contains the distinguished vertex v. Finally, let τ denote the partition of $V(G)$ into singletons.

Then for every relevant (H, w) it holds that

$$\begin{aligned} N\big((G,v) \to (H,w)\big) &= \sum_{\theta\in\Theta} N^{\mathrm{inj}}\big((G|_\theta, v_\theta) \to (H,w)\big) \\ &= N^{\mathrm{inj}}\big((G|_\tau, v_\tau) \to (H,w)\big) + \sum_{\theta\in\Theta\setminus\{\tau\}} N^{\mathrm{inj}}\big((G|_\theta, v_\theta) \to (H,w)\big) \\ &= N^{\mathrm{inj}}\big((G, v) \to (H,w)\big) + \sum_{\theta\in\Theta\setminus\{\tau\}} N^{\mathrm{inj}}\big((G|_\theta, v_\theta) \to (H,w)\big), \end{aligned} \tag{2.2}$$

where the third equality follows as $G|_\tau = G$.

Now we show that only relevant tuples $(G|_\theta, v_\theta)$ actually contribute to the sum in (2.2). First, note that since G is connected, so is $G|_\theta$.

In Case 1, every quotient graph $G|_\theta$ is reflexive. Therefore, for every $\theta \in \Theta \setminus \{\tau\}$, the tuple $(G|_\theta, v_\theta)$ is relevant.

In Case 2, H is an irreflexive bipartite graph with at least one edge. Therefore, we have $N^{\mathrm{inj}}\big((G|_\theta, v_\theta) \to (H,w)\big) > 0$ only if $G|_\theta$ is an irreflexive bipartite graph and also, θ is a proper vertex-colouring of G, i.e. every part of θ is an independent set. For such a partition θ, $G|_\theta$ has at least one edge if G does. We have now shown that only relevant tuples $(G|_\theta, v_\theta)$ contribute to the sum in (2.2).

Therefore, let Γ be the set of all partitions θ of $V(G)$ such that $(G|_\theta, v_\theta)$ is relevant. Then, we can rephrase (2.2) as follows.

$$N\big((G,v) \to (H,w)\big) = N^{\mathrm{inj}}\big((G, v) \to (H,w)\big) + \sum_{\theta\in\Gamma\setminus\{\tau\}} N^{\mathrm{inj}}\big((G|_\theta, v_\theta) \to (H,w)\big). \tag{2.3}$$

To prove the subclaim, we can set (H, w) in (2.3) to be (H_1, w_1). Similarly, we can set it to be (H_2, w_2). Then, we can use the induction hypothesis, the subclaim, on all tuples $(G|_\theta, v_\theta)$ in the sum as all these tuples are relevant and the partitions $\theta \in \Gamma \setminus \{\tau\}$ have strictly fewer than k parts. Applying (2.1), we obtain

$$N^{\mathrm{inj}}\big((G,v) \to (H_1,w_1)\big) = N^{\mathrm{inj}}\big((G, v) \to (H_2,w_2)\big),$$

which completes the induction and the proof of the subclaim. **(End of the proof of the subclaim.)**

We show next how to use the subclaim to derive a contradiction. In particular, in the subclaim we can set (G, v) to be either (H_1, w_1) or (H_2, w_2). This implies $(H_1, w_1) \cong (H_2, w_2)$, which gives the desired contradiction. Thus, we have shown contrary to (2.1) that there exists a relevant (G, v) with

$$N\big((G,v) \to (H_1,w_1)\big) \neq N\big((G, v) \to (H_2,w_2)\big)$$

and therefore we have proved the claim. **(End of the proof of the claim.)**

In Case 2, the claim is identical to the statement of the lemma. However, in Case 1 a relevant tuple (G, v) contains a reflexive graph G, whereas for the statement of the lemma, G has to be irreflexive. This is easily fixed as we can set G^0 to be the graph constructed from G by removing all loops. Using the fact that H_1 and H_2 are reflexive, we obtain for $i = 1$ and $i = 2$ that

$$N\big((G^0, v) \to (H_i, w_i)\big) = N\big((G, v) \to (H_i, w_i)\big).$$

Hence, the choice (G^0, v) has all the desired properties. □

In the following lemma, we generalise the pairwise property from Lemma 2.11. The result and the proof are adapted versions of [62, Lemma 6]. For ease of notation let $\binom{[k]}{2}$ denote the set of all pairs $\{i, j\}$ with $i, j \in [k]$ and $i \neq j$.

Lemma 2.12. *Let $H_1, \dots, H_k$ be connected graphs with distinguished vertices $w_1, \dots, w_k$ where $w_i \in V(H_i)$ for all $i \in [k]$ and, for every pair $\{i, j\} \in \binom{[k]}{2}$, we have $(H_i, w_i) \ncong (H_j, w_j)$. Suppose that one of the following cases holds:*

Case 1. $\forall i \in [k]$, H_i is a reflexive graph.

Case 2. $\forall i \in [k]$, H_i is an irreflexive bipartite graph that contains at least one edge.

Then

i) There exists a connected irreflexive graph G with a distinguished vertex $v \in V(G)$ such that, for every $\{i, j\} \in \binom{[k]}{2}$, it holds that $N\big((G, v) \to (H_i, w_i)\big) \neq N\big((G, v) \to (H_j, w_j)\big)$.

ii) In Case 2 we can assume that G contains at least one edge and is bipartite.

Proof. Again, we use the notion of relevant tuples but slightly modify the definition from the one given in the proof of Lemma 2.11. A tuple (G, v) is called relevant if G is a connected *irreflexive* graph and, in Case 2, if additionally G contains at least one edge and is bipartite. We show that there exists a relevant (G, v) such that for every $\{i, j\} \in \binom{[k]}{2}$ we have

$$N\big((G, v) \to (H_i, w_i)\big) \neq N\big((G, v) \to (H_j, w_j)\big).$$

We use induction on k, which is the number of graphs $H_1, \dots, H_k$. The base case for $k = 2$ is covered by Lemma 2.11. Now let us assume that the statement holds for $k - 1$ and the inductive step is for k. By the inductive hypothesis there exists a relevant (G, v) such that without loss of generality (possibly by renaming the graphs $H_1, \dots, H_k$)

$$N\big((G, v) \to (H_2, w_2)\big) > \dots > N\big((G, v) \to (H_k, w_k)\big).$$

Let $i^* \in [k] \setminus \{1\}$ be an index with

$$N\big((G, v) \to (H_1, w_1)\big) = N\big((G, v) \to (H_{i^*}, w_{i^*})\big).$$

If no such index exists, we can simply choose the graph G which then fulfils the statement of the lemma. Using the base case, there exists a relevant (G', v') such that

$$N\big((G', v') \to (H_1, w_1)\big) > N\big((G', v') \to (H_{i^*}, w_{i^*})\big),$$

possibly renaming (H_1, w_1) and (H_{i^*}, w_{i^*}). Let $i \in [k]$.

First, we show that for all $i \in [k]$ we have $N\big((G', v') \to (H_i, w_i)\big) \geq 1$. This is clearly true for Case 1, where w_i has a loop. In this case, we can always map all vertices of G' to the single vertex w_i.

In Case 2, as H_i is connected and contains at least one edge, there is some $w \in V(H_i)$ such that $\{w, w_i\} \in E(H_i)$. Since (G', v') is relevant, G' is connected and bipartite and has at least one edge. Let $\{A, B\}$ be a partition of $V(G')$ such that $v' \in A$ and A and B are independent sets of G. There is a homomorphism h from G' to H_i with $h(v') = w_i$ which maps all vertices in A to w_i and all vertices in B to w.

Therefore, in both cases we have shown that for all $i \in [k]$ we have

$$N\big((G', v') \to (H_i, w_i)\big) \geq 1.$$

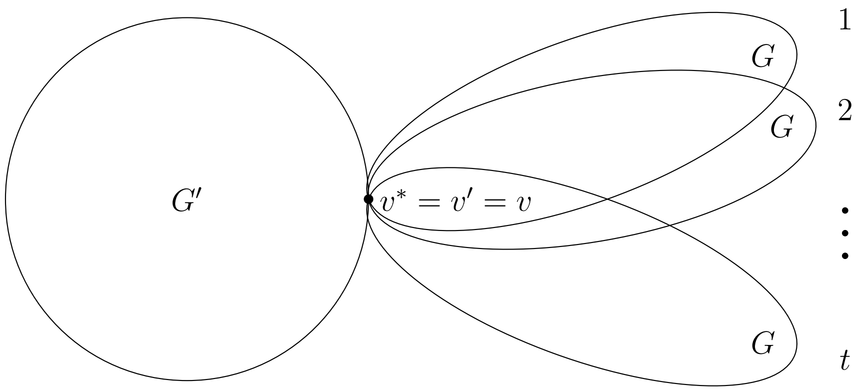


Figure 2.2: (G^*, v^*).

For a yet to be determined number t we construct a graph G^* from (G, v) and (G', v') by taking the graph G' and t copies of G and identifying the vertex v' with the t copies of v and call the resulting vertex v^*, cf. Figure 2.2. Note that from the fact that (G, v) and (G', v') are relevant, it is straightforward to show that (G^*, v^*) is relevant as well. Then, for any graph H and $w \in V(H)$ it holds that

$$N\big((G^*, v^*) \to (H, w)\big) = N\big((G', v') \to (H, w)\big) \cdot N\big((G, v) \to (H, w)\big)^t.$$

The goal is to choose t sufficiently large to achieve

$$\begin{aligned} N\big((G^*, v^*) \to (H_2, w_2)\big) > \ldots &> N\big((G^*, v^*) \to (H_{i^*-1}, w_{i^*-1})\big) \\ &> N\big((G^*, v^*) \to (H_1, w_1)\big) \\ &> N\big((G^*, v^*) \to (H_{i^*}, w_{i^*})\big) \\ &> \ldots \\ &> N\big((G^*, v^*) \to (H_k, w_k)\big). \end{aligned}$$

Accordingly, we define a permutation σ of the indices $\{1, \ldots, k\}$ that inserts index 1 between position $i^* - 1$ and i^*. The domain of σ corresponds to the new indices to which we assign the former indices. To avoid confusion, we give the function table in Table 2.1

Table 2.1: Function table of σ.

i	1	$\cdots$	i^*-2	i^*-1	i^*	$\cdots$	k
$\sigma(i)$	2	$\cdots$	i^*-1	1	i^*	$\cdots$	k

Formally,

$$\sigma(i) = \begin{cases} i+1 & \text{if } i \leq i^* - 2 \\ 1 & \text{if } i = i^* - 1 \\ i & \text{otherwise.} \end{cases}$$

Let $M = N\big((G, v) \to (H_2, w_2)\big)$. As $N\big((G', v') \to (H_j, w_j)\big) \geq 1$ for all $j \in [k]$, it is well-defined to set

$$C = \max_{j \in [k] \setminus \{i^*-1\}} \frac{N\big((G', v') \to (H_{\sigma(j+1)}, w_{\sigma(j+1)})\big)}{N\big((G', v') \to (H_{\sigma(j)}, w_{\sigma(j)})\big)}$$

and $t = \lceil CM \rceil$. Let G^* be as defined above. For ease of notation, for $j \in [k-1]$, we set

$$\xi(j) = \frac{N\big((G^*, v^*) \to (H_{\sigma(j)}, w_{\sigma(j)}\big)}{N\big((G^*, v^*) \to (H_{\sigma(j+1)}, w_{\sigma(j+1)}\big)}.$$

We want to show $\xi(j) > 1$ for all $j \in [k-1]$ to complete the proof.

For $j = i^* - 1$ we obtain

$$\begin{aligned} \xi(j) &= \frac{N\big((G^*, v^*) \to (H_{\sigma(i^*-1)}, w_{\sigma(i^*-1)})\big)}{N\big((G^*, v^*) \to (H_{\sigma(i^*)}, w_{\sigma(i^*)})\big)} \\ &= \frac{N\big((G^*, v^*) \to (H_1, w_1)\big)}{N\big((G^*, v^*) \to (H_{i^*}, w_{i^*})\big)} \\ &= \frac{N\big((G', v') \to (H_1, w_1)\big)}{N\big((G', v') \to (H_{i^*}, w_{i^*})\big)} > 1. \end{aligned}$$

For $j \in [k-1] \setminus \{i^* - 1\}$ we have

$$\begin{aligned} \xi(j) &= \frac{N\big((G^*, v^*) \to (H_{\sigma(j)}, w_{\sigma(j)})\big)}{N\big((G^*, v^*) \to (H_{\sigma(j+1)}, w_{\sigma(j+1)})\big)} \\ &= \frac{N\big((G', v') \to (H_{\sigma(j)}, w_{\sigma(j)})\big) \cdot N\big((G, v) \to (H_{\sigma(j)}, w_{\sigma(j)})\big)^t}{N\big((G', v') \to (H_{\sigma(j+1)}, w_{\sigma(j+1)})\big) \cdot N\big((G, v) \to (H_{\sigma(j+1)}, w_{\sigma(j+1)})\big)^t} \\ &\geq \frac{1}{C} \left(\frac{N\big((G, v) \to (H_{\sigma(j)}, w_{\sigma(j)})\big)}{N\big((G, v) \to (H_{\sigma(j+1)}, w_{\sigma(j+1)})\big)} \right)^t. \end{aligned}$$

Since $N\big((G,v) \to (H_{\sigma(j)}, w_{\sigma(j)})\big) \geq 1 + N\big((G,v) \to (H_{\sigma(j+1)}, w_{\sigma(j+1)})\big)$ for

$$j \in [k-1] \setminus \{i^* - 1\}$$

we have

$$\xi(j) \geq \frac{1}{C}\left(1 + \frac{1}{N\big((G,v) \to (H_{\sigma(j+1)}, w_{\sigma(j+1)})\big)}\right)^t.$$

Using $(1+x)^t \geq 1 + tx > tx$ for $t \geq 1$, $x \geq 0$ we obtain

$$\xi(j) > \frac{t}{C \cdot N\big((G,v) \to (H_{\sigma(j+1)}, w_{\sigma(j+1)})\big)}.$$

Finally, we use that for all $j \in [k-1] \setminus \{i^* - 1\}$ we have

$$N\big((G,v) \to (H_2, w_2)\big) > N\big((G,v) \to (H_{\sigma(j+1)}, w_{\sigma(j+1)})\big)$$

and conclude

$$\xi(j) > \frac{t}{C \cdot N\big((G,v) \to (H_2, w_2)\big)} \geq \frac{t}{CM} \geq 1.$$

Thus, we have shown $\xi(j) > 1$ as required, which completes the proof. □

In the following theorem, we use the separating instances that we obtain from Lemma 2.12 for interpolation-based reductions to $\#\textsc{GraphSetHom}^{\mathrm{C}}((\mathcal{H}, \lambda))$.

Theorem 2.13. *Let $(\mathcal{H}, \lambda)$ be a weighted graph set for which one of two cases holds:*

Case 1. All graphs in $\mathcal{H}$ are reflexive.

Case 2. All graphs in $\mathcal{H}$ are irreflexive and bipartite.

Then, for all $H \in \mathcal{H}$ with $\lambda(H) \neq 0$, $\#\textsc{Hom}^{C}(\mathrm{H}) \leq \#\textsc{GraphSetHom}^{C}((\mathcal{H}, \lambda))$.

Proof. If, in Case 2, $\mathcal{H}$ contains a graph without edges, i.e. a single-vertex graph K_1, let $(\mathcal{H}', \lambda')$ be a weighted graph set constructed from $(\mathcal{H}, \lambda)$ by removing the K_1 and its corresponding weight $\lambda(K_1)$. As $\#\textsc{Hom}(K_1)$ is in FP we have

$$\#\textsc{GraphSetHom}^{\mathrm{C}}((\mathcal{H}', \lambda')) \leq \#\textsc{GraphSetHom}^{\mathrm{C}}((\mathcal{H}, \lambda))$$

and

$$\#\textsc{Hom}^{\mathrm{C}}(\mathrm{K}_1) \leq \#\textsc{GraphSetHom}^{\mathrm{C}}((\mathcal{H}, \lambda)).$$

Therefore, for the remainder of this proof, we assume that every graph in $\mathcal{H}$ contains at least one edge. Let $\mathcal{H}^{\neq 0} = \{H_1, \ldots, H_k\}$ be the set of graphs in $\mathcal{H}$ that are assigned non-zero weights by λ. Note that all graphs in $\mathcal{H}^{\neq 0}$ are pairwise non-isomorphic, connected and non-empty by definition of a weighted graph set. Thus, for every pair $\{i,j\} \in \binom{[k]}{2}$ and every $w_i \in V(H_i)$, $w_j \in V(H_j)$ we have $(H_i, w_i) \not\cong (H_j, w_j)$.

Now, for each graph H_i we collect the vertices which are in the same orbit of the automorphism group of H_i. Formally, for each $i \in [k]$ and $w \in V(H_i)$, let $[w]$ be the orbit of w, i.e. the set of vertices w' such that $(H_i, w') \cong (H_i, w)$. Let W be a set which contains exactly one representative from each such orbit. Further, for each $i \in [k]$ set $W_i = W \cap V(H_i)$. Then, for each $w, w' \in W_i$ with $w' \neq w$, we have $(H_i, w) \ncong (H_i, w')$.

Let $k' = \sum_{i=1}^{k} |W_i|$ and let $(H'_1, w'_1), \ldots, (H'_{k'}, w'_{k'})$ be an enumeration of the tuples $\{(H_i, w_i) : i \in [k],\ w_i \in W_i\}$. Then we can apply Lemma 2.12 to the input $(H'_1, w'_1), \ldots, (H'_{k'}, w'_{k'})$ to obtain a connected irreflexive graph J with distinguished $u \in V(J)$ such that for every $i, j \in [k]$ and for all $w_i \in W_i$, $w_j \in W_j$ we have $N\big((J, u) \to (H_i, w_i)\big) \neq N\big((J, u) \to (H_j, w_j)\big)$.

Let $i \in [k]$ and suppose that $H_i \in \mathcal{H}$ and $\lambda(H_i) \neq 0$. Let G be a non-empty graph which is an input to the problem #HOM$^{\mathrm{C}}(\mathrm{H_i})$. Let v be an arbitrary vertex of G. We use the same construction as in Figure 2.2 to design a graph G_t as input to the problem #GRAPHSETHOM$^{\mathrm{C}}((\mathcal{H}, \lambda))$ by taking t copies of J as well as the graph G and identifying the t copies of vertex u with the vertex $v \in V(G)$. As both G and J are connected, G_t is as well. Then, using an oracle for #GRAPHSETHOM$^{\mathrm{C}}((\mathcal{H}, \lambda))$, we can compute $Z_{\mathcal{H},\lambda}(G_t)$ with

$$\begin{aligned} Z_{\mathcal{H},\lambda}(G_t) &= \sum_{H \in \mathcal{H}} \lambda(H) N\big(G_t \to H\big) \\ &= \sum_{i \in [k]} \lambda(H_i) N\big(G_t \to H_i\big) \\ &= \sum_{i \in [k]} \lambda(H_i) \sum_{w \in V(H_i)} N\big((G, v) \to (H_i, w)\big) \cdot N\big((J, u) \to (H_i, w)\big)^t \qquad (2.4) \end{aligned}$$

Now we collect the terms which belong to vertices in the same orbit. To this end, for $w \in W$ and $i \in [k]$ such that $w \in V(H_i)$, we define $\lambda_w = |[w]| \cdot \lambda(H_i)$, $N_w(G) = N\big((G, v) \to (H_i, w)\big)$ and $N_w(J) = N\big((J, u) \to (H_i, w)\big)$. Let $W = \{w_0, \ldots, w_r\}$. Then, continuing from Equation (2.4):

$$\begin{aligned} Z_{\mathcal{H},\lambda}(G_t) &= \sum_{i \in [k]} \lambda(H_i) \sum_{w \in V(H_i)} N\big((G, v) \to (H_i, w)\big) \cdot N\big((J, u) \to (H_i, w)\big)^t \\ &= \sum_{w \in W} \lambda_w N_w(G) N_w(J)^t. \end{aligned}$$

By choosing $r + 1$ different values for the parameter t — here it is sufficient to choose $t = 0, \ldots, r$ — we obtain a system of linear equations $\mathbf{b} = \mathbf{A}\mathbf{x}$ as follows:

$$\mathbf{b} = \begin{pmatrix} Z_{\mathcal{H},\lambda}(G_0) \\ \vdots \\ Z_{\mathcal{H},\lambda}(G_r) \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} \lambda_{w_0} N_{w_0}(J)^0 & \ldots & \lambda_{w_r} N_{w_r}(J)^0 \\ \vdots & \ddots & \vdots \\ \lambda_{w_0} N_{w_0}(J)^r & \ldots & \lambda_{w_r} N_{w_r}(J)^r \end{pmatrix}, \text{ and } \quad \mathbf{x} = \begin{pmatrix} N_{w_0}(G) \\ \vdots \\ N_{w_r}(G) \end{pmatrix}$$

The vector $\mathbf{b}$ can be computed using $r + 1$ #GRAPHSETHOM$^{\mathrm{C}}((\mathcal{H}, \lambda))$ oracle calls. Then

$$N\big(G \to H_i\big) = \sum_{w \in W_i} |[w]| N_w(G).$$

Thus, determining x is sufficient for computing the sought-for $N\big(G \to H_i\big)$. It remains to show that the matrix $\mathbf{A} \in \mathbb{Z}^{(r+1)\times(r+1)}$ is of full rank and therefore invertible. This can be easily seen by dividing each column by its first entry. The division is well-defined as for $t \in \{0 \ldots, r\}$ we have $\lambda_{w_t} \neq 0$ by definition of $\mathcal{H}^{\neq 0}$. The columns of the resulting matrix are pairwise different by the choice of (J, u) and as a consequence the resulting matrix is a Vandermonde matrix and therefore invertible. □

Next, we give a short technical lemma which follows from Definition 2.8 and is used in Lemma 2.15 to show that Theorem 2.13 gives hardness results for #COMP$^{\mathrm{C}}$(H).

Lemma 2.14. *Let H be a connected graph with at least one non-loop edge. Let H^- be the graph obtained from H by deleting exactly one non-loop edge (but keeping all vertices). If H^- is connected, then $\lambda_H(H^-) \neq 0$.*

Proof. As H^- is non-empty and connected, it is a valid input to λ_H and from the definition of λ_H (Definition 2.8) we obtain

$$\lambda_H(H^-) = - \sum_{\substack{H' \in \mathcal{S}_H \\ \text{s.t. } H' \not\cong H}} \mu_H(H')\lambda_{H'}(H^-).$$

Consider a graph $H' \in \mathcal{S}_H$ with $H' \not\cong H$ and $H' \not\cong H^-$. H' is a non-empty loop-hereditary connected subgraph of H and not isomorphic to H or H^-. Note that H^- is not isomorphic to any graph in $\mathcal{S}_{H'}$ which gives $\lambda_{H'}(H^-) = 0$. Furthermore, $\mu_H(H^-) \geq 1$. Thus, we proceed

$$\begin{aligned}\lambda_H(H^-) &= -\mu_H(H^-)\lambda_{H^-}(H^-) \\ &\leq -1.\end{aligned}$$

□

We now have most of the tools at hand to classify the complexity of #COMP(H). Tractability results come from Lemma 2.2. If H has a component that is not a reflexive clique or an irreflexive biclique then hardness will be lifted from Dyer and Greenhill's Theorem 1.4 via Theorem 2.5. The most difficult case is when all components of H are reflexive cliques or irreflexive bicliques, but some component is not an irreflexive star or a reflexive clique of size at most 2.

If H is connected then hardness will come from the following lemma, whose proof builds on the weighted graph set technology (Corollary 2.10) using Theorem 2.13 and Lemma 2.14 (using the stronger hardness result of Dyer and Greenhill, Theorem 2.1).

The remainder of the section generalises the connected case to the case in which H is not connected.

Lemma 2.15. *If H is a reflexive clique of size at least* 3 *then* #COMP^{C}(H) *is* #P*-hard. If H is an irreflexive biclique that is not a star then* #COMP^{C}(H) *is* #P*-hard.*

Proof. Suppose that H is a reflexive clique of size at least 3 or an irreflexive biclique that is not a star. Recall the definitions of $\mathcal{S}_H$, λ_H and weighted graph sets (Definitions 2.6, 2.7 and 2.8). Note that $(\mathcal{S}_H, \lambda_H)$ is a weighted graph set. Let H^- be a graph obtained from H by deleting a non-loop edge. Note that H^- is connected and it is not a reflexive clique or an irreflexive biclique. Thus Theorem 2.1 states that $\#\text{Hom}^{\text{C}}(\text{H}^-)$ is #P-complete. We will complete the proof of the lemma by showing $\#\text{Hom}^{\text{C}}(\text{H}^-) \le \#\text{Comp}^{\text{C}}(\text{H})$.

If H is a reflexive graph then the definition of $\mathcal{S}_H$ ensures that all graphs in $\mathcal{S}_H$ are reflexive. If H is an irreflexive bipartite graph, then the definition ensures that all graphs in $\mathcal{S}_H$ are irreflexive and bipartite. Since H^- is connected and therefore $\lambda_H(H^-) \neq 0$ by Lemma 2.14, we can apply Theorem 2.13 to the weighted graph set $(\mathcal{S}_H, \lambda_H)$ with $H^- \in \mathcal{S}_H$ to obtain $\#\text{Hom}^{\text{C}}(\text{H}^-) \le \#\text{GraphSetHom}^{\text{C}}((\mathcal{S}_{\text{H}}, \lambda_{\text{H}}))$. By Corollary 2.10, $\#\text{GraphSetHom}^{\text{C}}((\mathcal{S}_{\text{H}}, \lambda_{\text{H}})) \equiv \#\text{Comp}^{\text{C}}(\text{H})$. The lemma follows. □

We use the following two definitions in Lemmas 2.18 and 2.19 and in the proof of Theorem 1.5.

Definition 2.16. Let H be a graph. Suppose that every connected component that has more than j vertices is an irreflexive star. Suppose further that some connected component has j vertices and is not an irreflexive star. Let $\mathcal{A}(H)$ be the set of reflexive components of H with j vertices and let $\mathcal{B}(H)$ be the set of irreflexive non-star components of H with j vertices.

Definition 2.17. Let $L(H)$ denote the set of loops of a graph H. We define the graph $H^0 = (V(H), E(H) \setminus L(H))$.

Lemma 2.18. *Let H be a graph in which every component is a reflexive clique or an irreflexive biclique. If $J \in \mathcal{A}(H)$ then $\#\text{Comp}^{C}(\text{J}) \le \#\text{Comp}(H)$.*

Proof. Let H be a graph in which every component is a reflexive clique or an irreflexive biclique. Let $\mathcal{A}(H) = \{A_1, \ldots, A_k\}$. It follows from the definition of $\mathcal{A}(H)$ that all elements of $\mathcal{A}(H)$ are reflexive cliques of some size j (the same j for all graphs in $\mathcal{A}(H)$).

If $j \le 2$, the statement of the lemma is trivially true, since Lemma 2.2 shows that $\#\text{Comp}(A_i)$ is in FP, so the restricted problem $\#\text{Comp}^{\text{C}}(\text{A}_\text{i})$ is also in FP.

Now assume $j \ge 3$. Suppose without loss of generality that $J = A_1$. Let G be a (connected) input to $\#\text{Comp}^{\text{C}}(\text{J})$. For all $i \in [k]$, let $H \setminus A_i$ be the graph constructed from H by deleting the connected component A_i. Using Definition 2.17 we define the (irreflexive) graph $G' = (H \setminus J \oplus G)^0$ as an input to $\#\text{Comp}(H)$. Intuitively, to form G' from H we replace the connected component J with the graph G, then we delete all loops. We will prove the following claim.

Claim: Let $h\colon V(G') \to V(H)$ be a compaction from G' to H. Then the restriction $h|_{V(G)}$ is a compaction from G onto an element of $\mathcal{A}(H)$.

Proof of the claim: As h is a homomorphism, it maps each connected component of G' to a connected component of H. As, furthermore, h is a compaction and G'

and H have the same number of connected components, it follows that there exist connected components $C_1, \dots, C_k$ of G' such that for $i \in [k]$, $h|_{V(C_i)}$ is a compaction from C_i onto A_i. To prove the claim, we show that G is an element of $\mathcal{C} = \{C_1, \dots, C_k\}$. In order to use all vertices of a graph in $\mathcal{A}(H)$, i.e. a reflexive size-j clique, a graph in $\mathcal{C}$ has to have at least j vertices itself. Therefore and by the construction of G', an element of $\mathcal{C}$ can only be one of the following:

- a clique with j vertices,
- a biclique with j vertices,
- a star with at least j vertices
- or the copy of G.

Since $j \geq 3$, it is easy to see that there is no compaction from a star onto a clique with j vertices. In order to compact onto a reflexive clique of size j, an element of $\mathcal{C}$ also has to have at least $j(j-1)/2$ edges. Thus, $\mathcal{C}$ does not contain any bicliques. Finally, there are only $k-1$ connected components in G' that are j-vertex cliques other than (possibly) G. Therefore, G has to be an element of $\mathcal{C}$, which proves the claim. **(End of the proof of the claim.)**

Using the notation from Definition 2.17, the claim implies

$$N^{\mathrm{comp}}\big(G' \to H\big) = \sum_{i=1}^{k} N^{\mathrm{comp}}\big(G \to A_i\big) \cdot N^{\mathrm{comp}}\big((H \setminus A_i)^0 \to H \setminus A_i\big). \tag{2.5}$$

We can simplify the expression (2.5) using the fact that all elements of $\mathcal{A}(H)$ are reflexive size-j cliques:

$$N^{\mathrm{comp}}\big(G' \to H\big) = k \cdot N^{\mathrm{comp}}\big(G \to J\big) \cdot N^{\mathrm{comp}}\big((H \setminus J)^0 \to H \setminus J\big).$$

As $N^{\mathrm{comp}}\big((H \setminus J)^0 \to H \setminus J\big)$ does not depend on G, it can be computed in constant time. Thus, using a single $\#\textsc{Comp}(H)$ oracle call we can compute $N^{\mathrm{comp}}\big(G \to J\big)$ in polynomial time as required. □

Lemma 2.19. *Let H be a graph in which every component is a reflexive clique or an irreflexive biclique. If $\mathcal{A}(H)$ is empty but $\mathcal{B}(H)$ is non-empty, then there exists a component $J \in \mathcal{B}(H)$ such that $\#\textsc{Comp}^C(\mathrm{J}) \leq \#\textsc{Comp}(H)$.*

Proof. The proof is similar to that of Lemma 2.18. For completeness, we give the details. By Definition 2.16 the elements of $\mathcal{B}(H)$ are of the form $K_{a,b}$ with $a + b = j$ for some fixed j. As stars are excluded from $\mathcal{B}(H)$, we have $a, b \geq 2$. Let $\mathcal{B}^{\max}(H)$ denote the set of graphs with the maximum number of edges in $\mathcal{B}(H)$. The elements of $\mathcal{B}^{\max}(H)$ are pairwise isomorphic since the number of edges of a $K_{a,b}$ is $a \cdot b = a(j-a)$ and this function is strictly increasing for $a \leq j/2$. For concreteness, fix a and b so that each $J \in \mathcal{B}^{\max}(H)$ is isomorphic to $K_{a,b}$. Let $\mathcal{B}^{\max}(H) = \{B_1, \dots, B_k\}$. Take $J = B_1$.

For all $i \in [k]$, let $H \setminus B_i$ be the graph constructed from H by deleting the connected component B_i. Let $G' = (H \setminus J \oplus G)^0$ be an input to #Comp(H). We will prove the following claim.

Claim: Let $h\colon V(G') \to V(H)$ be a compaction from G' to H. Then the restriction $h|_{V(G)}$ is a compaction from G onto an element of $\mathcal{B}^{\max}(H)$.

Proof of the claim: As h is a homomorphism, it maps each connected component of G' to a connected component of H. As, furthermore, h is a compaction and G' and H have the same number of connected components, it follows that there exist connected components $C_1, \ldots, C_k$ of G' such that for $i \in [k]$, $h|_{V(C_i)}$ is a compaction from C_i onto B_i. To prove the claim, we show that G is an element of $\mathcal{C} = \{C_1, \ldots, C_k\}$. In order to compact onto a graph in $\mathcal{B}^{\max}(H)$, a graph in $\mathcal{C}$ has to have at least j vertices and $a \cdot b$ edges itself. By the construction of G' and the fact that $\mathcal{A}(H)$ is empty, a connected component in G' with at least j vertices and $a \cdot b$ edges can only be one of the following:

- a biclique $K_{a,b}$,
- a star with at least j vertices and at least $a \cdot b$ edges
- or the copy of G.

Since $a, b \geq 2$, it is easy to see that there is no compaction from a star onto a $K_{a,b}$. Finally, there are only $k-1$ connected components in G' that are bicliques of the form $K_{a,b}$ other than (possibly) G. Therefore, G has to be an element of $\mathcal{C}$, which proves the claim. **(End of the proof of the claim.)**

Using the notation from Definition 2.17, the claim implies

$$N^{\text{comp}}\big(G' \to H\big) = \sum_{i=1}^{k} N^{\text{comp}}\big(G \to B_i\big) \cdot N^{\text{comp}}\big((H \setminus B_i)^0 \to H \setminus B_i\big). \tag{2.6}$$

We can simplify the expression (2.6) using the fact that all elements of $\mathcal{B}^{\max}(H)$ are of the form $K_{a,b}$:

$$N^{\text{comp}}\big(G' \to H\big) = k \cdot N^{\text{comp}}\big(G \to J\big) \cdot N^{\text{comp}}\big((H \setminus J)^0 \to H \setminus J\big).$$

As $N^{\text{comp}}\big((H \setminus J)^0 \to H \setminus J\big)$ does not depend on G, it can be computed in constant time. Thus, using a single #Comp(H) oracle call we can compute $N^{\text{comp}}\big(G \to J\big)$ in polynomial time as required. □

Finally, we prove the main theorem of this section, which we restate at this point.

Theorem 1.5. *Let H be a graph. If every connected component of H is an irreflexive star or a reflexive clique of size at most 2 then #Comp(H) and #LComp(H) are in* FP. *Otherwise, #Comp(H) and #LComp(H) are #P-complete.*

Proof. The membership of #Comp(H) in #P is straightforward. We distinguish between a number of cases depending on the graph H.

Case 1: Suppose that every connected component of H is an irreflexive star or a reflexive clique of size at most 2. Then #LComp(H) is in FP by Lemma 2.2.

Case 2: Suppose that H contains a component that is not a reflexive clique or an irreflexive biclique. Then the hardness of #Hom(H) (from Theorem 1.4) together with the reduction #Hom(H) $\leq$ #Comp(H) (from Theorem 2.5) implies that #Comp(H) is #P-hard. The hardness of #LComp(H) follows from the trivial reduction from #Comp(H) to #LComp(H).

Case 3: Suppose that the components of H are reflexive cliques or irreflexive bicliques and that H contains at least one component that is not an irreflexive star or a reflexive clique of size at most 2. Every graph $J \in \mathcal{A}(H) \cup \mathcal{B}(H)$ is a reflexive clique of size at least 3 or an irreflexive biclique that is not a star. By Lemma 2.15, #Comp$^{\mathrm{C}}$(J) is #P-complete. Finally, as $\mathcal{A}(H) \cup \mathcal{B}(H)$ is non-empty, we can use either Lemma 2.18 or Lemma 2.19 to obtain the existence of $J \in \mathcal{A}(H) \cup \mathcal{B}(H)$ with #Comp$^{\mathrm{C}}$(J) $\leq$ #Comp(H). This implies that #Comp(H) is #P-hard. As in Case 2, the hardness of #LComp(H) follows from the trivial reduction from #Comp(H) to #LComp(H). □

2.3 Counting Surjective Homomorphisms

The proof of Theorem 1.6 is divided into two sections. The first of these deals with tractable cases and the second deals with hardness results and also contains the proof of the final theorem. Taken together, Theorem 1.6 and Dyer and Greenhill's Theorem 1.4 show that the problem of counting surjective homomorphisms to a fixed graph H has the same complexity characterisation as the problem of counting all homomorphisms to H.

Section 2.3.3 shows that this equivalence disappears in the uniform case, where H is part of the input, rather than being a fixed parameter of the problem. Specifically, Theorem 2.23 demonstrates a setting in which counting surjective homomorphisms is more difficult than counting all homomorphisms (assuming FP $\neq$ #P).

2.3.1 Tractability Results

Theorem 2.20. *Let H be a graph. Then #LSHom(H) $\leq$ #LHom(H).*

Proof. Let H be fixed and $|V(H)| = q$. Let $(G, \mathbf{S})$ be an input instance of #LSHom(H). Let $(v_1, \ldots, v_n)$ be the vertices of G in an arbitrary but fixed order. With respect to this ordering and with respect to a homomorphism from G to H, let us denote by v_{i_1} the first vertex of G which is assigned the first new vertex of H ($v_{i_1} = v_1$), v_{i_2} the first vertex of G which is assigned the second new vertex of H and so on. Every surjective homomorphism from G to H contains exactly one subsequence $\mathbf{v} = (v_{i_1}, \ldots, v_{i_q})$ and every homomorphism containing such a subsequence is surjective. The number of subsequences is bounded from above by $\binom{n}{q}$. Let $\sigma \colon \mathbf{v} \to V(H)$ be an assignment of the vertices of H to the vertices in $\mathbf{v}$. There are $q!$ such assignments.

We call $\psi = (\mathbf{v}, \sigma)$ a *configuration* of G and $\Psi(G)$ the set of all configurations for the given G. For every such configuration ψ we create a #LHom(H) instance $(G, \mathbf{S}^\psi)$ with $\mathbf{S}^\psi = \{S^\psi_{v_i} \subseteq V(H) \ : \ i \in [n]\}$ and

$$S^\psi_{v_i} = \begin{cases} S_{v_i} \cap \{\sigma(v_{i_j})\}, & \text{if } i = i_j \text{ for } j \in [q] \\ S_{v_i} \cap \{\sigma(v_{i_1}), \dots, \sigma(v_{i_j})\}, & \text{for } i_j < i < i_{j+1}. \end{cases}$$

Intuitively, we use lists to "pin" the vertices in $\mathbf{v}$ to the vertices assigned by σ and to prohibit the remainder of the vertices of G from being mapped to new vertices of H. Then

$$N^{\text{sur}}\big((G, \mathbf{S}) \to H\big) = \sum_{\psi \in \Psi(G)} N\big((G, \mathbf{S}^\psi) \to H\big)$$

We can compute $N^{\text{sur}}\big((G, \mathbf{S}) \to H\big)$ by making a #LHom(H) oracle call for every instance $(G, \mathbf{S}^\psi)$ and adding the results. The number of oracle calls $|\Psi(G)|$ is bounded from above by the polynomial $q!\binom{n}{q} \leq n^q$. □

Corollary 2.21. *Let H be a graph. If every connected component of H is a reflexive clique or an irreflexive biclique then* #LSHom(H) *is in* FP.

Proof. The statement follows directly from Theorem 2.20 using Dyer and Greenhill's dichotomy from Theorem 1.4. □

2.3.2 Hardness Results

The following result and proof are very similar to that of Theorem 2.5 and Lemma 2.4, respectively. For completeness, we repeat the proof in detail.

Theorem 2.22. *Let H be a graph. Then* #Hom(H) $\leq$ #SHom(H).

Proof. Let $|V(H)| = q$ and G be an input to #Hom(H). We design a graph $G_t = G \oplus W_t$ as an input to the problem #SHom(H) by adding a set W_t of t new isolated vertices to the graph G.

We introduce some additional notation. Let $S^k(G)$ be the number of homomorphisms σ from G to H that use exactly k of the vertices of H. Let $\{w_1, \dots, w_k\}$ be a set of k arbitrary but fixed vertices from H. We define $N^k(W_t)$ as the number of homomorphisms τ from W_t to H such that $\{w_1, \dots, w_k\}$ are amongst the vertices used by τ. The particular choice of vertices $\{w_1, \dots, w_k\}$ is not important when counting homomorphisms from a set of isolated vertices—$N^k(W_t)$ only depends on the numbers k and t.

We observe that, for each surjective homomorphism $\gamma \colon V(G_t) \to V(H)$, the restriction $\gamma|_{V(G)}$ uses a subset $V' \subseteq V(H)$ of vertices and does not use any vertices outside of V'. Suppose that V' has cardinality $|V'| = q - k$ for some $k \in \{0, \dots, q\}$. Then $\gamma|_{W_t}$ uses at least the remaining k fixed vertices of H.

Therefore, we obtain the following linear equation for a fixed $t \geq 0$:

$$\underbrace{N^{\text{sur}}\big(G_t \to H\big)}_{b_t} = \sum_{k=0}^{q} \underbrace{S^{q-k}(G)}_{x_k} \underbrace{N^k(W_t)}_{a_{t,k}}.$$

By choosing $q+1$ different values for the parameter t we obtain a system of linear equations. Here, we choose $t = 0, \ldots, q$. Then the system is of the form $\mathbf{b} = \mathbf{A}\mathbf{x}$ for

$$\mathbf{b} = \begin{pmatrix} b_0 \\ \vdots \\ b_q \end{pmatrix}, \qquad \mathbf{A} = \begin{pmatrix} a_{0,0} & \cdots & a_{0,q} \\ \vdots & \ddots & \vdots \\ a_{q,0} & \cdots & a_{q,q} \end{pmatrix}, \qquad \text{and} \qquad \mathbf{x} = \begin{pmatrix} x_0 \\ \vdots \\ x_q \end{pmatrix}.$$

Note, that the vector $\mathbf{b}$ can be computed using $q+1$ $\#\text{SH}\textsc{om}(H)$ oracle calls. Further,

$$\sum_{k=0}^{q} x_k = \sum_{k=0}^{q} S^{q-k}(G) = \sum_{k=0}^{q} S^{k}(G) = N\big(G \to H\big).$$

Thus, determining $\mathbf{x}$ is sufficient for computing the sought-for $N\big(G \to H\big)$. It remains to show that the matrix $\mathbf{A}$ is of full rank and is therefore invertible.

For $t < k$, clearly $a_{t,k} = N^k(W_t) = 0$. Further, for the diagonal elements we have $a_{t,t} = N^t(W_t) = t!$ for $t \in \{0, \ldots, q\}$. Hence,

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ * & 1! & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ * & \cdots & * & q! \end{pmatrix}$$

is a triangular matrix with non-zero diagonal entries, which completes the proof. □

Theorem 1.6. *Let H be a graph. If every connected component of H is a reflexive clique or an irreflexive complete bipartite graph, then $\#\text{SH}\textsc{om}(H)$ and $\#\text{LSH}\textsc{om}(H)$ are in* FP. *Otherwise, $\#\text{SH}\textsc{om}(H)$ and $\#\text{LSH}\textsc{om}(H)$ are #P-complete.*

Proof. The easiness result follows from Corollary 2.21 using the trivial reduction $\#\text{SH}\textsc{om}(H) \leq \#\text{LSH}\textsc{om}(H)$. The hardness result follows from the same trivial reduction, along with Theorem 2.22 and the dichotomy for $\#\text{H}\textsc{om}(H)$ from Theorem 1.4.

□

2.3.3 The Uniform Case

We have seen from Theorems 1.4 and 1.6 that the problem of counting homomorphisms to a fixed graph H has the same complexity as the problem of counting *surjective* homomorphisms to H.

Nevertheless, there are scenarios in which counting problems involving surjective homomorphisms are more difficult than those involving unrestricted homomorphisms. To illustrate this point, we consider the following *uniform* homomorphism-counting problems. Motivated by terminology from constraint satisfaction, we use "uniform" to indicate that the target graph H is part of the input, rather than being a fixed parameter.

Name: UNIFORM#HOMTOCLIQUES.
Input: Irreflexive graph G whose components are cliques and reflexive graph H whose components are cliques.
Output: $N\big(G \to H\big)$.

Name: UNIFORM#SHOMTOCLIQUES.
Input: Irreflexive graph G whose components are cliques and reflexive graph H whose components are cliques.
Output: $N^{\mathrm{sur}}\big(G \to H\big)$.

The main result of this section is the following theorem.

Theorem 2.23. UNIFORM#HOMTOCLIQUES *is in* FP *but* UNIFORM#SHOMTOCLIQUES *is* #P-*complete.*

In order to prove Theorem 2.23, we define a counting variant of the subset sum problem. Given a set of integers $\mathcal{A} = \{a_1, \ldots, a_n\}$ and an integer b let $S(\mathcal{A}, b)$, be the number of subsets $\mathcal{A}' \subseteq \mathcal{A}$ such that the sum of the elements in $\mathcal{A}'$ is equal to b. The counting problem is stated as follows.

Name: #SUBSETSUM.
Input: A set of positive integers $\mathcal{A} = \{a_1, \ldots, a_n\}$ and a positive integer b.
Output: $S(\mathcal{A}, b)$.

It is well known that #SUBSETSUM is #P-complete (see for instance the textbook by Papadimitriou [126, Theorems 9.9, 9.10 and 18.1]). Thus, Theorem 2.23 follows immediately from Lemmas 2.24 and 2.25.

Lemma 2.24. UNIFORM#HOMTOCLIQUES *is in* FP*.*

Proof. Let G and H be an input instance of UNIFORM#HOMTOCLIQUES. Let k be the number of connected components of G and let $a_1, \ldots, a_k$ be the number of vertices of these components, respectively. Let H have q connected components with $b_1, \ldots, b_q$ vertices, respectively. Then, as all components are cliques and H is reflexive,

$$N\big(G \to H\big) = \prod_{i=1}^{k} \sum_{j=1}^{q} b_j^{a_i}.$$

Thus, it is easy to compute $N\big(G \to H\big)$. □

Lemma 2.25. #SUBSETSUM $\leq$ UNIFORM#SHOMTOCLIQUES.

Proof. Let $\mathcal{A} = \{a_1, \ldots, a_k\}$, b be an input instance of #SUBSETSUM. We define $N = \sum_{i=1}^{k} a_i$. Now, we design a polynomial time algorithm to determine $S(\mathcal{A}, b)$ using an oracle for Uniform#SHomToCliques. If $N < b$, we have $S(\mathcal{A}, b) = 0$. Now assume $N \geq b$. We create an input of Uniform#SHomToCliques as follows. We set G to be an irreflexive graph with a connected component G_i for each $i \in [k]$, where G_i is a clique with a_i vertices. Furthermore, we set H to be a reflexive graph with two connected components H_1 and H_2. Let H_1 be a clique with b vertices and let H_2 be a clique with $N - b$ vertices. By $\left\{ n \atop k \right\}$ we denote the Stirling number of the second kind, i.e. the number of partitions of a set of n elements into k non-empty subsets. By definition, we have $\left\{ n \atop k \right\} = 0$ if $n < k$.

Let $h \colon V(G) \to V(H)$ be a homomorphism from G to H and let b' be the number of vertices of G that are mapped to the connected component H_1. Note that h has to map each connected component of G to a connected component of H. By the construction of G, this implies that there exists a subset $\mathcal{A}' \subseteq \mathcal{A}$ such that the sum of elements in $\mathcal{A}'$ is equal to b'. Furthermore, as all connected components of G and H are cliques and H is reflexive, the number of surjective homomorphisms from G to H that assign exactly b' fixed vertices to H_1 is equal to the number of surjective mappings from $[b']$ to $[b]$, which is $b!\left\{ b' \atop b \right\}$. Therefore, we can express $N^{\text{sur}}\big(G \to H\big)$ as follows.

$$N^{\text{sur}}\big(G \to H\big) = \sum_{b'=0}^{N} S(\mathcal{A}, b') \cdot b!\left\{ b' \atop b \right\} \cdot (N-b)!\left\{ N-b' \atop N-b \right\}, \tag{2.7}$$

where the factor $(N-b)!\left\{ N-b' \atop N-b \right\}$ corresponds to the number surjective mappings from the remaining $N - b'$ fixed vertices of G to the component H_2. Finally, we use the fact that the summands in (2.7) are non-zero only if $b' \geq b$ and $N - b' \geq N - b$, which implies $b' = b$. Thus,

$$\begin{aligned} N^{\text{sur}}\big(G \to H\big) &= S(\mathcal{A}, b) \cdot b!\left\{ b \atop b \right\} \cdot (N-b)!\left\{ N-b \atop N-b \right\} \\ &= b!(N-b)! \cdot S(\mathcal{A}, b). \end{aligned}$$

□

2.4 Reductions and Retractions

The following simple observation follows directly from the problem definitions.

Observation 2.26. *Let H be a graph. Then*

$$\#\textsc{Ret}(H) \leq \#\textsc{LHom}(H) \textit{ and } \#\textsc{Hom}(H) \leq \#\textsc{Ret}(H).$$

Observation 2.26 together with the Dyer and Greenhill dichotomy (Theorem 1.4) immediately implies the following dichotomy characterisation for the problem of counting retractions.

Corollary 2.27. *Let H be a graph. If every connected component of H is a reflexive clique or an irreflexive biclique, then #Ret(H) is in FP. Otherwise, #Ret(H) is #P-complete.*

Therefore, we can use our Theorems 1.5 and 1.6 to obtain the following reductions.

Lemma 1.7. *Let H be a graph. Then*

$$\#\textsc{Hom}(H) \equiv \#\textsc{LHom}(H) \equiv \#\textsc{SHom}(H) \equiv \#\textsc{LSHom}(H) \equiv \#\textsc{Ret}(H) \leq \#\textsc{Comp}(H) \equiv \#\textsc{LComp}(H).$$

Furthermore, there is a graph H for which $\#\textsc{Comp}(H)$ and $\#\textsc{LComp}(H)$ are #P-complete, but $\#\textsc{Hom}(H)$, $\#\textsc{LHom}(H)$, $\#\textsc{SHom}(H)$, $\#\textsc{LSHom}(H)$ and $\#\textsc{Ret}(H)$ are in FP*.*

Proof. Theorems 1.4, 1.5, 1.6 and Corollary 2.27 give complexity classifications for all of the problems. The reductions in the corollary follow from three easy observations.

- All problems in FP are trivially inter-reducible.
- All #P-complete problems are inter-reducible.
- All problems in FP are reducible to all #P-complete problems.

The separating graph H can be taken to be any reflexive clique of size at least 3 or any irreflexive biclique that is not a star. □

Part II

Approximate Counting

Chapter 3

Approximately Counting Retractions to Graphs of Girth at least 5

I have had my results for a long time; but I do not yet know how I am to arrive at them.

–Carl Friedrich Gauß, quoted in *The Mind and the Eye (1954)*

This chapter is based on the following paper:

[58] Jacob Focke, Leslie Ann Goldberg, and Stanislav Živný. The complexity of approximately counting retractions. *ACM Transactions of Computation Theory*, 12(3):Art. 15, 43, 2020.

- A preliminary version of this work appeared in the Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, pp. 2205-2215.

Parts of Section 3.4 are from the addendum of

[57] Jacob Focke, Leslie Ann Goldberg, and Stanislav Živný.
The complexity of counting surjective homomorphisms and compactions.
SIAM Journal on Discrete Mathematics, 33(2):1006–1043, 2019.

- A preliminary version of this work appeared in the Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, pp. 1772-1781.

Contents

Organisation of this Chapter

There are two main parts to this chapter.

First, in Section 3.2, we prove Theorem 1.11 (the complexity trichotomy of approximately counting retractions for graphs with girth at least 5). Even though this result is generalised in Chapter 4, we present the proof of Theorem 1.11 in full, as the more general Theorem 1.13 builds upon it. In Section 3.2.1, we concentrate on irreflexive graphs H. Notably, the corresponding classification (Theorem 3.7) already covers all irreflexive square-free graphs H, including those that have triangles.

Second, in Section 3.3, we locate the problem of approximately counting retractions in the complexity landscape of related counting homomorphisms problems. The main results of this part are an AP-reduction from counting compactions to counting retractions, and an AP-reduction from counting surjective homomorphisms to counting retractions. In Section 3.4, we give some additional reductions that complete our current picture of the approximate counting homomorphisms complexity landscape.

We start off by giving, in Section 3.1.1, an overview of the methods used in these proofs.

3.1 Introduction

3.1.1 Methods

In the proof of Theorem 1.11 we use several different techniques. In the #BIS-easiness proof for partially bristled reflexive paths (Lemma 3.8) we build upon a technique that was introduced by Dyer et al. [37] and extended by Kelk [108] to reduce the problem of approximately counting homomorphisms to the problem of approximately counting the downsets of a partial order. In order to obtain more general results, we formalise this technique and use it in the context of the constraint satisfaction framework. This framework is convenient for generating #BIS-easiness results, not only for counting homomorphisms but also for counting retractions, both in the setting of undirected graphs (as used in this chapter) and even in the setting of directed graphs.

In order to obtain the #SAT-hardness part of Theorem 1.11, we analyse, and classify, different local structures in graphs. For instance, we use modifications of, and a more careful analysis of, a gadget from [76] to prove the #SAT-hardness for irreflexive square-free graphs that have an induced subgraph J_3 (Lemma 3.6). Some hardness results are also based on NP-completeness results for the retraction decision problem that carry over to the approximate counting version.

The algorithms captured by the reductions $\#\textsc{SHom}(H) \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$ and $\#\textsc{Comp}(H) \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$ are based on a Monte Carlo approach (Lemma 3.23). We will discuss this approach here in the context of counting surjective homomorphisms to H. The details, and the related approach for counting compactions, are described in Section 3.3.1. The Monte Carlo approach is applicable for a reduction from $\#\textsc{SHom}(H)$ to $\#\textsc{Ret}(H)$ because $\#\textsc{Ret}(H)$ is a so-called self-reducible problem. Recall that the output of $\#\textsc{Ret}(H)$, given a graph G and lists $\mathbf{S} = \{S_v \subseteq V(H) \mid v \in V(G)\}$ such that, for all $v \in V(G)$, $|S_v| \in \{1, |V(H)|\}$, is the number of homomorphisms from $(G, \mathbf{S})$ to H. Using an oracle for approximating this number, it is also possible to *sample* a homomorphism from $(G, \mathbf{S})$ to H approximately uniformly at random (following the general method of Jerrum, Valiant, and Vazirani [105]). A naive approach for sampling surjective homomorphisms (and hence for approximately counting them) is as follows: Start with an input G to $\#\textsc{SHom}(H)$. Let $\mathbf{S}$ be the trivial set of lists $\mathbf{S} = \{S_v = V(H) \mid v \in V(G)\}$. Using the oracle for $\#\textsc{Ret}(H)$, obtain a random homomorphism from $(G, \mathbf{S})$ to H (which is just a random homomorphism from G to H). Reject (and repeat) if this homomorphism is not surjective. Eventually, we obtain a random surjective homomorphism from G to H, as required. While this approach is certainly straightforward, it does not lead to an *efficient* algorithm for sampling surjective homomorphisms because the number of surjective homomorphisms might be very small compared to the total number of homomorphisms.

Our method to shrink the sample space is based on the following fact. For every surjective homomorphism h from G to H there exists a constant-size set of vertices $U \subseteq V(G)$ such that the restriction of h to U is already surjective. We can enumerate all these constant-size sets U and use single vertex lists to fix their images. Consequently we obtain a (polynomial) number of instances $(G, \mathbf{S}^1), \dots, (G, \mathbf{S}^k)$ of the problem $\#\textsc{Ret}(H)$. For $i \in \{1, \dots, k\}$ let R_i be the set of homomorphisms from $(G, \mathbf{S}^i)$ to H. Then the set of surjective homomorphisms from G to H is the union $R = \bigcup_{i=1}^{k} R_i$. The final building block of our reduction is the idea that we can sample the union R by first sampling from the disjoint union $R^+ = \bigcup_{i=1}^{k} \{(h, i) \mid h \in R_i\}$. This idea is explained more generally, for instance, in [123, Section 11.2.2]. The point is, that we can sample uniformly from R^+ by using a $\#\textsc{Ret}(H)$ oracle, and the union R is relatively dense in the disjoint union R^+ (its size is at least $|R^+|/k$). So we can obtain a sample from R. Then the samples can be combined to obtain, with high probability, an approximate count. A lot of AP-reductions are based on the use of gadgets and we have not seen the use of Monte Carlo algorithms in AP-reductions before.

3.1.2 Preliminaries

For a positive integer n let $[n] = \{1, \ldots, n\}$.

Graph Theory Recall that a *graph* may or may not have loops, but does not have parallel edges. If not stated otherwise, a graph is assumed to be undirected. We use $\mathcal{H}((G, \mathbf{S}), H)$ to denote the set of homomorphisms from $(G, \mathbf{S})$ to H and we use $N\big((G, \mathbf{S}) \to H\big)$ to denote the size of $\mathcal{H}((G, \mathbf{S}), H)$, i.e., the number of homomorphisms.

A *cycle* is a walk $w_0 w_1 \cdots w_k w_0$ where $k > 1$ and all vertices in $\{w_0, \ldots, w_k\}$ are distinct. The *length* of the cycle is $k + 1$. As mentioned in the introduction, we sometimes refer to length-3 cycles as "triangles" and to length-4 cycles as "squares". The *girth* of a graph H is the length of a shortest cycle in H. If H is acyclic (that is if H is a forest with possibly some loops) then its girth is infinity. A *tree* may be irreflexive, reflexive, or neither, but it may not have any cycles.

As partially bristled reflexive paths appear in a number of our results we give a formal definition of this class of graphs. We also give an example in Figure 3.1.

Definition 3.1. A *partially bristled reflexive path* is a reflexive path, or a tree with the following form. Let Q be a positive integer and let S be a non-empty subset of $[Q]$. Then $V(H) = \{c_0, \ldots, c_{Q+1}\} \cup \bigcup_{i \in S} \{g_i\}$ and $E(H) = \bigcup_{i=0}^{Q} \{c_i, c_{i+1}\} \cup \bigcup_{i=0}^{Q+1} \{c_i, c_i\} \cup \bigcup_{i \in S} \{c_i, g_i\}$.

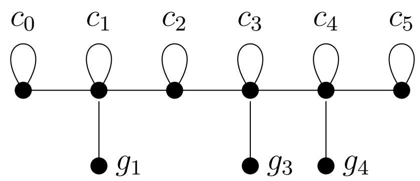


Figure 3.1: Partially bristled reflexive path with $Q = 4$ and $S = \{1, 3, 4\}$.

For a graph H and a vertex $u \in V(H)$ we define the *(distance-1) neighbourhood of* u as $\Gamma(u) = \{v \in V(H) \mid \{v, u\} \in E(H)\}$. Similarly, the *distance-2 neighbourhood of* u is defined as $\Gamma^2(u) = \{v \in V(H) \mid \exists w \in V(H) : \{v, w\}, \{w, u\} \in E(H)\}$. Let U be a subset of $V(H)$. Then $\Gamma(U) = \bigcap_{u \in U} \Gamma(u)$ is the *set of common neighbours* of the vertices in U. The set of vertices that have a neighbour in S is denoted by $\Phi(S) = \bigcup_{v \in S} \Gamma(v)$.

We will also use the concept of *induced graphs.* Given a subset U of $V(H)$, the graph $H[U] = (U, \{\{u_1, u_2\} \in E(H) \mid u_1, u_2 \in U\})$ is called *the subgraph of H induced by U*. The graph $H' = (V', E')$ is a *subgraph* of $H = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$.

Connected Graphs We conclude this section with some simple remarks regarding the connectivity of graphs when investigating the complexity of counting retractions. We will show that in the context of approximately counting retractions we can restrict to connected graphs without loss of generality. We define the following problem which restricts the input to connected graphs.

Name: $\#\mathrm{RET}^{\mathrm{C}}(H)$.
Input: An irreflexive *connected* graph G and a collection of lists $\mathbf{S} = \{S_v \subseteq V(H) \mid v \in V(G)\}$ such that, for all $v \in V(G)$, $|S_v| \in \{1, |V(H)|\}$.
Output: $N\big((G, \mathbf{S}) \to H\big)$.

The following observation is well known.

Observation 3.2. *Let H be a graph. Then $\#\mathrm{RET}(H) \equiv_{\mathrm{AP}} \#\mathrm{RET}^{C}(H)$.*

Proof. The fact that $\#\mathrm{RET}(H) \geq_{\mathrm{AP}} \#\mathrm{RET}^{\mathrm{C}}(H)$ is trivial. We now show that $\#\mathrm{RET}(H) \leq_{\mathrm{AP}} \#\mathrm{RET}^{\mathrm{C}}(H)$. Let $(G, \mathbf{S})$ be an instance of $\#\mathrm{RET}(H)$ and let $\varepsilon \in (0, 1)$ be the desired precision. Let $C_1, \ldots, C_k$ be the connected components of G. For each $i \in [k]$, let $\mathbf{S}_i = \{S_v \mid v \in V(C_i)\}$. Then $N\big((G, \mathbf{S}) \to H\big) = \prod_{i=1}^{k} N\big((C_i, \mathbf{S}_i) \to H\big)$. The algorithm which, for each $i \in [k]$, makes a $\#\mathrm{RET}^{\mathrm{C}}(H)$ oracle call with precision $\delta = \varepsilon/k$ and input $(C_i, \mathbf{S}_i)$, and returns the product of outputs, approximates $N\big((G, \mathbf{S}) \to H\big)$ with the desired precision. □

Remark 3.3. Let H be a graph with connected components $H_1, \ldots, H_k$ and let $(G, \mathbf{S})$ be an input to $\#\mathrm{RET}^{\mathrm{C}}(H)$. For $j \in [k]$ let $S_v^j = S_v \cap V(H_j)$ and let $\mathbf{S}^j = \{S_v^j \mid v \in V(G)\}$. Then, as G is connected, it holds that

$$N\big((G, \mathbf{S}) \to H\big) = \sum_{j \in [k]} N\big((G, \mathbf{S}^j) \to H_j\big).$$

Therefore, given an oracle for $\#\mathrm{RET}^{\mathrm{C}}(H_j)$ for each $j \in [k]$, we obtain an algorithm for $\#\mathrm{RET}^{\mathrm{C}}(H)$. By Observation 3.2 this means that given an oracle for $\#\mathrm{RET}(H_j)$ for each $j \in [k]$, we obtain an algorithm for $\#\mathrm{RET}(H)$.

In the opposite direction, it is straightforward to see that for each $j \in [k]$ we have $\#\mathrm{RET}^{\mathrm{C}}(H_j) \leq_{\mathrm{AP}} \#\mathrm{RET}^{\mathrm{C}}(H)$ (and therefore $\#\mathrm{RET}(H_j) \leq_{\mathrm{AP}} \#\mathrm{RET}(H)$). The details are as follows: Let $(G, \mathbf{S})$ be an input to $\#\mathrm{RET}^{\mathrm{C}}(H_j)$. If all lists in $\mathbf{S}$ have size 1, computing $N\big((G, \mathbf{S}) \to H_j\big)$ is trivial. Otherwise we fix some vertex $v \in V(G)$ with $S_v = V(H_j)$. For each $u \in V(H)$ and $w \in V(G)$ we define

$$S_w^u = \begin{cases} \{u\}, & \text{if } w = v \\ S_w, & \text{if } |S_w| = 1 \\ V(H), & \text{otherwise.} \end{cases}$$

and $\mathbf{S}^u = \{S_w^u \mid w \in V(G)\}$. As G is connected, a homomorphism from G to H maps all vertices of G to the same connected component of H. Therefore, $N\big((G, \mathbf{S}) \to H_j\big) = \sum_{u \in V(H_j)} N\big((G, \mathbf{S}^u) \to H\big)$. This shows that $|V(H_j)|$ calls to a $\#\mathrm{RET}^{\mathrm{C}}(H)$ oracle are sufficient to approximate the number of retractions to a component H_j of H.

Miscellanea Sometimes it is useful to switch between different notions of accuracy. To this end we use the following observation which follows immediately from the Taylor expansion of the exponential function.

Observation 3.4. *Let ε be in $(0,1)$. Then $1+\varepsilon \le e^{\varepsilon} \le 1+2\varepsilon$ and $1-\varepsilon \le e^{-\varepsilon} \le 1-\varepsilon/2$.*

3.2 Approximately Counting Retractions to Graphs without short Cycles

We start off by restricting to irreflexive graphs in Section 3.2.1. The corresponding classification (Theorem 3.7) is for irreflexive square-free graphs. Subsequently, in Section 3.2.2, we consider graphs that have at least one loop. In this case, we restrict to graphs of girth at least 5. The two results together give a classification for all graphs of girth at least 5 (Theorem 1.11), the proof of which is presented in Section 3.2.3. Afterwards, in Chapter 4, we will extend this result to a complete classification for all square-free graphs.

3.2.1 Irreflexive Square-free Graphs

The goal of this section is to prove Theorem 3.7. The most difficult part is Lemma 3.6, which shows that, if H is a square-free graph containing an induced J_3, then $\#\mathrm{SAT} \le_{\mathrm{AP}} \#\textsc{Ret}(H)$.

The proof of Lemma 3.6 generalises ideas from the proof of Lemma 3.6 of [76], so we start with some definitions from there. A *multiterminal cut* of a graph G with distinguished vertices α, β and γ (called *terminals*) is a set of edges $E' \subseteq E(G)$ that disconnects the terminals (i.e. ensures that there is no path in $(V(G), E(G) \setminus E')$ that connects any two distinct terminals). The *size* of a multiterminal cut is its cardinality. We consider the following computational problem.

Name: $\#\textsc{MultiterminalCut}(3)$.
Input: A connected irreflexive graph G with 3 distinct terminals $\alpha,\ \beta,\ \gamma \in V(G)$ and a positive integer B. The input has the property that every multiterminal cut has size at least B.
Output: The number of size-B multiterminal cuts of G with terminals α, β and γ.

For motivation we consider the case where H is a tree. Suppose that H is an irreflexive tree with an induced J_3, labelled as in Figure 3.2. Lemma 3.6 of [76] gives an AP-reduction from $\#\textsc{MultiterminalCut}(3)$ to the problem of counting "weighted" homomorphisms to H. In fact, the weights used in the proof are Boolean values, so the proof actually reduces $\#\textsc{MultiterminalCut}(3)$ to the problem of counting list homomorphisms to H. Given an input G, α, β, γ of $\#\textsc{MultiterminalCut}(3)$, a homomorphism instance is created in which lists ensure that the terminals α, β and γ are mapped to the vertices x_0, y_0 and z_0 of J_3, respectively. Lists also ensure that all other vertices of G are mapped to $\{x_0, y_0, z_0\}$. Finally, lists ensure that all remaining

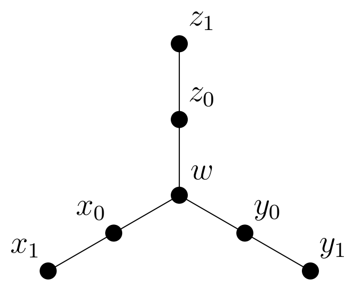


Figure 3.2: The graph J_3.

vertices of the homomorphism instance are mapped to the vertices $\{w, x_1, y_1, z_1\}$ of J_3. Our proof shows how to refine the gadgets so that lists have size 1 or size $|V(H)|$. Thus, our reduction is to the more refined problem #RET(H). We also show how to handle graphs H that are not trees (as long as they are square-free). The details are given in the proof of Lemma 3.6. We use the following technical lemma, known as Dirichlet's approximation lemma, which bounds the extent to which reals can be approximated by integers. Using this lemma is a standard technique in this line of research (see for instance [62]).

Lemma 3.5 ([131, p. 34]). *Let $\lambda_1, \ldots, \lambda_d > 0$ be real numbers and N be a natural number. Then there exist positive integers $p_1, \ldots, p_d, r$ with $r \leq N$ such that $|r\lambda_i - p_i| \leq 1/N^{1/d}$ for every $i \in [d]$.*

Using Lemma 3.5, we can prove our main lemma.

Lemma 3.6. *Let H be an irreflexive square-free graph that contains an induced J_3. Then #SAT $\leq_{\text{AP}}$ #RET(H).*

Proof. Suppose that H is a square-free graph with q vertices and an induced J_3, which we label as shown in Figure 3.2. The problem #MULTITERMINALCUT(3) is shown in [76, Lemma 3.5] to be equivalent to #SAT with respect to AP-reductions. We will give a reduction from #MULTITERMINALCUT(3) to #RET(H).

Let G, α, β, γ, B be an instance of #MULTITERMINALCUT(3) with $n = |V(G)|$ and let ε be an error bound in $(0, 1)$. From this instance we construct an input $(J, \mathbf{S})$ to #RET(H) as follows. Each of the terminals α, β and γ will be a vertex of J. J will also have a vertex ω which is distinct from α, β and γ. Let s_α, s_β and s_γ be positive integers (we will give their precise values later). For every edge $e = \{u, v\} \in E(G)$ we define the set of vertices

$$V'(e) = \{(e, \alpha, 1), \ldots, (e, \alpha, s_\alpha), (e, \beta, 1), \ldots, (e, \beta, s_\beta), (e, \gamma, 1), \ldots, (e, \gamma, s_\gamma)\}.$$

The label "α" in the name of the vertex (e, α, i) indicates that, in the instance $(J, \mathbf{S})$, this vertex will be adjacent to α. The labels "β" and "γ" are similar. The label "e" in the name of the vertex (e, α, i) indicates that this vertex is in $V'(e)$.

For each edge $e = \{u, v\} \in E(G)$ we then define a graph $J(e)$ with vertex set $V(J(e)) = V'(e) \cup \{\alpha, \beta, \gamma, \omega\}$. Note that the vertices in $V'(e)$ are distinct for each

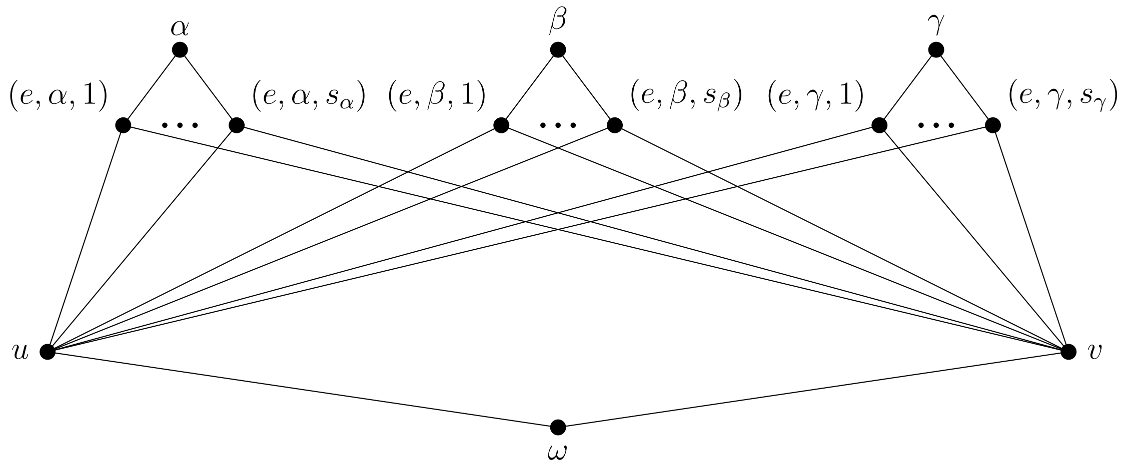


Figure 3.3: The graph $J(e)$ for $e = \{u, v\}$.

edge e whereas α, β, γ and ω are identical for all e. The edge set $E(J(e))$ of the graph $J(e)$ is defined as shown in Figure 3.3.

Then we define $J = \left(V(G) \cup \{\omega\} \cup \bigcup_{e\in E(G)} V'(e), \bigcup_{e\in E(G)} E(J(e))\right)$. Intuitively, J is constructed from the graph G by replacing each edge $e \in E(G)$ with the corresponding graph $J(e)$. Since G is connected, every vertex $v \in V(G)$ is a member of some edge in $E(G)$. This ensures that $\{v, \omega\}$ is an edge of J.

Next we define the set of lists $\mathbf{S}$ in the instance $(J, \mathbf{S})$. We set $S_\omega = \{w\}$, $S_\alpha = \{x_0\}$, $S_\beta = \{y_0\}$, $S_\gamma = \{z_0\}$ and $S_v = V(H)$ for all $v \in V(J) \setminus \{\alpha, \beta, \gamma, \omega\}$. Then we define $\mathbf{S} = \{S_v \subseteq V(H) \mid v \in V(J)\}$. This is depicted in Figure 3.4.

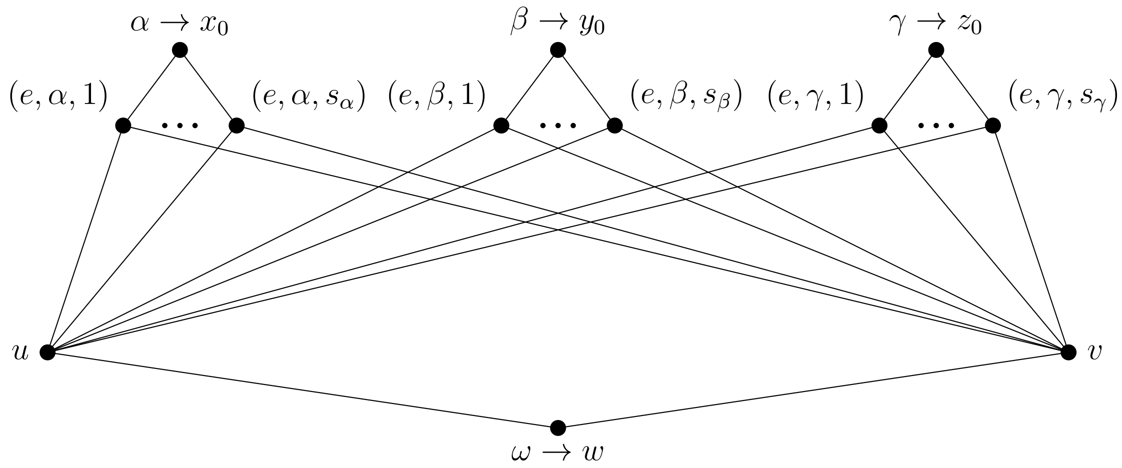


Figure 3.4: The graph $J(e)$ for $e = \{u, v\}$. A label of the form $a \to b$ means that the vertex $a \in V(G)$ is pinned to $b \in V(H)$ since $S_a = \{b\}$.

We now show how a multiterminal cut of G, α, β, γ corresponds to a certain set of homomorphisms from $(J, \mathbf{S})$ to H. Every multiterminal cut E' of G, α, β, γ induces a partition of $V(G)$ into connected components. These are the connected components of $(V(G), E(G) \setminus E')$. Let $\kappa(E')$ be the number of these components. Let $\Gamma(w)$ be the set of neighbours of vertex w in H and let $d_w \geq 3$ be the degree of w in H. Let $\Psi(E')$ be the set of functions $\psi\colon V(G) \to \Gamma(w)$ such that

- ψ maps the vertices of the components containing the terminals α, β, and γ to x_0, y_0 and z_0, respectively, and
- the set of bichromatic edges $\{\{u,v\} \in E(G) \mid \psi(u) \neq \psi(v)\}$ is exactly the cut E'.

Then

$$|\Psi(E')| = {d_w}^{\kappa(E')-3}. \tag{3.1}$$

Now, for every $\psi \in \Psi(E')$, let $X(\psi) = \{\{u,v\} \in E(G) \mid \psi(u) = \psi(v) = x_0\}$. Note that $X(\psi)$ is the set of monochromatic edges in G whose endpoints are mapped to x_0. Similarly, let $Y(\psi) = \{\{u,v\} \in E(G) \mid \psi(u) = \psi(v) = y_0\}$ and $Z(\psi) = \{\{u,v\} \in E(G) \mid \psi(u) = \psi(v) = z_0\}$.

Given a multiterminal cut E' of G, α, β, γ and a map $\psi \in \Psi(E')$, we say that a homomorphism $\sigma \in \mathcal{H}((J, \mathbf{S}), H)$ *agrees* with ψ if, for all $v \in V(G)$, we have $\sigma(v) = \psi(v)$. Let $\Sigma(\psi)$ be the set of all $\sigma \in \mathcal{H}((J, \mathbf{S}), H)$ that agree with ψ. Given a multiterminal cut E' of G, α, β, γ we say that a homomorphism $\sigma \in \mathcal{H}((J, \mathbf{S}), H)$ *agrees* with E' if there is a $\psi \in \Psi(E')$ such that σ agrees with ψ. Let $Z_{E'} = \sum_{\psi \in \Psi(E')} |\Sigma(\psi)|$ be the number of homomorphisms from $(J, \mathbf{S})$ to H that agree with the cut E'.

Now consider a multiterminal cut E' of G, α, β, γ. We will bound $Z_{E'}$ by considering two cases. Recall that B is a positive integer and part of the instance of #MultiterminalCut(3).

Case 1: $|E'| = B$.

If $\kappa(E') \geq 4$ then, since G is connected, the input G, α, β, γ has a multiterminal cut of size less than B, which contradicts the definition of #MultiterminalCut(3). Hence, it must be the case that $\kappa(E') = 3$, which means that $|\Psi(E')| = 1$. For the single $\psi \in \Psi(E')$ we have $|X(\psi)| + |Y(\psi)| + |Z(\psi)| = |E(G)| - B$. We will consider the possible homomorphisms $\sigma \in \Sigma(\psi)$.

Let $d_x, d_y, d_z \geq 2$ be the degrees of x_0, y_0 and z_0 in H, respectively.

- Consider any edge e of G that is not in the cut E'. Then, as $|E'| = B$, e has to be in $X(\psi)$, $Y(\psi)$ or $Z(\psi)$.
  - Suppose that e is in $X(\psi)$. Consider a vertex (e, β, i) of $J(e)$. This vertex is adjacent to the terminal β, which is mapped to y_0 by ψ and also to its endpoints, which are mapped to x_0 by ψ. Thus, σ has to map (e, β, i) to a mutual neighbour (in H) of x_0 and y_0. Since H is square-free, the only possibility is vertex w. Similarly, σ has to map each vertex (e, γ, i) of $J(e)$ to w. There is more choice concerning each vertex (e, α, i) of $J(e)$ — the homomorphism σ can map this vertex to any of the d_x neighbours of x_0 in H. Putting this together, the edge e contributes a factor of ${d_x}^{s_\alpha}$ to the number of homomorphisms in $\Sigma(\psi)$.

- Suppose that e is in $Y(\psi)$. Similarly, the edge e contributes a factor of ${d_y}^{s_\beta}$ to $|\Sigma(\psi)|$.
- Suppose that e is in $Z(\psi)$. Similarly, the edge e contributes a factor of ${d_z}^{s_\gamma}$ to $|\Sigma(\psi)|$.

- Consider any edge $e = \{u, v\}$ of G that is in the cut E'. Then a homomorphism $\sigma \in \Sigma(\psi)$ has to map all vertices of $V'(e)$ to a common neighbour of $\psi(u)$ and $\psi(v)$ in H. Since $\psi(u) \neq \psi(v)$ and H is square-free, the only possibility is to map all vertices of $V'(e)$ to w. Thus, the edge e contributes a factor of 1 to $|\Sigma(\psi)|$.

Putting all of this together, we have

$$Z_{E'} = {d_x}^{s_\alpha |X(\psi)|} {d_y}^{s_\beta |Y(\psi)|} {d_z}^{s_\gamma |Z(\psi)|}.$$

Now our goal is to choose s_α, s_β and s_γ so that $Z_{E'}$ depends only on the size of E' rather than on the sizes of $X(\psi)$, $Y(\psi)$ and $Z(\psi)$. (Intuitively, we want to design our graph $J(e)$ in such a way that it balances out the weights that are induced by the different degrees of x_0, y_0 and z_0.) We are limited by the fact that s_α, s_β and s_γ have to be integers. We use Lemma 3.5 to get around this. We set $\delta' = \log_q e^{\varepsilon/42}$ which we will use in the error bound of the Dirichlet approximation (the reasons behind our choice of δ' will become clear at the end of the proof). Note that $1/\delta' \in \mathrm{poly}(\varepsilon^{-1})$. Further, let $s = 2 + |E(G)| + \lceil \log_2 q \rceil |V(G)|$. We use Lemma 3.5 to approximate the real values $\lambda_1 = \log_{d_x}(2^s)$, $\lambda_2 = \log_{d_y}(2^s)$ and $\lambda_3 = \log_{d_z}(2^s)$ and obtain positive integers p_1, p_2, p_3 and r with $r \leq (n^2/\delta')^3 \in \mathrm{poly}(n, \varepsilon^{-1})$ such that for all $i \in \{1, 2, 3\}$ we have $|r\lambda_i - p_i| \leq \delta'/n^2$. Note that $p_1, p_2, p_3 \in \mathrm{poly}(n, \varepsilon^{-1})$. We set $s_\alpha = p_1$, $s_\beta = p_2$ and $s_\gamma = p_3$ to obtain

$$\begin{aligned} Z_{E'} &= {d_x}^{p_1|X(\psi)|} {d_y}^{p_2|Y(\psi)|} {d_z}^{p_3|Z(\psi)|} \\ &\leq {d_x}^{(r\lambda_1 + \delta'/n^2)|X(\psi)|} {d_y}^{(r\lambda_2 + \delta'/n^2)|Y(\psi)|} {d_z}^{(r\lambda_3 + \delta'/n^2)|Z(\psi)|} \\ &\leq 2^{sr(|X(\psi)| + |Y(\psi)| + |Z(\psi)|)} q^{\delta'/n^2(|X(\psi)| + |Y(\psi)| + |Z(\psi)|)} \end{aligned}$$

where we used the fact that $d_x, d_y, d_z \leq q$. Since $|X(\psi)| + |Y(\psi)| + |Z(\psi)| = |E(G)| - B \leq n^2$ it holds that

$$Z_{E'} \leq q^{\delta'} 2^{sr(|E(G)| - B)}.$$

Analogously we obtain $q^{-\delta'} 2^{sr(|E(G)|-B)} \leq Z_{E'}$.

Let $Z^* = 2^{sr(|E(G)|-B)}$ (this value will be used later in the proof). Then

$$q^{-\delta'} Z^* \leq Z_{E'} \leq q^{\delta'} Z^*. \tag{3.2}$$

(**End of Case 1.**)

Case 2: $|E'| > B$.

In this case we have $\kappa(E') \geq 3$. For any $\psi \in \Psi(E')$, as in Case 1, each edge in $X(\psi)$ contributes a factor of $d_x^{s_\alpha}$ to $|\Sigma(\psi)|$, each edge in $Y(\psi)$ contributes a factor of $d_y^{s_\beta}$, and each edge in $Z(\psi)$ contributes a factor of $d_z^{s_\gamma}$.

Consider any $\psi \in \Psi(E')$ and let $E'' = E(G) \setminus (X(\psi) \cup Y(\psi) \cup Z(\psi))$. Then E'' consists of edges in E' and edges in $\{\{u,v\} \in E(G) \mid \psi(u) = \psi(v) \text{ and } \psi(u) \notin \{x_0, y_0, z_0\}\}$. Any edge e in E'' contributes a factor of 1 to $|\Sigma(\psi)|$ as every vertex in $V'(e)$ has to be mapped to w. Putting all of this together and simplifying as in Case 1, we have

$$\begin{aligned} Z_{E'} &= \sum_{\psi \in \Psi(E')} {d_x}^{p_1|X(\psi)|} {d_y}^{p_2|Y(\psi)|} {d_z}^{p_3|Z(\psi)|} \\ &\leq \sum_{\psi \in \Psi(E')} q^{\delta'} 2^{sr(|X(\psi)|+|Y(\psi)|+|Z(\psi)|)}. \end{aligned}$$

We can analogously derive a lower bound for $Z_{E'}$ to obtain

$$q^{-\delta'} \sum_{\psi \in \Psi(E')} 2^{sr(|X(\psi)|+|Y(\psi)|+|Z(\psi)|)} \leq Z_{E'} \leq q^{\delta'} \sum_{\psi \in \Psi(E')} 2^{sr(|X(\psi)|+|Y(\psi)|+|Z(\psi)|)} \tag{3.3}$$

(**End of Case 2.**)

Let $\mathcal{M}$ denote the set of multiterminal cuts of G with terminals α, β and γ and let T be the number of multiterminal cuts in $\mathcal{M}$ with size B. We would like to show how to estimate T using an approximation for the number of homomorphisms from $(J, \mathbf{S})$ to H. Towards this end, define Z as follows.

$$Z = TZ^* + \sum_{E' \in \mathcal{M}:|E'|>B} \sum_{\psi \in \Psi(E')} 2^{sr(|X(\psi)|+|Y(\psi)|+|Z(\psi)|)}.$$

The proof is in two parts.

Part 1: We show that $Z/Z^* \in [T, T+1/4]$.

Since $|X(\psi)| + |Y(\psi)| + |Z(\psi)| \leq |E(G)| - |E'|$, we have

$$Z \leq TZ^* + \sum_{E' \in \mathcal{M}:|E'|>B} \sum_{\psi \in \Psi(E')} 2^{sr(|E(G)|-|E'|)}.$$

Using $|\Psi(E')| = {d_w}^{\kappa(E')-3}$ (from (3.1)) and the definition of Z^* (just before (3.2)), we obtain

$$Z \leq TZ^* + \sum_{E' \in \mathcal{M}:|E'|>B} {d_w}^{\kappa(E')-3} \frac{Z^*}{2^{sr(|E'|-B)}}.$$

Then, in the following expression, the first inequality follows from the definition of Z and the second inequality follows from the fact that there are at most $2^{|E(G)|}$ multiterminal cuts and from the bounds ${d_w}^{\kappa(E')-3} \leq q^n$, $|E'| - B \geq 1$ and $r \geq 1$. The third inequality follows from the choice of s.

$$T \leq \frac{Z}{Z^*} \leq T + \frac{2^{|E(G)|} q^n}{2^s} \leq T + 1/4.$$

We have verified that $Z/Z^* \in [T, T+1/4]$.

Part 2: We show that we can obtain a close approximation to Z/Z^* using an oracle for approximating $\#\mathbf{Ret}(H)$.

Recall that $N\big((J,\mathbf{S}) \to H\big)$ is the number of homomorphisms from $(J,\mathbf{S})$ to H and note that

$$N\big((J,\mathbf{S}) \to H\big) = \sum_{E' \in \mathcal{M}: |E'|=B} Z_{E'} + \sum_{E' \in \mathcal{M}: |E'|>B} Z_{E'}.$$

Using Inequalities (3.2) and (3.3), and the fact that G, α, β, γ has T multiterminal cuts of size B, we have

$$q^{-\delta'} Z \leq N\big((J,\mathbf{S}) \to H\big) \leq q^{\delta'} Z.$$

Let $\hat{Q}$ be a solution returned by the $\#\textsc{Ret}(H)$ oracle when called with input $\big((J,\mathbf{S}), \varepsilon/42\big)$. Then

$$e^{-\varepsilon/42} q^{-\delta'} Z \leq e^{-\varepsilon/42} N\big((J,\mathbf{S}) \to H\big) \leq \hat{Q} \leq e^{\varepsilon/42} N\big((J,\mathbf{S}) \to H\big) \leq e^{\varepsilon/42} q^{\delta'} Z.$$

The choice of $\delta' = \log_q e^{\varepsilon/42}$ yields $e^{-\varepsilon/21}\frac{Z}{Z^*} \leq \frac{\hat{Q}}{Z^*} \leq e^{\varepsilon/21}\frac{Z}{Z^*}$. Note that Z^* is easy to compute. The fact that this precision in the approximation of Z suffices to obtain the required accuracy of the output $\hat{Q}/Z^*$ as an approximation of T is derived in [37, Proof of Theorem 3]. □

We can now give a classification of $\#\textsc{Ret}(H)$ for irreflexive square-free graphs.

Theorem 3.7. *Suppose that H is an irreflexive square-free graph.*

(i) If every connected component of H is a star, then $\#\textsc{Ret}(H)$ is in FP.

(ii) Otherwise, if every connected component of H is a caterpillar, then $\#\textsc{Ret}(H)$ approximation-equivalent to #BIS.

(iii) Otherwise, $\#\textsc{Ret}(H)$ approximation-equivalent to #SAT.

Proof. We first give the classification assuming that H is a connected graph. Then we use Remark 3.3 to recover the full classification.

Suppose that H is a connected irreflexive square-free graph. We have $\#\textsc{Hom}(H) \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$ and $\#\textsc{Ret}(H) \leq_{\mathrm{AP}} \#\textsc{LHom}(H)$ by Observation 1.18. Therefore, $\#\textsc{Ret}(H)$ inherits hardness results from $\#\textsc{Hom}(H)$ hardness results and it inherits easiness results from $\#\textsc{LHom}(H)$ easiness results. Thus, since a star is a complete bipartite graph, item (i) follows from Theorem 1.10. Since a square-free graph that is not a star cannot be a complete bipartite graph, the #BIS-hardness part of item (ii) follows from Theorem 1.9. It is known that a caterpillar is a bipartite permutation graph [109] (see also [63, Appendix A]), so the #BIS-easiness part of item (ii) follows again from Theorem 1.10. Theorem 1.10 also implies that $\#\textsc{Ret}(H)$ is always #SAT-easy, giving the easiness result in item (iii).

It remains to show the hardness result in item (iii). If H is not a caterpillar, then it contains either a cycle or an induced J_3 [91, Theorem 1].

Case 1: H **contains an induced** J_3**.** The fact that #SAT $\leq_{\text{AP}}$ #RET(H) follows from Lemma 3.6. **End of Case 1.**

Case 2: H **contains a cycle.**

- Suppose that H contains a cycle of odd length. Then H is not bipartite and even the problem of deciding whether there exists a homomorphism to H is NP-complete due to Hell and Nešetřil [96]. This homomorphism decision problem reduces to the retraction decision problem RET(H) [6] and therefore RET(H) is NP-hard as well. Then, NP-hardness of RET(H) implies #SAT-hardness of the corresponding approximate counting problem #RET(H) by [37, Theorem 1].

- Suppose that H contains exactly one cycle of even length. Then H is a pseudotree and, as H is square-free, the cycle has length at least 6. Therefore RET(H) is NP-complete by Theorem 1.3. Then, as before, it follows that #RET(H) is #SAT-hard under AP-reductions by [37, Theorem 1].

- Suppose that H contains at least 2 cycles and all cycles in H have even length. We will show that H contains an induced J_3 and therefore is covered by Case 1. Let C be a shortest cycle in H. As there are at least two cycles in H and H is connected, there exists a path $P_1 = w, z_0, z_1$ such that w is in C and z_0 is not in C. As C has length at least 6, there exists a path P_2 in C of the form x_0, x_1, w, y_0, y_1. As z_0 is not in C it does not coincide with any of the vertices of P_2. Further, as C is a shortest cycle, z_1 cannot coincide with any of the vertices of P_2. Therefore the vertices of P_1 and P_2 form a graph J_3 as shown in Figure 3.2. This subgraph J_3 is induced as H does not contain any cycles of length less than 6.

End of Case 2.

The theorem now follows easily by Remark 3.3. If every connected component is easy, so is H. If any connected component is hard, so is H. □

3.2.2 Graphs with Loops

In this section we consider graphs that are not irreflexive.

3.2.2.1 #BIS-Easiness Results for Graphs with Loops

The point of this section is to prove the following lemma.

Lemma 3.8. *Let H be a partially bristled reflexive path with at least 3 vertices. Then* #RET(H) $\equiv_{\text{AP}}$ #BIS.

This lemma builds on Kelk [108, Appendix A.8], who shows #HOM(H) $\equiv_{\text{AP}}$ #BIS for partially bristled reflexive paths H. Thus, our work in this section is generalising Kelk's work from homomorphism-counting to retraction-counting. For us, the main interest is actually that we manage to classify *all* graphs of girth at least 5, rather than that we show that these particular graphs are #BIS-equivalent. Nevertheless, partially bristled reflexive paths allow us to explore some interesting ideas, providing a convenient setting for generalising useful techniques.

In particular, in order to reduce #RET(H) to #BIS, we generalise a technique that was introduced by Dyer et al. [37, Lemma 8] in order to reduce homomorphism-counting problems to #BIS. Although the graphs H that we consider in this chapter are undirected, we show that the technique also applies to directed graphs. We expect this to be useful for future work.[1] A homomorphism from a digraph G to a digraph H is simply a function $h\colon V(G) \to V(H)$ such that, for all $(u, v) \in E(G)$, the image $(h(u), h(v))$ is in $E(H)$. A homomorphism from $(G, \mathbf{S})$ to H must satisfy $h(v) \in S_v$, as in the undirected case. As for undirected graphs, we use $N\big((G, \mathbf{S}) \to H\big)$ to denote the number of homomorphisms from $(G, \mathbf{S})$ to H. Thus, we consider the following directed retraction problem.

Name: #DIR-RET(H).
Input: An irreflexive digraph G and a collection of lists $\mathbf{S} = \{S_v \subseteq V(H) \mid v \in V(G)\}$ such that, for all $v \in V(G)$, $|S_v| \in \{1, |V(H)|\}$.
Output: $N\big((G, \mathbf{S}) \to H\big)$.

The main method used in the literature to prove #BIS-easiness of approximate homomorphism-counting problems is to reduce them to the problem of counting the downsets of a partial order, which is known to be #BIS-equivalent [37]. In order to obtain more general results, we formalise the technique introduced in the proof of [37, Lemma 8] and expanded by Kelk [108], and use it in the context of the constraint satisfaction framework. Let $\mathcal{L}$ be a set of Boolean relations (called a constraint language). The counting constraint satisfaction problem (CSP) with parameter $\mathcal{L}$ is defined as follows.

Name: #CSP($\mathcal{L}$).
Input: A set of variables X and a set of constraints C, where each constraint applies a relation from $\mathcal{L}$ to a list of variables from X.
Output: The number of assignments $\sigma\colon X \to \{0, 1\}$ that satisfy all constraints in C.

The constraint language that we will use consists of the two unary Boolean relations $\delta_0 = \{(0)\}$ and $\delta_1 = \{(1)\}$ and the arity-two Boolean relation $\mathrm{Imp} = \{(0, 0), (0, 1), (1, 1)\}$. Note that the constraint $\delta_0(x)$ forces a satisfying assignment to assign the value 0 to the variable x and the constraint $\delta_1(x)$ forces a satisfying assignment to assign the value 1 to x. The constraint $\mathrm{Imp}(x, y)$ ensures that, in any satisfying assignment σ, we have $\sigma(x) \implies \sigma(y)$ (that is, if $\sigma(x) = 1$ then $\sigma(y) = 1$). It is known that the counting constraint satisfaction problem is #BIS-equivalent when the constraint language contains (exactly) these three relations.

Lemma 3.9. *[39, Theorem 3]* #CSP($\{\mathrm{Imp}, \delta_0, \delta_1\}$) $\equiv_{\mathrm{AP}}$ #BIS.

[1] The technique also applies if the input G to #RET(H) is allowed to have loops. This is the main observation needed to show that Theorem 1.11 extends to the setting where G might have loops.

We now formalise the downsets reduction technique from [37, Lemma 8] and state it as a technique for reducing homomorphism-counting problems to #CSP($\{\text{Imp}, \delta_0, \delta_1\}$). We generalise the original technique in two ways. First, we allow size-1 and size-$|V(H)|$ lists in the input, so we obtain #BIS-easiness results for #RET(H) and not merely for #HOM(H). Second, even though the main focus of this chapter is on undirected graphs, we set up the machinery to enable (stronger) #BIS-easiness results for the directed problem #DIR-RET(H).

The main idea is as follows. Given *any* instances I_v, I_e, I_f and I_b of #CSP($\{\text{Imp}\}$) on a variable set X we will define (Definition 3.10) an undirected graph $H_{I_\text{v},I_\text{e}}$ and (Definition 3.11) a digraph $H_{I_\text{v},I_\text{f},I_\text{b}}$. Then Lemma 3.12 will show that the problems #RET($H_{I_\text{v},I_\text{e}}$) and #DIR-RET($H_{I_\text{v},I_\text{f},I_\text{b}}$) both reduce to the #BIS-easy problem #CSP($\{\text{Imp}, \delta_0, \delta_1\}$). Finally, to prove the #BIS-easiness of #RET(H) when H is a partially bristled reflexive path (in order to achieve our goal of proving Lemma 3.8), we have to show, given a partially bristled reflexive path H, how to set up the corresponding instances I_v and I_e of #CSP($\{\text{Imp}\}$) so that $H_{I_\text{v},I_\text{e}} = H$.

Before defining the graph $H_{I_\text{v},I_\text{e}}$ and the digraph $H_{I_\text{v},I_\text{f},I_\text{b}}$, it helps to explain the notation. The subscript "v" stands for "vertex" and the #CSP($\{\text{Imp}\}$) instance I_v is used to define the vertices of the graph $H_{I_\text{v},I_\text{e}}$ and the vertices of the digraph $H_{I_\text{v},I_\text{f},I_\text{b}}$. The subscript "e" stands for "edge" and the CSP instance I_e is used to define the edges of $H_{I_\text{v},I_\text{e}}$. The instance I_f gives the "forward" constraints for each directed edge of $H_{I_\text{v},I_\text{f},I_\text{b}}$ and the instance I_b gives the corresponding "backward" constraints. We will use C_v, C_e, C_f, and C_b to denote the constraint sets of the instances I_v, I_e, I_f, and I_b, respectively.

Definition 3.10. Let $I_\text{v} = (X, C_\text{v})$ and $I_\text{e} = (X, C_\text{e})$ be instances of #CSP($\{\text{Imp}\}$). We define the undirected graph $H_{I_\text{v},I_\text{e}}$ as follows. The vertices of $H_{I_\text{v},I_\text{e}}$ are the satisfying assignments of I_v. Given any assignments σ and σ' in $V(H_{I_\text{v},I_\text{e}})$, there is an edge $\{\sigma, \sigma'\}$ in $H_{I_\text{v},I_\text{e}}$ if and only if the following holds: For every constraint $\text{Imp}(x, y)$ in I_e, we have $\sigma(x) \Rightarrow \sigma'(y)$ and $\sigma'(x) \Rightarrow \sigma(y)$.

The definition of the digraph $H_{I_\text{v},I_\text{f},I_\text{b}}$ is similar.

Definition 3.11. Let $I_\text{v} = (X, C_\text{v})$, $I_\text{f} = (X, C_\text{f})$ and $I_\text{b} = (X, C_\text{b})$ be instances of #CSP($\{\text{Imp}\}$). We define the directed graph $H_{I_\text{v},I_\text{f},I_\text{b}}$ as follows. The vertices of $H_{I_\text{v},I_\text{f},I_\text{b}}$ are the satisfying assignments of I_v. Given any assignments σ and σ' in $V(H_{I_\text{v},I_\text{f},I_\text{b}})$, there is a (directed) edge (σ, σ') in $H_{I_\text{v},I_\text{f},I_\text{b}}$ if and only if the following holds:

- For every constraint $\text{Imp}(x, y)$ in I_f, we have $\sigma(x) \Rightarrow \sigma'(y)$, and
- for every constraint $\text{Imp}(x, y)$ in I_b, we have $\sigma'(x) \Rightarrow \sigma(y)$.

Lemma 3.12. *Let* $I_\text{v} = (X, C_\text{v})$, $I_\text{e} = (X, C_\text{e})$, $I_\text{f} = (X, C_\text{f})$ *and* $I_\text{b} = (X, C_\text{b})$ *be instances of* #CSP($\{\text{Imp}\}$). *Then* #RET($H_{I_\text{v},I_\text{e}}$) $\leq_\text{AP}$ #CSP($\{\text{Imp}, \delta_0, \delta_1\}$) *and* #DIR-RET($H_{I_\text{v},I_\text{f},I_\text{b}}$) $\leq_\text{AP}$ #CSP($\{\text{Imp}, \delta_0, \delta_1\}$).

Proof. **Undirected case:** We first show the reduction from #RET($H_{I_\text{v},I_\text{e}}$) to #CSP($\{\text{Imp}, \delta_0, \delta_1\}$) and extend this to the directed result afterwards. The reductions we show are parsimonious. From an instance $(G, \mathbf{S})$ of $\#\textsc{Ret}(H_{I_\mathrm{v},I_\mathrm{e}})$ we create an instance I of $\#\mathrm{CSP}(\{\mathrm{Imp}, \delta_0, \delta_1\})$ as follows. The set of variables of I is $V(G) \times X$ and the set of constraints C of I is constructed as follows.

(1) For each $v \in V(G)$ and each constraint $\mathrm{Imp}(x, y) \in I_\mathrm{v}$, we add the constraint $\mathrm{Imp}((v, x), (v, y))$ to C.

(2) For each edge $\{u, v\} \in E(G)$ and each constraint $\mathrm{Imp}(x, y) \in I_\mathrm{e}$, we add the constraints $\mathrm{Imp}((u, x), (v, y))$ and $\mathrm{Imp}((v, x), (u, y))$ to C.

(3) For each $v \in V(G)$ with $|S_v| = 1$ let τ be the (only) element of S_v. If $\tau(x) = 0$ then add the constraint $\delta_0((v, x))$ to C. Otherwise, add the constraint $\delta_1((v, x))$ to C.

To complete the reduction from $\#\textsc{Ret}(H_{I_\mathrm{v},I_\mathrm{e}})$ to $\#\mathrm{CSP}(\{\mathrm{Imp}, \delta_0, \delta_1\})$, we will show that there is a bijection between homomorphisms from $(G, \mathbf{S})$ to $H_{I_\mathrm{v},I_\mathrm{e}}$ and satisfying assignments of I. This bijection ensures that the number of satisfying assignments of I is equal to $N\big((G, \mathbf{S}) \to H_{I_\mathrm{v},I_\mathrm{e}}\big)$. Hence the approximation to $N\big((G, \mathbf{S}) \to H_{I_\mathrm{v},I_\mathrm{e}}\big)$ can be achieved using a single oracle call to $\#\mathrm{CSP}(\{\mathrm{Imp}, \delta_0, \delta_1\})$ with the desired accuracy ε.

To establish the bijection, we present an (invertible) map from satisfying assignments of I to homomorphisms from $(G, \mathbf{S})$ to $H_{I_\mathrm{v},I_\mathrm{e}}$. The map is constructed as follows. Let σ be any satisfying assignment of I.

- For every vertex $v \in V(G)$, define a function $\sigma_v \colon X \to \{0, 1\}$ as follows. For all $x \in X$, let $\sigma_v(x) = \sigma((v, x))$. The constraints added to C in item (1) ensure that, since σ is a satisfying assignment of I, the assignment σ_v is a satisfying assignment of I_v. Thus, σ_v is a vertex of $H_{I_\mathrm{v},I_\mathrm{e}}$.
- Next, we will argue that the function from $V(G)$ to $V(H_{I_\mathrm{v},I_\mathrm{e}})$ that maps every vertex $v \in V(G)$ to σ_v is a homomorphism from $(G, \mathbf{S})$ to $H_{I_\mathrm{v},I_\mathrm{e}}$.
  - Consider an edge $\{u, v\}$ of G. We must show that $\{\sigma_u, \sigma_v\}$ is an edge of $H_{I_\mathrm{v},I_\mathrm{e}}$. Using Definition 3.10, this is equivalent to showing that, for every constraint $\mathrm{Imp}(x, y)$ in I_e, we have $\sigma_u(x) \Rightarrow \sigma_v(y)$ and $\sigma_v(x) \Rightarrow \sigma_u(y)$. Using the construction of σ_u and σ_v, this is equivalent to showing that, for every constraint $\mathrm{Imp}(x, y)$ in I_e, we have $\sigma(u, x) \Rightarrow \sigma(v, y)$ and $\sigma(v, x) \Rightarrow \sigma(u, y)$. This is ensured by the fact that σ is a satisfying assignment of I, so it satisfies the constraints added in item (2).
  - Consider a vertex $v \in V(G)$ with $S_v = \{\tau\}$. We must show that $\sigma_v = \tau$. This is ensured by the constraints added in item (3).

Starting from the satisfying assignment σ of I, we produced a homomorphism from $(G, \mathbf{S})$ to $H_{I_\mathrm{v},I_\mathrm{e}}$, namely the homomorphism that maps every vertex $v \in V(G)$ to σ_v. To finish the proof, we need only note that this construction is invertible — given any homomorphism from $(G, \mathbf{S})$ to $H_{I_\mathrm{v},I_\mathrm{e}}$ we can let σ_v denote the image of v under this homomorphism. Given the collection $\{\sigma_v \mid v \in V(G)\}$, we construct an assignment σ from $V(G) \times X$ to $\{0,1\}$ by inverting the above construction: For every $v \in V(G)$ and $x \in X$, let $\sigma((v,x)) = \sigma_v(x)$. We must then check that σ is satisfying.

- For each $v \in V(G)$, the assignment σ satisfies the relevant constraints added in item (1) because σ_v is a vertex of H_{I_v,I_e}, hence a satisfying assignment of I_v.
- For each $\{u,v\} \in E(G)$ and each pair of constraints $\text{Imp}((u,x),(v,y))$ and $\text{Imp}((v,x),(u,y))$ added to C in item (2), σ satisfies the constraints because $\{\sigma_u, \sigma_v\}$ is an edge of H_{I_v,I_e} (so $\sigma_u(x) \implies \sigma_v(y)$ and $\sigma_v(x) \implies \sigma_u(y)$).
- Finally, for any $s \in \{0,1\}$, consider a constraint $\delta_s((v,x))$ introduced in item (3). The procedure in item (3) ensures that, for some τ with $S_v = \{\tau\}$, we have $\tau(x) = s$. Our homomorphism has $\sigma_v = \tau$. Thus, the constraint $\sigma((v,x)) = s$ is satisfied by σ.

Directed Case: The reduction from #DIR-RET(H_{I_v,I_f,I_b}) to #CSP($\{\text{Imp}, \delta_0, \delta_1\}$) is similar to the one given in the undirected case. Starting with an instance $(G, \mathbf{S})$ of #DIR-RET(H_{I_v,I_f,I_b}) we create an instance I of #CSP($\{\text{Imp}, \delta_0, \delta_1\}$) as follows. The set of variables of I is $V(G) \times X$, as in the undirected reduction. The set of constraints C of I is constructed in the same way as in the undirected reduction, except that item (2) is replaced with the following.

(2)' For each (directed) edge $(u,v) \in E(G)$, we add the following constraints to C. For each constraint $\text{Imp}(x,y) \in I_f$, we add the constraint $\text{Imp}((u,x),(v,y))$ to C. For each constraint $\text{Imp}(x,y) \in I_b$, we add the constraint $\text{Imp}((v,x),(u,y))$ to C.

As in the undirected case, we complete the proof by establishing a bijection from satisfying assignments of I to homomorphisms from $(G, \mathbf{S})$ to H_{I_v,I_f,I_b}. Let σ be any satisfying assignment of I. The construction of σ_v from σ is the same as in the undirected case. Only one difference arises in the verification that the function from $V(G)$ to $V(H_{I_v,I_f,I_b})$ that maps every vertex $v \in V(G)$ to σ_v is a homomorphism from $(G, \mathbf{S})$ to H_{I_v,I_f,I_b}. Consider any directed edge (u,v) of G. We must show that (σ_u, σ_v) is an edge of H_{I_v,I_f,I_b}. Using Definition 3.11 and the construction of σ_u and σ_v, this is equivalent to showing

- For every constraint $\text{Imp}(x,y)$ in I_f, we have $\sigma(u,x) \Rightarrow \sigma(v,y)$, and
- for every constraint $\text{Imp}(x,y)$ in I_b, we have $\sigma(v,x) \Rightarrow \sigma(u,y)$.

This is ensured by the constraints added in item (2)'.

As in the undirected case, we next show that we have a bijection by starting with a homomorphism from $(G, \mathbf{S})$ to H_{I_v,I_f,I_b} and letting σ_v denote the image of v under this homomorphism. Given the collection $\{\sigma_v \mid v \in V(G)\}$ we construct an assignment σ from $V(G) \times X$ to $\{0,1\}$ exactly as in the undirected case. We must check that σ is a satisfying assignment of I. This is the same as the undirected case except when checking that σ satisfies the constraints added in item (2)'. For $(u,v) \in E(G)$ and a constraint $\mathrm{Imp}((u,x),(v,y))$ added to C because $\mathrm{Imp}(x,y) \in I_{\mathrm{f}}$, note that, since (σ_u, σ_v) is an edge of $H_{I_{\mathrm{v}},I_{\mathrm{f}},I_{\mathrm{b}}}$, by Definition 3.11, we have $\sigma_u(x) \implies \sigma_v(x)$, so the constraint is satisfied. Similarly, for a constraint $\mathrm{Imp}((v,x),(u,y))$ added to C because $\mathrm{Imp}(x,y) \in I_{\mathrm{b}}$ we again have $\sigma_v(x) \implies \sigma_u(y)$, so the constraint is satisfied.

So we have a bijection from satisfying assignments of I to homomorphisms from $(G, \mathbf{S})$ to $H_{I_{\mathrm{v}},I_{\mathrm{f}},I_{\mathrm{b}}}$ and the reduction from #Dir-Ret($H_{I_{\mathrm{v}},I_{\mathrm{f}},I_{\mathrm{b}}}$) to #CSP($\{\mathrm{Imp}, \delta_0, \delta_1\}$) follows. □

Although, to be general, we have presented undirected and directed reductions in Lemma 3.12, our goal in Lemma 3.8 is to prove #BIS-easiness of #Ret(H) for a partially bristled reflexive path, which is an undirected graph. So we will use the undirected reduction from Lemma 3.12 for this. The rough idea will be to take a partially bristled reflexive path H and show how to set up the corresponding instances I_{v} and I_{e} of #CSP({Imp}) so that $H_{I_{\mathrm{v}},I_{\mathrm{e}}} = H$. Then Lemma 3.12 shows that #Ret(H) reduces to #CSP($\{\mathrm{Imp}, \delta_0, \delta_1\}$), so #Ret($H$) is #BIS-easy. Unfortunately, we can't precisely achieve this goal, but we can set up the corresponding instances I_{v} and I_{e} so that $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$ is equal to H, together with some additional small connected components, which turn out not to matter. The fact that these small connected components don't cause trouble was first observed by Kelk [108] in the context of counting homomorphisms. In Lemma 3.14 we show that this is also true when counting retractions. Lemma 3.13 states the well-known fact that subtracting polynomial-size entities does not spoil an AP-reduction, which is, for instance, pointed out in [108, Lemma 6.6]. For the sake of completeness we give a short proof.

Lemma 3.13. *Let H and H' be graphs. For any graph G, let $f(G) = N(G \to H) - N(G \to H')$. If $f(G)$ is non-negative and bounded from above by a polynomial in $|V(G)|$, and can be computed in polynomial time, then #Hom(H') $\leq_{\mathrm{AP}}$ #Hom(H).*

Proof. Let G be an instance of #Hom(H') and let $\varepsilon \in (0,1)$ be the desired precision. To shorten notation, let $N = N(G \to H)$. From the definition of f in the statement of the lemma, $N(G \to H') = N - f(G)$. First, the algorithm computes $k = f(G)$ in polynomial time. If $k = 0$, then $N(G \to H') = N(G \to H)$, and the algorithm simply returns the result of a #Hom(H) oracle call with precision ε.

Suppose instead that $k \geq 1$. In this case, the algorithm makes a #Hom(H) oracle call with input G and precision $\delta \leq \frac{\varepsilon}{16k}$. Let R be the integer solution returned by this oracle call (note that R is an approximation to N satisfying $e^{-\delta}N \leq R \leq e^{\delta}N$). The algorithm returns $R - k$. We show that this output approximates $N(G \to H')$ with the desired precision.

If $N(G \to H') = 0$ then $N = k$ and $e^{-\delta}k \leq R \leq e^{\delta}k$. By Observation 3.4 and the facts that $\varepsilon < 1$ and $k \geq 1$ this implies $R \in (k - 1/4, k + 1/4)$ and since R is integer this gives $R = k$. Thus, in this case the algorithm returns 0, which is the exact solution.

Suppose instead that $N(G \to H') \geq 1$. In this case, $N \geq k + 1$ and by Observation 3.4 we have

$$R - k \leq e^{\delta}N - k \leq (1 + 2\delta)N - k = (1 + 2\delta)(N - k) + 2k\delta.$$

Since $N \geq k+1$ and $2\delta \leq \varepsilon/8$ we have $2k\delta \leq \varepsilon/8 \leq \varepsilon/8 \cdot (N-k)$ and consequently

$$(1+2\delta)(N-k) + 2k\delta \leq (1+\varepsilon/4)(N-k).$$

Analogously, we obtain $R - k \geq (1-\delta)(N-k) - k\delta \geq (1-\varepsilon/8)(N-k)$. Finally, by Observation 3.4, this implies $e^{-\varepsilon}(N-k) \leq R-k \leq e^{\varepsilon}(N-k)$ and thus returning $R-k$ has the desired precision. □

Lemma 3.14. *Let H' be a graph and let H be the graph consisting of a connected component that is isomorphic to H' together with some additional connected components $C_1, \ldots, C_k$. Suppose that, for each $i \in [k]$, C_i is one of the following graphs: a singleton vertex, with or without a loop, or an unlooped edge. Then $\#\text{RET}(H') \leq_{\text{AP}} \#\text{RET}(H)$.*

Proof. Recall the definition of $\#\text{RET}^{\text{C}}(H)$ from Section 3.1.2. To prove the lemma we show

$$\#\text{RET}(H') \leq_{\text{AP}} \#\text{RET}^{\text{C}}(H') \leq_{\text{AP}} \#\text{RET}^{\text{C}}(H) \leq_{\text{AP}} \#\text{RET}(H). \tag{3.4}$$

The first and the trivial third reduction follow from Observation 3.2. It remains to show that $\#\text{RET}^{\text{C}}(H') \leq_{\text{AP}} \#\text{RET}^{\text{C}}(H)$. Let $(G, \mathbf{S}')$ be an input to $\#\text{RET}^{\text{C}}(H')$ and let $\varepsilon \in (0,1)$ be the desired precision. From the problem definition, G is connected. Now define the lists S_v for $v \in V(G)$ as follows. If $S'_v = V(H')$ then let $S_v = V(H)$. Otherwise, let $S_v = S'_v$. Let $\mathbf{S} = \{S_v \mid v \in V(G)\}$

First, the algorithm tests whether there is a list $S'_v \in \mathbf{S}'$ with $|S'_v| = 1$. If there is such a list, then there is a particular component of H' with the property that every homomorphism from G to H' maps all vertices of G to this component, and every homomorphism from G to H maps all vertices of G to this component. Thus, $N\big((G,\mathbf{S}') \to H'\big) = N\big((G,\mathbf{S}) \to H\big)$. So a single oracle call with precision ε gives the sought-for approximation.

If there is no list $S'_v \in \mathbf{S}'$ with $|S'_v| = 1$ then every list S'_v is equal to $V(H')$ and every list S_v is equal to $V(H)$. Thus, $N\big((G,\mathbf{S}') \to H'\big) = N\big(G \to H'\big)$ and $N\big((G,\mathbf{S}) \to H\big) = N\big(G \to H\big)$. As G is connected we also have $N\big(G \to H\big) = N\big(G \to H'\big) + \sum_{i=1}^{k} N\big(G \to C_i\big)$. As $C_1, \ldots C_k$ are either singleton vertices or unlooped edges, the algorithm can compute $\sum_{i=1}^{k} N\big(G \to C_i\big)$ efficiently. Also, for each $i \in [k]$, $N\big(G \to C_i\big) \leq 2$. Setting $f(G) = \sum_{i=1}^{k} N\big(G \to C_i\big)$ in Lemma 3.13 gives the sought-for AP-reduction. □

As noted at the beginning of this section, approximately counting homomorphisms to partially bristled reflexive paths is shown to be #BIS-easy in [108, Appendix A.8]. Using the same construction and our Lemma 3.12 we can now prove Lemma 3.8, which is the generalisation for counting retractions. We restate the lemma and recast the construction in our setting for convenience.

Lemma 3.8. *Let H be a partially bristled reflexive path with at least 3 vertices. Then $\#\text{RET}(H) \equiv_{\text{AP}} \#\text{BIS}$.*

Proof. The #BIS-hardness part of the statement is inherited from #Hom(H) using Theorem 1.9 and the reduction #Hom(H) $\leq_{\mathrm{AP}}$ #Ret(H) from Observation 1.18. We now show #BIS-easiness.

Matching the notation from Definition 3.1, the partially bristled reflexive path H can be described as follows. There exists a positive integer Q and a be a subset S of $[Q]$ such that $V(H) = \{c_0, \dots, c_{Q+1}\} \cup \bigcup_{i\in S}\{g_i\}$ and $E(H) = \bigcup_{i=0}^{Q}\{c_i, c_{i+1}\} \cup \bigcup_{i=0}^{Q+1}\{c_i, c_i\} \cup \bigcup_{i\in S}\{c_i, g_i\}$. Note that S can be empty.

Let $X = \{x_0, \dots, x_Q\}$. Define the instances $I_{\mathrm{v}} = (X, C_{\mathrm{v}})$ and $I_{\mathrm{e}} = (X, C_{\mathrm{e}})$ of #CSP({Imp}) as follows.

- For each $i \in [Q] \setminus S$, we add a constraint $\mathrm{Imp}(x_i, x_{i-1})$ to C_{v}.
- For each pair (i, j) satisfying $0 \leq i < j \leq Q$, we add a constraint $\mathrm{Imp}(x_j, x_i)$ to C_{e}.

We claim that the graph $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$, as defined in Definition 3.10, has a connected component that is isomorphic to H and that all other connected components of $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$ are singleton vertices (without loops). Given the claim, the reduction from #Ret(H) to #BIS follows from Lemmas 3.14, 3.12 and 3.9 (applied in that order).

We conclude the proof by showing the claim. For each $i \in \{0, \dots, Q+1\}$ let $\sigma_i \colon X \to \{0, 1\}$ be the following assignment of Boolean values to variables in X.

$$\sigma_i(x_j) = \begin{cases} 1, & \text{if } j < i \\ 0, & \text{otherwise.} \end{cases}$$

Note that σ_0 maps all arguments to 0 and σ_{Q+1} maps all arguments to 1.

The indices of $\sigma_0, \dots, \sigma_{Q+1}$ are chosen this way to match the indices of the vertices c_0 to c_{Q+1} of the graph H. Note that $\sigma_0, \dots, \sigma_{Q+1}$ are satisfying assignments of I_{v} and, therefore, they are vertices of $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$. By the definition of I_{e}, these vertices are looped in $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$. Also, for all $i \in [Q]$, we have $\{\sigma_i, \sigma_{i+1}\} \in E(H_{I_{\mathrm{v}},I_{\mathrm{e}}})$. Hence, the vertices $\sigma_0, \dots, \sigma_{Q+1}$ form a reflexive path in $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$.

Now for each $i \in [Q]$ let $\sigma_i' \colon X \to \{0, 1\}$ be the following assignment of Boolean values to variables in X.

$$\sigma_i'(x_j) = \begin{cases} 1, & \text{if } j \leq i \text{ and } j \neq i-1 \\ 0, & \text{otherwise.} \end{cases}$$

For every $i \in [Q]$, we have $\sigma_i'(x_{i-1}) = 0$ and $\sigma_i'(x_i) = 1$, so σ_i' is not equal to any $\sigma_{i'}$.

Consider a vertex c_i of H with $i \in [Q]$.

- If $i \in S$: In this case, σ_i' is a satisfying assignment of I_{v}. By the definition of I_{e}, σ_i' has degree 1 and is adjacent to σ_i in $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$. Thus, the vertex σ_i' of $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$ corresponds to the vertex g_i of H.
- If $i \notin S$: In this case, as $i \geq 1$, the constraint $\mathrm{Imp}(x_i, x_{i-1})$ in I_{v} ensures that σ_i' is not a satisfying assignment of I_{v} and, therefore, σ_i' is not a vertex of $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$.

We will next show that the edges that we have already described constitute all of the edges of H_{I_v,I_e}. This means that the rest of the vertices of H_{I_v,I_e} have degree 0, so we are finished.

To this end, let σ be any function from X to $\{0,1\}$. From the definition of I_e, we obtain the following necessary condition for σ to have a neighbour in H_{I_v,I_e}: Let $i \in \{0,\dots,Q\}$ be the largest index for which $\sigma(x_i) = 1$. If ψ is a neighbour of σ then, for all $j \le i-1$, $\psi(x_j) = 1$ and hence, for all $j \le i-2$, $\sigma(x_j) = 1$. Thus, for σ to have a neighbour in H_{I_v,I_e} it has to be of the form σ_i or σ'_i. □

Remark 3.15. One interesting feature of Lemma 3.8 is that it shows that there are graphs H for which #RET(H) is #BIS-equivalent, whereas #LHOM(H) is #SAT-hard. Thus, subject to the complexity assumption that #BIS is not #SAT-equivalent, there is a graph H for which the complexity of #RET(H) differs from that of #LHOM(H). The smallest example from the class of partially bristled reflexive paths for which this separation holds is the so-called 2-Wrench, depicted in Figure 3.5. The fact that #SAT $\le_{\mathrm{AP}}$ #LHOM(2-Wrench) follows from Theorem 1.10.

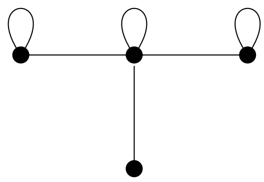


Figure 3.5: The graph 2-Wrench.

3.2.2.2 #SAT-Hardness Results for Graphs with Loops

The goal of this section is to prove the hardness results given in Lemmas 3.18, 3.19 and 3.20. In order to show #SAT-hardness results we will prove that certain neighbourhood structures induce hardness. To this end consider the following easy and well-known observation proved here for completeness.

Observation 3.16. *Let H be a graph and let u be a vertex of H. Then* #HOM($H[\Gamma(u)]$) $\le_{\mathrm{AP}}$ #RET(H).

Proof. Let G be an input to #HOM($H[\Gamma(u)]$) and let $v_1,\dots,v_n$ be the vertices of G. Let w be a vertex distinct from the vertices in G. Then we construct the graph G' with vertices $V(G') = V(G) \cup \{w\}$ and edges $E(G') = E(G) \cup \{\{w, v_i\} \mid i \in [n]\}$. We set $S_w = \{u\}$ and $S_v = V(H)$ for all remaining vertices of G'. Let $\mathbf{S} = \{S_v \mid v \in V(G')\}$. Then $N\big(G \to H[\Gamma(u)]\big) = N\big((G', \mathbf{S}) \to H\big)$. □

First we combine some known results to show hardness that is derived from the analysis of distance-1 neighbourhoods (Lemmas 3.18 and 3.19). Then we show hardness results derived from the analysis of distance-2 neighbourhoods in the more difficult Lemma 3.20, which is the main result of this section.

For Lemmas 3.18 and 3.19 we use gadgets based on complete bipartite graphs where two states dominate (see, e.g., [37, Lemma 25], [78, Section 5] and [108, Lemma 5.1]). We use the version of Kelk [108]. Let $F(H) = \{u \in V(H) \mid \Gamma(u) = V(H)\}$. For a set of vertices S recall the set of common neighbours $\Gamma(S)$ from Section 3.1.2.

Lemma 3.17 ([108, Lemma 5.1])**.** *Let H be a graph with $\emptyset \subsetneq F(H) \subsetneq V(H)$. Suppose that, for every pair (S,T) with $\emptyset \subseteq S,T \subseteq V(H)$ satisfying $S \subseteq \Gamma(T)$ and $T \subseteq \Gamma(S)$, at least one of the following holds:*

(1) $S = F(H)$.

(2) $T = F(H)$.

(3) $|S| \cdot |T| < |F(H)| \cdot |V(H)|$.

Then $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Hom}(H)$.

Lemma 3.17 is not difficult to prove. A homomorphism from a complete bipartite graph to H will typically map one side to $F(H)$ and the other to $V(H)$. So it is easy to reduce from counting independent sets.

Let WR_q be a reflexive star with q leaves. (The name is not relevant here but it comes from the Widom-Rowlinson model [150] from statistical physics.) The non-leaf vertex of WR_q is called its centre.

Lemma 3.18. *Let H be a graph that has a looped vertex b such that $H[\Gamma(b)]$ is isomorphic to* WR_q *for some* $q \geq 3$. *Then* $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$.

Proof. The problem $\#\textsc{Hom}(\mathrm{WR}_q)$ is the same as $\#\textsc{Hom}(H[\Gamma(b)])$, and by Observation 3.16 we obtain $\#\textsc{Hom}(H[\Gamma(b)]) \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$. For $q \geq 4$ Dyer et al. [37, Lemma 26] show $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Hom}(\mathrm{WR}_q)$. For $q = 3$ this fact is due to Kelk [108, Section 2.3]. Summarising we obtain

$$\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Hom}(\mathrm{WR}_q) \equiv_{\mathrm{AP}} \#\textsc{Hom}(H[\Gamma(b)]) \leq_{\mathrm{AP}} \#\textsc{Ret}(H).$$

□

Recall the 2-Wrench as given in Figure 3.5.

Lemma 3.19. *Let H be a triangle-free graph that has a looped vertex b which has an unlooped neighbour. If $H[\Gamma(b)]$ is not isomorphic to a 2-Wrench, then* $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$.

Proof. By Observation 3.16 we know $\#\textsc{Hom}(H[\Gamma(b)]) \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$. We show $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Hom}(H[\Gamma(b)])$ to obtain $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$.

To shorten the notation let $H_b = H[\Gamma(b)]$. We consider different cases depending on the graph H_b. By assumption the vertex b is looped and has at least one unlooped neighbour. First consider the case where H_b has at most 4 vertices. Since, by assumption, H_b is triangle-free and not isomorphic to a 2-Wrench, it has to be isomorphic to one of the graphs depicted in Figure 3.6. Approximately counting homomorphisms to the first graph in Figure 3.6 is well-known to be equivalent to #IS (the problem of approximately counting independent sets in a graph) which is #SAT-equivalent [37, Theorem 3]. The second and fourth graphs correspond to weighted versions of #IS which are known to be #SAT-equivalent [108, Lemma 2.3]. The third graph is the so-called 1-Wrench and the corresponding #SAT-hardness is shown in [37, Theorem 21]. Finally, approximately counting homomorphisms to the fifth graph in Figure 3.6 is shown to be #SAT-hard in [108, Section 2.3].

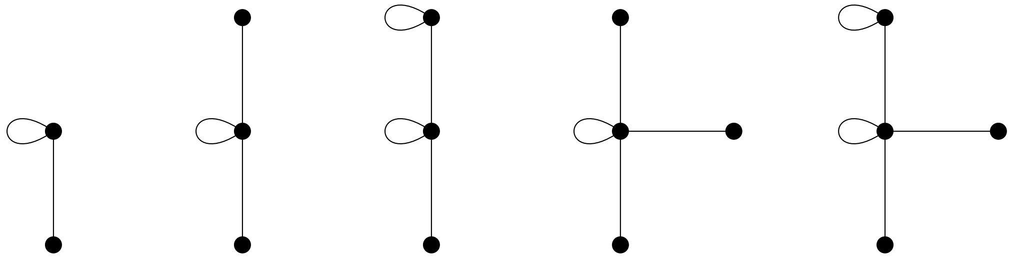


Figure 3.6: Possible graphs H_b with at most 4 vertices.

Now consider the case where H_b has 5 or more vertices. We claim that, under this assumption, Lemma 3.17 gives $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Hom}(H_b)$. To see this, note that $F(H_b) = \{b\}$ and $|F(H_b)||V(H_b)| \geq 5$. Consider any pair (S, T) with $\emptyset \subseteq S, T \subseteq V(H_b)$, $S \subseteq \Gamma(T)$ and $T \subseteq \Gamma(S)$. We distinguish between different cases depending on the cardinalities of S and T and show that in each case the conditions of Lemma 3.17 are fulfilled.

- If $|S| = 1$ then either item (1) or item (3) of Lemma 3.17 are satisfied.
- If $|T| = 1$ then either item (2) or item (3) of Lemma 3.17 are satisfied.
- If $|S| \geq 3$ then $T = \{b\}$ since $T \subseteq \Gamma(S)$ and H is triangle-free. So $|T| = 1$.
- If $|T| \geq 3$ then $S = \{b\}$ since $S \subseteq \Gamma(T)$ and H is a triangle-free. So $|S| = 1$.
- If $|S| = |T| = 2$ then $|S| \cdot |T| = 4$ and item (3) of Lemma 3.17 is satisfied.

So Lemma 3.17 gives $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Hom}(H_b)$. □

Lemma 3.20. *Let H be graph that has a looped vertex b such that, for some positive integer k, H'_k (see Figure 3.8) is a subgraph of $H[\Gamma^2(b)]$, and $H[\Gamma^2(b)]$ in turn is a subgraph of H_k (see Figure 3.7). Then $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$.*

Lemma 3.20 is the final piece to show the classification for graphs of girth at least 5, as stated in Theorem 1.11. Here, we omit our proof of Lemma 3.20 from [58] because Lemma 3.20 is subsumed by the more general Lemma 4.33, which is presented in Chapter 4 — and the proof of Lemma 4.33 is much shorter than the long and technical proof of Lemma 3.20. For completeness, the proof of Lemma 3.20 from [58], which does not rely on definitions from Chapter 4, is presented in Appendix B.

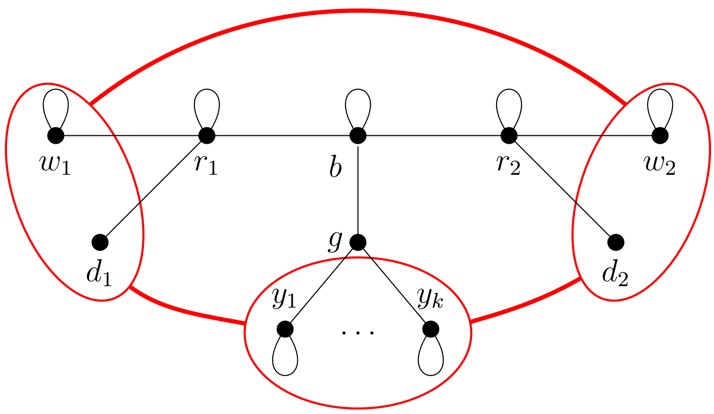


Figure 3.7: The graph H_k. Circled sets of vertices are independent sets of possibly looped vertices. Sets of vertices that are connected by a thick red edge have a complete set of edges between them.

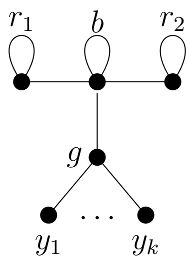


Figure 3.8: The graph H'_k.

3.2.3 Putting the Pieces together

Now finally we have all the tools at hand to prove the main classification result for counting retractions to graphs of girth at least 5, which we restate at this point.

Theorem 1.11. *Let H be a graph of girth at least 5.*

- *i) If every connected component of H is an irreflexive star or a reflexive clique of size at most 2, then #RET(H) is in FP.*
- *ii) Otherwise, if every connected component of H is an irreflexive caterpillar or a partially bristled reflexive path, then #RET(H) is approximation-equivalent to #BIS.*
- *iii) Otherwise, #RET(H) is approximation-equivalent to #SAT.*

Proof. As in the proof of Theorem 3.7, the fact that the classification extends from connected graphs to graphs with multiple connected components follows from Remark 3.3. Now assume without loss of generality that H is a connected graph. If H is an irreflexive graph then the statement of the theorem follows from the slightly more general Theorem 3.7 (in the irreflexive case we only require H to be square-free).

Now suppose that H has at least one looped vertex. From Observation 1.18 we know that, in general, hardness results for #HOM(H) carry over to #RET(H) and easiness results carry over from #LHOM(H). Then, by Theorem 1.10, #RET(H) is in FP if H is a reflexive clique of size at most 2. Otherwise, since it is triangle-free, H cannot be a complete reflexive graph, so #RET(H) is #BIS-hard with respect to AP-reductions by Theorem 1.9. The #BIS-easiness for partially bristled reflexive paths follows from our Lemma 3.8. Theorem 1.10 implies that #RET(H) is always #SAT-easy.

It remains to show the #SAT-hardness result for graphs H that have at least one looped vertex but are not partially bristled reflexive paths. To this end we distinguish two disjoint cases:

1. Suppose that every unlooped vertex in H has degree 1. Let H^* be the subgraph induced by the looped vertices of H. As all unlooped vertices have degree 1, the fact that H is connected implies that H^* is connected. Recall that WR_q is a reflexive star with q leaves. Then, in general, H^* is either a reflexive path, a reflexive cycle or it contains a subgraph WR_q for some $q \geq 3$.

   (a) Suppose that H^* is a reflexive path $u_1, \ldots, u_t$. By the fact that H is not a partially bristled reflexive path and all unlooped vertices have degree 1, it follows that either some u_i has more than one unlooped neighbour or at least one of the endpoints u_1 or u_t has an unlooped neighbour. Then #SAT-hardness follows from Lemma 3.19.

   (b) If H^* is a reflexive cycle, then by the fact that every unlooped vertex in H has degree 1, it holds that H^* is the only cycle in H. Then H is a pseudotree and, as H has girth at least 5, the reflexive cycle H^* has length at least 5. Therefore RET(H) is NP-complete by Theorem 1.3 and it follows that #RET(H) is #SAT-hard under AP-reductions by [37, Theorem 1].

   (c) If H^* contains a subgraph WR_q for some $q \geq 3$, then H contains a looped vertex with at least 3 looped neighbours apart from itself. As H is triangle-free, the subgraph WR_q is induced and #SAT-hardness follows either from Lemma 3.18 or Lemma 3.19.

2. Suppose there exists an unlooped vertex in H that has degree at least 2. As H is connected and contains at least one looped vertex, it follows that there exists a looped vertex b with an unlooped neighbour g, which has degree $k+1$ for some $k \geq 1$, i.e. has neighbours $y_1, \ldots, y_k$ apart from b. Then $H[\Gamma(b)]$ is isomorphic to a 2-Wrench, or otherwise hardness follows from Lemma 3.19. Therefore b has exactly 2 looped neighbours apart from itself. Let us call them r_1 and r_2. Then, as H has girth at least 5, the vertices $\{r_1, r_2, b, g, y_1, \ldots, y_k\}$ are distinct. This shows that $V(H'_k) \subseteq V(H[\Gamma^2(b)])$ and $E(H'_k) \subseteq E(H[\Gamma^2(b)])$, i.e. that H'_k (see Figure 3.8) is a subgraph of $H[\Gamma^2(b)]$.

   For $i = 1, 2$ the following hold:

   (a) Apart from b and r_i itself, the vertex r_i has at most 1 other looped neighbour, or otherwise hardness follows either from Lemma 3.18 or from Lemma 3.19.

(b) If r_i has an unlooped neighbour, then $H[\Gamma(r_i)]$ is isomorphic to a 2-Wrench, or otherwise hardness follows from Lemma 3.19.

From items 2a and 2b it follows that for $i = 1, 2$ the vertex r_i has at most one looped and one unlooped neighbour apart from b and r_i itself. (If they exist let us call the looped neighbour w_i and the unlooped neighbour d_i.) Therefore, $V(H[\Gamma^2(b)]) \subseteq \{w_1, d_1, r_1, w_2, d_2, r_2, b, g, y_1, \ldots, y_k\} \subseteq V(H_k)$.

Note that d_1, d_2 and g are unlooped vertices in H. Furthermore, for $i = 1, 2$ we have shown the following

- $E(H'_k) \subseteq E(H[\Gamma^2(b)])$.
- $\{w_i, r_i\} \in E(H[\Gamma^2(b)])$ if $w_i \in V(H[\Gamma^2(b)])$.
- $\{d_i, r_i\} \in E(H[\Gamma^2(b)])$ if $d_i \in V(H[\Gamma^2(b)])$.

The edges $E(H'_k)$ together with $\{w_1, r_1\}$, $\{w_2, r_2\}$, $\{d_1, r_1\}$, $\{d_2, r_2\}$ (if these exist) form a tree on the vertices $\Gamma^2(b)$ (Recall that a tree might have loops but no cycles). By the fact that H has girth at least 5, all named vertices are distinct, and it follows that $E(H[\Gamma^2(b)]) \subseteq E(H_k)$, which shows that $H[\Gamma^2(b)]$ is a subgraph of H_k.

Summarising, H'_k is a subgraph of $H[\Gamma^2(b)]$ and $H[\Gamma^2(b)]$ is a subgraph of H_k and we can apply Lemma 3.20 to obtain #SAT-hardness.

Items 1 and 2 cover all graphs H that have at least one looped vertex but are not partially bristled reflexive paths. (Note that item 1 includes the case where H is reflexive.) □

3.3 Approximately Counting Retractions is at least as hard as Counting Surjective Homomorphisms or Compactions

This section studies the place of the problem #Ret(H) within the landscape of a number of closely related counting problems.

As in Chapter 2, we use $N^{\text{sur}}(G \to H)$ to denote the number of surjective homomorphisms from G to H, and we use $N^{\text{comp}}(G \to H)$ to denote the number of compactions from G to H. We recall the formal definitions of the corresponding counting problems.

Name: #SHom(H).
Input: An irreflexive graph G.
Output: $N^{\text{sur}}(G \to H)$.

Name: #Comp(H).
Input: An irreflexive graph G.
Output: $N^{\text{comp}}(G \to H)$.

We also define the corresponding list versions of these two problems. We use $N^{\text{sur}}((G, \mathbf{S}) \to H)$ and $N^{\text{comp}}((G, \mathbf{S}) \to H)$ to denote the number of surjective homomorphisms from $(G, \mathbf{S})$ to H and the number of compactions from $(G, \mathbf{S})$ to H, respectively. Note that the list version of the problem #RET(H) is simply the problem #LHOM(H).

Name: #LSHOM(H).
Input: An irreflexive graph G and a collection of lists $\mathbf{S} = \{S_v \subseteq V(H) \mid v \in V(G)\}$.
Output: $N^{\text{sur}}\big((G, \mathbf{S}) \to H\big)$.

Name: #LCOMP(H).
Input: An irreflexive graph G and a collection of lists $\mathbf{S} = \{S_v \subseteq V(H) \mid v \in V(G)\}$.
Output: $N^{\text{comp}}\big((G, \mathbf{S}) \to H\big)$.

Furthermore, we use a generalisation of the problems #HOM(H), #RET(H) and #LHOM(H). Let $\mathcal{P}(V(H)) = \{S \mid S \subseteq V(H)\}$ be the power set of $V(H)$. For a fixed graph H and a set $\mathcal{L} \subseteq \mathcal{P}(V(H))$ we define

Name: #HOM($H, \mathcal{L}$).
Input: An irreflexive graph G and a collection of lists $\mathbf{S} = \{S_v \in \mathcal{L} \mid v \in V(G)\}$.
Output: $N\big((G, \mathbf{S}) \to H\big)$.

As a measure of distance between two distributions π and π' on a finite universe Ω we use the *total variation distance* $\mathrm{d}_{\mathrm{TV}}(\pi, \pi') = \frac{1}{2}\sum_{\omega\in\Omega}|\pi(\omega) - \pi'(\omega)|$. For a set $A \subseteq \Omega$, $\mathrm{Uni}(A)$ is the uniform distribution on A. Furthermore, $\mathrm{Be}(p)$ is the Bernoulli distribution with parameter p. In general, we write $X \sim D$ if a random variable X has distribution D.

3.3.1 Reductions using a Monte Carlo Approach

The main goal of this section is to prove Corollaries 3.24 and 3.26. Together they constitute Theorem 1.19 which states that both #SHOM(H) and #COMP(H) are AP-reducible to #RET(H).

In the following two lemmas we prove some necessary ingredients that we use in the proof of Lemma 3.23. From Section 3.1.2 recall that a RAS for #HOM($H, \mathcal{L}$) is an (ε, δ)-approximation for #HOM($H, \mathcal{L}$) with $\delta = 1/4$. First, we point out the well-known fact that this can be powered to obtain an (ε, δ)-approximation for smaller δ.

Lemma 3.21. *Let H be a graph and $\mathcal{L} \subseteq \mathcal{P}(V(H))$. There is an algorithm* COUNTHOM$_{H,\mathcal{L}}$ *which uses oracle access to a RAS for* #HOM($H, \mathcal{L}$) *and has the following properties.*

- *It is given an input $(G, \mathbf{S})$ to* #HOM($H, \mathcal{L}$) *together with accuracy parameters ε and δ in $(0, 1)$.*
- *It returns a natural number X with* $\Pr\left(e^{-\varepsilon} \le \frac{X}{N\left((G,\mathbf{S})\to H\right)} \le e^{\varepsilon}\right) \ge 1 - \delta$.
- *Its running time is bounded by a polynomial in ε^{-1}, $\log \delta^{-1}$, and the number of vertices of G.*

Proof. The lemma is basically the same as [105, Lemma 6.1] applied to the problem $\#\text{Hom}(H, \mathcal{L})$. The only difference is that [105, Lemma 6.1] gives a precision guarantee of the form

$$\Pr\left((1-\varepsilon) \leq \frac{X}{N\big((G,\mathbf{S}) \to H\big)} \leq (1+\varepsilon)\right) \geq 1-\delta.$$

However, by Observation 3.4, using accuracy parameter $\varepsilon/2$ instead of ε in [105, Lemma 6.1] suffices to obtain the desired result. □

Second, we point out that if $\mathcal{L}$ contains the set of singletons $\{\{v\} \mid v \in V(H)\}$ then $\#\text{Hom}(H, \mathcal{L})$ is *self-reducible*. So the technique of Jerrum, Valiant and Vazirani [105] reduces the problem of approximately sampling homomorphisms with lists in $\mathcal{L}$ to the problem of approximately counting them. The original notion of self-reducibilty, due to Schnorr [132], relies on careful encodings of instances, so we use instead the more general *self-partitionability* notion of Dyer and Greenhill. Dyer and Greenhill show [41] that the technique of Jerrum, Valiant and Vazirani applies to every self-partitionable problem. Thus, we get the following lemma.

Lemma 3.22. *Let H be a graph and $\mathcal{L} \subseteq \mathcal{P}(V(H))$ such that $\{\{v\} \mid v \in V(H)\} \subseteq \mathcal{L}$. There is an algorithm* $\text{SampleHom}_{H,\mathcal{L}}$ *which uses oracle access to a RAS for $\#\text{Hom}(H, \mathcal{L})$ and has the following properties.*

- *It is given an input $(G, \mathbf{S})$ to $\#\text{Hom}(H, \mathcal{L})$ together with an accuracy parameter $\varepsilon \in (0, 1)$.*
- *The distribution D of its outputs satisfies $\mathrm{d_{TV}}\big(D, \mathrm{Uni}\big(\mathcal{H}((G, \mathbf{S}), H)\big)\big) \leq \varepsilon$.*
- *Its running time is bounded by a polynomial in $\log \varepsilon^{-1}$ and the number of vertices of G.*

Proof. Rather than repeating the (lengthy) formal definition of *self-partitionability* from [41], we state the (self-evident) relevant properties of $\#\text{Hom}(H, \mathcal{L})$ which imply that $\#\text{Hom}(H, \mathcal{L})$ is self-partitionable. The lemma follows immediately from [41].

Let $(G, \mathbf{S})$ be an input to $\#\text{Hom}(H, \mathcal{L})$. If $v \in V(G)$ and $s \in S_v$, then let $\mathbf{S}^{v\to s} = \{S_u^{v\to s} \mid u \in V(G)\}$ be defined as follows.

$$S_u^{v\to s} = \begin{cases} \{s\} & \text{if } u = v \\ S_u & \text{otherwise.} \end{cases}$$

Note that $(G, \mathbf{S}^{v\to s})$ is a valid input to $\#\text{Hom}(H, \mathcal{L})$ as $\{\{v\} \mid v \in V(H)\} \subseteq \mathcal{L}$.

The relevant properties are

1. If for all $v \in V(G)$ we have $S_v \in \{\{u\} \mid u \in V(H)\}$ then the function τ which maps each vertex $v \in V(G)$ to the single element in the corresponding list S_v is the only mapping from G to H that respects the lists. It is then easy to check whether τ is a homomorphism. Therefore, computing $N\big((G, \mathbf{S}) \to H\big)$ and sampling from the set of list homomorphisms from $(G, \mathbf{S})$ to H is trivial.

2. If $v \in V(G)$ then

$$\mathcal{H}((G, \mathbf{S}), H) = \bigcup_{s \in S_v} \mathcal{H}((G, \mathbf{S}^{v \to s}), H). \tag{3.5}$$

Note that the right-hand-side of (3.5) is a union of disjoint sets since all of the homomorphisms in $\mathcal{H}((G, \mathbf{S}^{v \to s}), H)$ map v to s.

These properties imply that $\#\text{Hom}(H, \mathcal{L})$ is self-partitionable in the sense of Dyer and Greenhill, thus the lemma follows from the technique of Jerrum, Valiant and Vazirani, as demonstrated in [41].

This completes the proof of the lemma, but for the reader who wants to relate the above properties to the notation of Dyer and Greenhill, we take the size of an instance $(G, \mathbf{S})$ to be $|\{v \in V(G) \mid |S_v| > 1\}|$. The set of smaller instances $\Xi(G, \mathbf{S})$ considered in [41] can be constructed by fixing any $v \in V(G)$ with $|S_v| > 1$ and then setting $\Xi(G, \mathbf{S}) = \{(G, \mathbf{S}^{v \to s}) \mid s \in S_v\}$. The functions k_ξ mentioned in [41] can all be taken to be constant functions, with output 1. The injection $\phi_{(G, \mathbf{S}^{v \to s})}$ is the identity. Finally $W((G, \mathbf{S}), \tau)$ is just the indicator function that is 1 if τ is a homomorphism from $(G, \mathbf{S})$ to H and 0 otherwise. □

Our first goal is Corollary 3.24 which is an AP-reduction from $\#\text{Comp}(H)$ to $\#\text{Ret}(H)$. The reduction uses a Monte Carlo Algorithm (Algorithm 1). The algorithm is presented more generally, with lists, so that we can also use it in the reductions of Corollaries 3.25, 3.26 and 3.27. The following observation provides the basis for the algorithm. Let H be a graph, G be an irreflexive graph and $\mathbf{S}$ be a corresponding set of lists. If there is a compaction from $(G, \mathbf{S})$ to H then there exists a set $U \subseteq V(G)$ with $|U| \le |V(H)| + 2|E(H)|$ and a compaction τ from $G[U]$ to H. Accordingly, we define

$$\begin{aligned} T_{G,\mathbf{S}} = \{(U, \tau) \mid\ & U \subseteq V(G), |U| \le |V(H)| + 2|E(H)|, \\ & \tau \text{ is a compaction from } G[U] \text{ to } H \text{ such that } \forall u \in U, \tau(u) \in S_u\} \end{aligned} \tag{3.6}$$

and $t_{G,\mathbf{S}} = |T_{G,\mathbf{S}}|$. Let $(U_i, \tau_i)_{i \in [t_{G,\mathbf{S}}]}$ be an arbitrary indexing of the elements of $T_{G,\mathbf{S}}$. For $i \in [t_{G,\mathbf{S}}]$ we define

$$\Omega_{G,\mathbf{S},i} = \{\sigma \in \mathcal{H}((G, \mathbf{S}), H) \mid \ \sigma|_{U_i} = \tau_i\}, \tag{3.7}$$

$$\Omega^+_{G,\mathbf{S}} = \{(i, \sigma) \mid i \in [t_{G,\mathbf{S}}] \text{ and } \sigma \in \Omega_{G,\mathbf{S},i}\}, \text{and} \tag{3.8}$$

$$\Omega_{G,\mathbf{S}} = \Big\{(i, \sigma) \in \Omega^+_{G,\mathbf{S}} \mid \sigma \notin \bigcup_{k=1}^{i-1} \Omega_{G,\mathbf{S},k}\Big\}. \tag{3.9}$$

Note that $\left|\Omega^+_{G,\mathbf{S}}\right| = \sum_{i \in [t_{G,\mathbf{S}}]} |\Omega_{G,\mathbf{S},i}|$. As every element of a set $\Omega_{G,\mathbf{S},i}$ is a compaction from $(G, \mathbf{S})$ to H and every such compaction is contained in a set $\Omega_{G,\mathbf{S},i}$, we have

$$\begin{aligned} |\Omega_{G,\mathbf{S}}| &= \Big|\bigcup_{i \in [t_{G,\mathbf{S}}]} \Omega_{G,\mathbf{S},i}\Big| = \Big|\{\sigma \in \mathcal{H}((G, \mathbf{S}), H) \mid \exists i \in [t_{G,\mathbf{S}}] \text{ such that } \sigma \in \Omega_{G,\mathbf{S},i}\}\Big| \\ &= N^{\text{comp}}\big((G, \mathbf{S}) \to H\big). \end{aligned}$$

It is clear from the definitions that $|\Omega_{G,\mathbf{S}}| \geq |\Omega^+_{G,\mathbf{S}}|/t_{G,\mathbf{S}}$. Thus,

$$N^{\text{comp}}\big((G,\mathbf{S}) \to H\big) = |\Omega_{G,\mathbf{S}}| \geq \frac{|\Omega^+_{G,\mathbf{S}}|}{t_{G,\mathbf{S}}}. \tag{3.10}$$

Intuitively, for some fixed graph H and $\mathcal{L} \subseteq \mathcal{P}(V(H))$ we use this lower bound to construct a Monte Carlo algorithm (Algorithm 1) in the style of [123, Algorithm 11.2], which approximately samples from $\Omega^+_{G,\mathbf{S}}$ to approximately compute $|\Omega_{G,\mathbf{S}}| = N^{\text{comp}}\big((G,\mathbf{S}) \to H\big)$. To this end the algorithm relies on access to a RAS oracle for $\#\text{Hom}(H,\mathcal{L}^*)$ where $\mathcal{L}^* = \mathcal{L} \cup \{\{v\} \mid v \in V(H)\}$.

Algorithm 1 Approximate Computation of $|\Omega_{G,\mathbf{S}}|$. Let H be a fixed graph, $\mathcal{L} \subseteq \mathcal{P}(V(H))$ and $\mathcal{L}^* = \mathcal{L} \cup \{\{v\} \mid v \in V(H)\}$. Then $\textsc{CountHom}_{H,\mathcal{L}^*}$ and $\textsc{SampleHom}_{H,\mathcal{L}^*}$ are the routines from Lemma 3.21 and 3.22, respectively. Let $T_{G,\mathbf{S}}$, $t_{G,\mathbf{S}}$, $\Omega_{G,\mathbf{S},i}$, $\Omega^+_{G,\mathbf{S}}$ and $\Omega_{G,\mathbf{S}}$ be defined as in Equations (3.6)-(3.9). Note that (U_i,τ_i) is the i'th element of $T_{G,\mathbf{S}}$.

Input: Irreflexive graph G with lists $\mathbf{S} = \{S_v \in \mathcal{L} \mid v \in V(G)\}$ and $\varepsilon,\ \delta \in (0,1)$.
  if $t_{G,\mathbf{S}} = 0$
    $Y = 0$.
  else
    $\varepsilon' = \frac{\varepsilon}{12}$, $\delta' = \frac{\delta}{2}$, $\delta'' = \frac{\delta'}{t_{G,\mathbf{S}}}$.
    for $i = 1, \ldots, t_{G,\mathbf{S}}$
      For all $v \in V(G)$, if $v \in U_i$, set $S^i_v = \{\tau_i(v)\}$, otherwise set $S^i_v = S_v$.
      $\mathbf{S}^i = \{S^i_v \mid v \in V(G)\}$
      $\omega_i = \textsc{CountHom}_{H,\mathcal{L}^*}(G,\mathbf{S}^i,\varepsilon',\delta'')$.
    $\omega = \sum_{i=1}^{t_{G,\mathbf{S}}} \omega_i$.
    $m = \left\lceil 6t_{G,\mathbf{S}} \cdot \frac{\ln(2/\delta')}{\varepsilon'^2} \right\rceil$.
    for $j = 1, \ldots, m$
      Choose $i \in [t_{G,\mathbf{S}}]$ with probability $\frac{\omega_i}{\omega}$.
      $\sigma_j = \textsc{SampleHom}_{H,\mathcal{L}^*}(G,\mathbf{S}^i,\varepsilon'/(2|V(H)|^n))$.
      Let X_j be 1 in the event $(i,\sigma_j) \in \Omega_{G,\mathbf{S}}$ and 0 otherwise.
    $Y = \frac{\omega}{m}\sum_{j=1}^m X_j$.
Output: Y

Lemma 3.23. *Algorithm 1 returns an (ε,δ)-approximation of $|\Omega_{G,\mathbf{S}}|$ if it has access to a RAS oracle for $\#\text{Hom}(H,\mathcal{L}^*)$ and every list in $\mathbf{S}$ is an element of $\mathcal{L}$. For fixed δ, the algorithm runs in time polynomial in $n = |V(G)|$ and ε^{-1}.*

Proof. First note that given oracle access to a RAS for $\#\text{Hom}(H,\mathcal{L}^*)$, the routines $\textsc{CountHom}_{H,\mathcal{L}^*}$ and $\textsc{SampleHom}_{H,\mathcal{L}^*}$ exist as shown in Lemmas 3.21 and 3.22 (by definition $\mathcal{L}^*$ contains $\{\{v\} \mid v \in V(H)\}$). Furthermore, the input to these routines is valid: A list $S_v^i \in \mathbf{S}^i$ is either of the form $\{\tau_i(v)\}$ or otherwise $S_v^i = S_v \in \mathcal{L}$. Therefore, in general, $S_v^i \in \mathcal{L}^*$. Thus Algorithm 1 is well-defined.

Next we show that the runtime condition is met. Assume δ to be fixed. Note that we can determine $T_{G,\mathbf{S}}$ exactly by enumerating all possible assignments of at most $|V(H)| + 2|E(H)|$ vertices of G to the vertices of H and checking whether the resulting assignment is a compaction. Checking can be done in polynomial time and $t_{G,\mathbf{S}} = |T_{G,\mathbf{S}}| \leq \sum_{k=1}^{|V(H)|+2|E(H)|} n^k \in \mathrm{poly}(n)$. It follows that $m \in \mathrm{poly}(n, \varepsilon^{-1})$. The runtime of the routine $\mathrm{CountHom}_{H,\mathcal{L}^*}(G, \mathbf{S}^i, \varepsilon', \delta'')$ is in $\mathrm{poly}(n, 1/\varepsilon')$ by Lemma 3.21. Finally, the runtime of $\mathrm{SampleHom}_{H,\mathcal{L}^*}(G, \mathbf{S}^i, \varepsilon'/(2|V(H)|^n))$ is in $\mathrm{poly}(n, \log(1/\varepsilon'))$ by Lemma 3.22. It is essential here that the runtime of $\mathrm{SampleHom}_{H,\mathcal{L}^*}$ has logarithmic dependence on the precision parameter as the precision we use is $\varepsilon'/(2|V(H)|^n)$, which is exponential in n.

If $|T_{G,\mathbf{S}}| = t_{G,\mathbf{S}} = 0$ then $|\Omega_{G,\mathbf{S}}| = 0$ and the algorithm returns an exact solution. To prove the correctness of the algorithm it remains to show that otherwise it is an (ε, δ)-approximation.

Note that by the definition of the $\mathbf{S}^i$ in the first part of the algorithm, $\Omega_{G,\mathbf{S},i} = \mathcal{H}((G, \mathbf{S}^i), H)$. So, by Lemma 3.21 and the definition of δ'', $\mathrm{CountHom}_{H,\mathcal{L}^*}(G, \mathbf{S}^i, \varepsilon', \delta'')$ returns a number ω_i with $\Pr(e^{-\varepsilon'}|\Omega_{G,\mathbf{S},i}| \leq \omega_i \leq e^{\varepsilon'}|\Omega_{G,\mathbf{S},i}|) \geq 1 - \delta'/t_{G,\mathbf{S}}$. By the union bound, with probability of at least $1 - \delta'$, we have

$$e^{-\varepsilon'}|\Omega_{G,\mathbf{S},i}| \leq \omega_i \leq e^{\varepsilon'}|\Omega_{G,\mathbf{S},i}| \qquad (\forall i \in [t_{G,\mathbf{S}}]). \tag{3.11}$$

The following two subclaims are based on this assumption. Let $p = |\Omega_{G,\mathbf{S}}|/\left|\Omega^+_{G,\mathbf{S}}\right|$.

Subclaim 1: Assume that (3.11) holds. Then for all $j \in [m]$ we have $X_j \sim \mathrm{Be}(p')$ where $e^{-3\varepsilon'}p \leq p' \leq e^{3\varepsilon'}p$.

Proof of Subclaim 1: Consider fixed $\omega_1, \ldots, \omega_{t_{G,\mathbf{S}}}$ satisfying (3.11). Note that the distribution of (i, σ_j), conditioned on these, does not depend on the index j. Let D be the distribution (conditioned on $\omega_1, \ldots, \omega_{t_{G,\mathbf{S}}}$) such that for all $j \in [m]$ we have $(i, \sigma_j) \sim D$. By Lemma 3.22 and the fact that $\Omega_{G,\mathbf{S},k} = \mathcal{H}((G, \mathbf{S}^k), H)$ we have

$$D((k,\sigma)) = \Pr(\sigma_j = \sigma \,|\, i = k) \cdot \Pr(i = k) \leq \left(\frac{1}{|\Omega_{G,\mathbf{S},k}|} + \frac{\varepsilon'}{2|V(H)|^n}\right) \cdot \Pr(i = k)$$

First using $|\Omega_{G,\mathbf{S},k}| \leq |V(H)|^n$ and then using Observation 3.4 it follows

$$D((k,\sigma)) \leq \left(1 + \frac{\varepsilon'}{2}\right)\frac{1}{|\Omega_{G,\mathbf{S},k}|} \cdot \Pr(i = k) \leq e^{\varepsilon'}\frac{1}{|\Omega_{G,\mathbf{S},k}|} \cdot \Pr(i = k) = e^{\varepsilon'}\frac{1}{|\Omega_{G,\mathbf{S},k}|} \cdot \frac{\omega_k}{\sum_{i=1}^{t_{G,\mathbf{S}}} \omega_i}.$$

Using the assumption of this subclaim, we obtain

$$D((k,\sigma)) \leq e^{\varepsilon'}\frac{1}{|\Omega_{G,\mathbf{S},k}|} \cdot e^{2\varepsilon'}\frac{|\Omega_{G,\mathbf{S},k}|}{\sum_{i \in [t_{G,\mathbf{S}}]}|\Omega_{G,\mathbf{S},i}|} = e^{3\varepsilon'}\frac{1}{\left|\Omega^+_{G,\mathbf{S}}\right|}$$

and thus

$$p' = \Pr(X_j = 1) = \sum_{(i,\sigma) \in \Omega_{G,\mathbf{S}}} D((i,\sigma)) \leq e^{3\varepsilon'} \sum_{(i,\sigma) \in \Omega_{G,\mathbf{S}}} \frac{1}{\left|\Omega^+_{G,\mathbf{S}}\right|} = e^{3\varepsilon'}p.$$

Analogously we obtain the lower bound $e^{-3\varepsilon'}p \le p'$. **(End of the proof of Subclaim 1.)**

Subclaim 2: Assume that (3.11) holds. Then $e^{-4\varepsilon'}|\Omega_{G,\mathbf{s}}| \le \mathbf{E}[Y] \le e^{4\varepsilon'}|\Omega_{G,\mathbf{s}}|$.

Proof of Subclaim 2: Consider fixed $\omega_1, \dots, \omega_{t_{G,\mathbf{s}}}$ satisfying (3.11). Conditioned on these, we have $X_j \sim \mathrm{Be}(p')$ and

$$\mathbf{E}[Y] = \frac{\sum_{i\in[t_{G,\mathbf{s}}]}\omega_i}{m}\sum_{j\in[m]}\mathbf{E}[X_j] = \sum_{i\in[t_{G,\mathbf{s}}]}\omega_i \cdot p'.$$

We now use (3.11) as well as Subclaim 1 to obtain

$$\mathbf{E}[Y] \le e^{\varepsilon'}\sum_{i\in[t_{G,\mathbf{s}}]}|\Omega_{G,\mathbf{s},i}| \cdot e^{3\varepsilon'}p = e^{4\varepsilon'}|\Omega_{G,\mathbf{s}}|$$

and

$$\mathbf{E}[Y] \ge e^{-\varepsilon'}\sum_{i\in[t_{G,\mathbf{s}}]}|\Omega_{G,\mathbf{s},i}| \cdot e^{-3\varepsilon'}p = e^{-4\varepsilon'}|\Omega_{G,\mathbf{s}}|.$$

(End of the proof of Subclaim 2.)

Next we show that, conditioned on computing ω_i's that satisfy (3.11), the number of samples m is sufficiently large. First, by Subclaim 1, $X_1, \dots, X_m$ are independent indicator random variables that have distribution $\mathrm{Be}(p')$ and expected value p'. By Subclaim 1 and Observation 3.4 we have

$$(1-6\varepsilon')p \le e^{-3\varepsilon'}p \le p' \le e^{3\varepsilon'}p \le (1+6\varepsilon')p.$$

From the definition of ε' it follows that $|p'-p| \le 6\varepsilon' p \le \varepsilon p/2$ and consequently $p' \ge p/2$. Using this fact and taking into account that by Equation (3.10) we have $t_{G,\mathbf{s}} \ge 1/p$, it follows that

$$m = \left\lceil 6t_{G,\mathbf{s}} \cdot \frac{\ln(2/\delta')}{\varepsilon'^2}\right\rceil \ge 6\frac{\ln(2/\delta')}{\varepsilon'^2 p} \ge 3\frac{\ln(2/\delta')}{\varepsilon'^2 p'}.$$

Thus, we can use [123, Theorem 11.1] to obtain $\Pr(|Y-\mathbf{E}[Y]| \ge \varepsilon'\mathbf{E}[Y]) \le \delta'$ which is conditioned on the fact that (3.11) holds. Now taking into account the fact that, with probability at least $1-\delta'$, $\omega_1, \dots, \omega_{t_{G,\mathbf{s}}}$ satisfy (3.11), we have shown that, with probability of at least $(1-\delta')^2 \ge 1-\delta$, we have

$$|Y-\mathbf{E}[Y]| \le \varepsilon'\mathbf{E}[Y] = \frac{\varepsilon}{12}\mathbf{E}[Y].$$

By Subclaim 2 and Observation 3.4 we also know that

$$|\mathbf{E}[Y] - |\Omega_{G,\mathbf{s}}|| \le 8\varepsilon'|\Omega_{G,\mathbf{s}}| = \frac{2\varepsilon}{3}|\Omega_{G,\mathbf{s}}|.$$

Summarising, with probability of at least $1-\delta$, we have

$$\begin{aligned}|Y - |\Omega_{G,\mathbf{s}}|| &\le |Y-\mathbf{E}[Y]| + |\mathbf{E}[Y] - |\Omega_{G,\mathbf{s}}|| \le \frac{\varepsilon}{12}\mathbf{E}[Y] + \frac{2\varepsilon}{3}|\Omega_{G,\mathbf{s}}| \\ &\le \frac{\varepsilon}{12}\left(|\Omega_{G,\mathbf{s}}| + \frac{2\varepsilon}{3}|\Omega_{G,\mathbf{s}}|\right) + \frac{2\varepsilon}{3}|\Omega_{G,\mathbf{s}}| \le \varepsilon|\Omega_{G,\mathbf{s}}|.\end{aligned}$$

Hence, Y is an (ε, δ)-approximation of $|\Omega_{G,\mathbf{s}}|$. □

Corollary 3.24. *Let H be a graph. Then* $\#\textsc{Comp}(H) \leq_{\text{AP}} \#\textsc{Ret}(H)$.

Proof. Let $\mathcal{L} = \{V(H)\}$. Then the problem $\#\textsc{Hom}(H, \mathcal{L}^*)$ is identical to $\#\textsc{Ret}(H)$. Furthermore, given an irreflexive graph G and a set $\mathbf{S} = \{S_v \in \mathcal{L} \mid v \in G\}$, for this choice of $\mathcal{L}$ it holds that $N^{\text{comp}}\big((G, \mathbf{S}) \to H\big) = N^{\text{comp}}\big(G \to H\big)$.

Then, by Lemma 3.23, given a RAS oracle for $\#\textsc{Ret}(H)$, Algorithm 1 computes an (ε, δ)-approximation of $|\Omega_{G,\mathbf{S}}| = N^{\text{comp}}\big((G, \mathbf{S}) \to H\big) = N^{\text{comp}}\big(G \to H\big)$. If we choose $\delta = 1/4$ then the algorithm is an FPRAS for $\#\textsc{Comp}(H)$. □

Corollary 3.25. *Let H be a graph. Then* $\#\text{L}\textsc{Comp}(H) \leq_{\text{AP}} \#\text{L}\textsc{Hom}(H)$.

Proof. Let $\mathcal{L} = \mathcal{P}(V(H))$. Then the problem $\#\textsc{Hom}(H, \mathcal{L}^*) = \#\textsc{Hom}(H, \mathcal{L})$ is identical to $\#\text{L}\textsc{Hom}(H)$.

From Lemma 3.23 it follows that given a RAS oracle for $\#\text{L}\textsc{Hom}(H)$, Algorithm 1 returns an (ε, δ)-approximation of $|\Omega_{G,\mathbf{S}}| = N^{\text{comp}}\big((G, \mathbf{S}) \to H\big)$. In particular, as $\mathcal{L}$ is unrestricted, it does so for any valid input $(G, \mathbf{S})$ of the problem $\#\text{L}\textsc{Comp}(H)$. Thus, if we choose $\delta = 1/4$, the algorithm is an FPRAS for $\#\text{L}\textsc{Comp}(H)$. □

To obtain Corollaries 3.24 and 3.25, the only property of compactions we use is the fact that for every compaction from G to H there exists a preimage U of polynomial size, i.e. a set $U \subseteq V(G)$ with $|U| \leq |V(H)| + 2|E(H)|$ and a compaction τ from $G[U]$ to H. (This is used in Equation (3.6).)

Similarly, for every surjective homomorphism from G to H there exists a set $U \subseteq V(G)$ with $|U| = |V(H)|$ such that there exists a surjective homomorphism τ from $G[U]$ to H. If we substitute

$$\begin{aligned} T_{G,\mathbf{S}} = \{(U, \tau) \mid\ & U \subseteq V(G), |U| = |V(H)|, \\ & \tau \text{ is a surjective homomorphism from } G[U] \text{ to } H \text{ such that } \forall u \in U, \tau(u) \in S_u\} \end{aligned}$$

for Equation (3.6), Lemma 3.23 still holds and now $|\Omega_{G,\mathbf{S}}| = N^{\text{sur}}\big((G, \mathbf{S}) \to H\big)$.

Therefore, analogously to the previous two corollaries we obtain the following.

Corollary 3.26. *Let H be a graph. Then* $\#\text{S}\textsc{Hom}(H) \leq_{\text{AP}} \#\textsc{Ret}(H)$.

Corollary 3.27. *Let H be a graph. Then* $\#\text{LS}\textsc{Hom}(H) \leq_{\text{AP}} \#\text{L}\textsc{Hom}(H)$.

Corollaries 3.24 and 3.26 together constitute Theorem 1.19.

3.4 Additional Reductions and Consequences

We start by showing that both approximately counting surjective homomorphisms and approximately counting compactions are at least as hard as approximately counting unrestricted homomorphisms.

Lemma 3.28. *Let H be a graph. Then* $\#\textsc{Hom}(H) \leq_{\text{AP}} \#\text{S}\textsc{Hom}(H)$.

Proof. Let $q = |V(H)|$. Given any positive integer t, let $s_{t,q}$ denote the number of surjective functions from $[t]$ to $[q]$. Clearly, $s_{t,q} \geq q^t - 2^q(q-1)^t$, since the range of every non-surjective function from $[t]$ to $[q]$ is a proper subset of $[q]$, and there are at most 2^q of these. Also, the number of functions from $[t]$ onto this subset is at most $(q-1)^t$.

Given any n-vertex input G to the problem $\#\textsc{Hom}(H)$, let

$$t = \lceil \log(5q^n 2^q)/\log(q/(q-1)) \rceil.$$

Clearly, $t = O(n)$, and t can be computed in time poly(n). Note that

$$\left(\frac{q}{q-1}\right)^t \geq 5q^n 2^q \geq 4q^n 2^q + 2^q. \tag{3.12}$$

Let G_t be the graph constructed from G by adding a set I_t of t isolated vertices that are distinct from the vertices in $V(G)$. We claim that

$$s_{t,q} N\big(G \to H\big) \leq N^{\text{sur}}\big(G_t \to H\big) \leq s_{t,q} N\big(G \to H\big) + (q^t - s_{t,q})q^n.$$

To see this, note that any homomorphism from G to H, together with a surjective homomorphism from the I_t to $V(H)$, constitutes a surjective homomorphism from G_t to H. Any other surjective homomorphism from G_t to H consists of a non-surjective homomorphism from I_t to H (and there are $q^t - s_{t,q}$ of these) together with some homomorphism from G to H (and there are at most q^n of these). Dividing through by $s_{t,q}$ and applying our lower bound for $s_{t,q}$ and then inequality (3.12), we have

$$\begin{aligned} N\big(G \to H\big) \leq \frac{N^{\text{sur}}\big(G_t \to H\big)}{s_{t,q}} &\leq N\big(G \to H\big) + \left(\frac{q^t - s_{t,q}}{s_{t,q}}\right) q^n \\ &\leq N\big(G \to H\big) + \frac{2^q(q-1)^t q^n}{q^t - 2^q(q-1)^t} \\ &= N\big(G \to H\big) + \frac{q^n}{\frac{q^t}{2^q(q-1)^t} - 1} \\ &\leq N\big(G \to H\big) + \frac{1}{4}. \end{aligned} \tag{3.13}$$

Given Equation (3.13), the proof of [37, Theorem 3] shows that, in order to approximate $N\big(G \to H\big)$ with accuracy ε, we need only use the oracle to obtain an approximation $\widehat{S}$ for $N^{\text{sur}}\big(G_t \to H\big)$ with accuracy $\varepsilon/21$. We can then return the floor of $\widehat{S}/s_{t,q}$. The only remaining issue is how to compute $s_{t,q}$. However, it is easy to do this in time poly(t) = poly(n) since $s_{t,q} = \left\{ {t \atop q} \right\} q! = \sum_{j=0}^{q} (-1)^{q-j} \binom{q}{j} j^t$, where $\left\{ {t \atop q} \right\}$ is a Stirling number of the second kind. □

Lemma 3.29. *Let H be a connected graph. Then $\#\textsc{Hom}(H) \leq_{\text{AP}} \#\textsc{Comp}(H)$.*

Proof. If not explicitly defined otherwise, we use the same notation and observations as in the proof of Lemma 3.28. In addition let p be the number of non-loop edges in H

and $c_{t,p} = 2^t s_{t,p}$. If G is an input to #HOM(H) of size n, G_t is the graph constructed from G by adding a set of t isolated edges distinct from the edges in G. If H is a graph of size 1 the statement of the lemma clearly holds. If otherwise H is a connected graph of size at least 2, every homomorphism that uses all non-loop edges of H is also surjective and therefore a compaction. Thus, we obtain

$$c_{t,p}N\big(G \to H\big) \leq N^{\mathrm{comp}}\big(G_t \to H\big) \leq c_{t,p}N\big(G \to H\big) + (2^t p^t - c_{t,p})q^n.$$

Dividing through by $c_{t,p}$ gives

$$N\big(G \to H\big) \leq \frac{N^{\mathrm{comp}}\big(G_t \to H\big)}{c_{t,p}} \leq N\big(G \to H\big) + \left(\frac{p^t - s_{t,p}}{s_{t,p}}\right)q^n.$$

If we choose $t = \lceil \log(5q^n 2^p)/\log(p/(p-1)\rceil$ the remainder of this proof is analogous to that of Lemma 3.28. □

Using the established reductions we can now prove Theorem 1.21 and Corollary 1.22, which we re-state for the convenience of the reader. The tractability results in Theorem 1.21 come from the tractability of #LHOM(H) from the the Dyer and Greenhill dichotomy for exact counting (Theorem 1.4) together with our reductions from Theorem 1.19 and Observation 1.18. The #BIS-hardness results carry over from results by Galanis, Goldberg and Jerrum for approximately counting homomorphisms (Theorem 1.9) using our reductions from Lemma 1.17 and Observation 1.18.

Theorem 1.21. *Let H be a connected graph. If H is a reflexive clique or an irreflexive biclique, then #SHOM(H), #RET(H) and #COMP(H) are in* FP. *Otherwise, each of these problems is #BIS-hard under approximation-preserving reductions.*

As a corollary we obtain the following #BIS-equivalence results for graphs that are not necessarily square-free using a result by Galanis, Goldberg and Jerrum [63].

Corollary 1.22. *Let H be one of the following:*

- *A reflexive proper interval graph but not a complete graph.*
- *An irreflexive bipartite permutation graph but not a complete bipartite graph.*

Then #SHOM(H), #COMP(H) and #RET(H) are #BIS-equivalent.

The #BIS-easiness results in Corollary 1.22 come from our Theorem 1.19 together with Observation 1.18 and the #BIS-easiness results for #LHOM(H) from Galanis, Goldberg and Jerrum (Theorem 1.10). The corresponding #BIS-hardness comes from Theorem 1.21.

The following simple reductions complete our current knowledge of the complexity landscape given in Figure 1.7.

Lemma 3.30. *Let H be a graph. Then #LHOM(H) $\leq_{\mathrm{AP}}$ #LSHOM(H) and #LHOM(H) $\leq_{\mathrm{AP}}$ #LCOMP(H).*

Proof. Let $v_1, \dots, v_q$ be the vertices of H and let $(G, \mathbf{S})$ be an input to #LHom(H). Further, let H' be a copy of H and let $u_1, \dots, u_q$ be the vertices of H' ordered in the same way as $v_1, \dots, v_q$. For $i \in [q]$ let $S_{u_i} = \{v_i\}$ and let $\mathbf{S}' = \mathbf{S} \cup \{S_{u_i} \colon i \in [q]\}$. Let G' be the disjoint union of G and H'. Then $N\big((G, \mathbf{S}) \to H\big) = N^{\mathrm{sur}}\big((G', \mathbf{S}') \to H\big) = N^{\mathrm{comp}}\big((G', \mathbf{S}') \to H\big)$. □

From Corollaries 3.25 and 3.27 as well as Lemma 3.30 we immediately obtain Theorem 1.20 which we restate at this point.

Theorem 1.20. *Let H be a graph. Then #LSHom(H) $\equiv_{\mathrm{AP}}$ #LHom(H) and #LComp(H) $\equiv_{\mathrm{AP}}$ #LHom(H).*

Using Theorem 1.19 and Corollary 1.14 we can deduce that #SHom(H) and #LHom(H) are also separated subject to the assumption that #BIS and #SAT are not AP-interreducible. The same holds for #Comp(H) and #LHom(H). Moreover, from Theorem 1.20 it follows that we can replace the problem #LHom(H) with #LSHom(H) or #LComp(H) in these separations.

Chapter 4

Approximately Counting Retractions to Square-Free Graphs

A large part of mathematics which becomes useful developed with absolutely no desire to be useful, and in a situation where nobody could possibly know in what area it would become useful; and there were no general indications that it ever would be so. By and large it is uniformly true in mathematics that there is a time lapse between a mathematical discovery and the moment when it is useful; and that this lapse of time can be anything from 30 to 100 years, in some cases even more; and that the whole system seems to function without any direction, without any reference to usefulness, and without any desire to do things which are useful.

–John von Neumann, *The Role of Mathematics (1954)*

This chapter is based on the following preprint:

[59] Jacob Focke, Leslie Ann Goldberg, and Stanislav Živný. The complexity of approximately counting retractions to square-free graphs. *arXiv preprint arXiv:1907.02319*, 2019.

Some parts of Section 4.3.3 are from

[58] Jacob Focke, Leslie Ann Goldberg, and Stanislav Živný. The complexity of approximately counting retractions. *ACM Transactions of Computation Theory*, 12(3):Art. 15, 43, 2020.

– A preliminary version of this work appeared in the Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, pp. 2205-2215.

Contents

Organisation of this Chapter

In this chapter, we extend Theorem 1.11 from Chapter 3 to a complete complexity trichotomy for all square-free graphs (Theorem 1.13). An interesting feature of this classification is the rich and surprising class of graphs for which the problem is AP-interreducible with #BIS. The corresponding easiness results (including the proof of Theorem 1.12) are presented in Section 4.2.

The complementing #SAT-hardness results are collected in Section 4.3. Proving #SAT-hardness is the bulk of this work because of the combinatorial complexity of designing reductions which establish #SAT-hardness for all square-free graphs (apart from reflexive cliques, irreflexive caterpillars and those in $\mathcal{H}_{\mathrm{BIS}}$). Here we identify a number of different structures that, if present in a square-free graph H, induce hardness.

We combine all these results in Section 4.4 in order to prove the main result, Theorem 1.13.

4.1 Preliminaries

For a non-negative integer k we use $[k]$ to denote the set $\{1, \dots, k\}$. For sets X and Y we define $X \times Y = \{\{x, y\} \mid x \in X, y \in Y\}$ as an undirected version of the Cartesian product. The elements of $X \times Y$ are *multisets* of size exactly 2. Using this notation the set of edges $E(H)$ of a graph $H = (V(H), E(H))$ is a subset of $V(H) \times V(H)$. Recall that an edge with two identical elements is a *loop*. Correspondingly, a vertex $v \in V(H)$ is called *looped* if $\{v, v\} \in E(H)$ and unlooped otherwise. From the introduction recall that the *girth* of a graph H is the length of a shortest cycle in H. All cycles have length at least 3. (We do not consider a loop as a cycle.)

We have already defined reflexive and irreflexive graphs in Chapter 1. A graph H is a *mixed* graph if it contains both looped and unlooped vertices, i.e., if it is neither reflexive nor irreflexive. Given a graph H and a subset U of $V(H)$, $H[U]$ is the *subgraph of H induced by U*.

Given graphs G and H, $\mathcal{H}(G,H)$ is the set of homomorphisms from G to H and $N\big(G \to H\big)$ denotes its size. Analogously, given a corresponding set of lists $\mathbf{S}$, $\mathcal{H}((G,\mathbf{S}),H)$ is the set of homomorphisms from $(G,\mathbf{S})$ to H and $N\big((G,\mathbf{S}) \to H\big)$ denotes its size.

We recall the formal definitions of the problems of counting retractions and counting homomorphisms to H (for a fixed graph H which may have loops but does not have multi-edges).

Name: #RET(H).
Input: An irreflexive graph G and a collection of lists $\mathbf{S} = \{S_v \subseteq V(H) \mid v \in V(G)\}$ such that, for all $v \in V(G)$, $|S_v| \in \{1, |V(H)|\}$.
Output: $N\big((G,\mathbf{S}) \to H\big)$.

Name: #HOM(H).
Input: An irreflexive graph G.
Output: $N\big(G \to H\big)$.

The list homomorphisms counting problem, defined as follows, is a generalisation of #RET(H).

Name: #LHOM(H).
Input: An irreflexive graph G and a collection of lists $\mathbf{S} = \{S_v \subseteq V(H) \mid v \in V(G)\}$.
Output: $N\big((G,\mathbf{S}) \to H\big)$.

The concept of approximation-preserving reductions (AP-reductions) was introduced in Chapter 1. Our goal is to determine the relative complexity of approximately counting retractions. For convenience we say that a graph H is #BIS-easy if the problem #RET(H) AP-reduces to #BIS. We say that a graph H is #BIS-hard if #BIS AP-reduces to #RET(H). We similarly use the terms #SAT-easy, and #SAT-hard.

4.2 #BIS-Easiness Results

In this section we prove Theorem 1.12, which states that approximately counting retractions to any graph from the class $\mathcal{H}_{\text{BIS}}$ (Definition 4.2) is #BIS-equivalent. The proof is built on the method for generating #BIS-easiness results from Section 3.2.2.1, which uses the framework of constraint satisfaction problems. Intuitively, the method takes as input two CSP instances, say I_{v} and I_{e}, and produces a graph $H_{I_{\text{v}},I_{\text{e}}}$ for which #RET($H_{I_{\text{v}},I_{\text{e}}}$) $\leq_{\text{AP}}$ #BIS. The challenge is to find the right instances I_{v} and I_{e} and to identify and generate corresponding general classes of #BIS-easy graphs. For the convenience of the reader we repeat some definitions introduced in Chapter 3. Let $\mathcal{L}$ be a set of Boolean relations.

Name: #CSP($\mathcal{L}$).
Input: A set of variables X and a set of constraints C, where each constraint applies a relation from $\mathcal{L}$ to a list of variables from X.
Output: The number of assignments $\sigma\colon X \to \{0,1\}$ that satisfy all constraints in C.

Imp $= \{(0,0),(0,1),(1,1)\}$ is an arity-two Boolean relation. The constraint Imp(x,y) ensures that, in any satisfying assignment σ, we have $\sigma(x) \implies \sigma(y)$.

Definition 3.10. Let $I_{\mathrm{v}} = (X, C_{\mathrm{v}})$ and $I_{\mathrm{e}} = (X, C_{\mathrm{e}})$ be instances of #CSP({Imp}). We define the undirected graph $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$ as follows. The vertices of $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$ are the satisfying assignments of I_{v}. Given any assignments σ and σ' in $V(H_{I_{\mathrm{v}},I_{\mathrm{e}}})$, there is an edge $\{\sigma,\sigma'\}$ in $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$ if and only if the following holds: For every constraint Imp(x,y) in I_{e}, we have $\sigma(x) \Rightarrow \sigma'(y)$ and $\sigma'(x) \Rightarrow \sigma(y)$.

Lemma 4.1 (Lemmas 3.9 and 3.12)**.** *Let* $I_{\mathrm{v}} = (X, C_{\mathrm{v}})$ *and* $I_{\mathrm{e}} = (X, C_{\mathrm{e}})$ *be instances of* #CSP({Imp})*. Then* #RET$(H_{I_{\mathrm{v}},I_{\mathrm{e}}}) \leq_{\mathrm{AP}}$ #BIS*.*

Definition 4.2. A graph H is in $\mathcal{H}_{\mathrm{BIS}}$ if it can be defined as follows. For some positive integer Q, the vertex set $V(H)$ is of the form $V(H) = \bigcup_{i=0}^{Q} K_i \cup \bigcup_{i=1}^{Q} B_i$ where $K_0,\dots,K_Q$ induce reflexive cliques in H, and $B_1,\dots,B_Q$ are disjoint sets of unlooped degree-1 vertices (called *bristles*). There are $Q+2$ vertices $p_0,\dots,p_{Q+1}$ in $V(H)$ such that each clique K_i contains both p_i and p_{i+1}. The intersection of the cliques is given as follows.

- For $i \in [Q]$, $K_{i-1} \cap K_i = \{p_i\}$.
- For $i,j \in \{0,\dots,Q\}$ with $|j-i|>1$, $K_i \cap K_j = \emptyset$.

The size of each set B_i of bristles satisfies $0 \leq |B_i| \leq (|K_{i-1}|-1)\cdot(|K_i|-1)$. Finally, the edge set of H is given as follows.

$$E(H) = \bigcup_{i=0}^{Q}(K_i \times K_i) \cup \bigcup_{i=1}^{Q}(\{p_i\} \times B_i).$$

For an example graph from the class $\mathcal{H}_{\mathrm{BIS}}$ see Figure 4.1.

Let $H \in \mathcal{H}_{\mathrm{BIS}}$ be as defined in Definition 4.2. The high-level-structure of the proof of Theorem 1.12 is as follows. We first define two instances I_{v} and I_{e} of #CSP({Imp}), then we establish that H is isomorphic to $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$ (Lemma 4.13), which then allows us to apply Lemma 4.1.

To give more intuition we will use a running example where H is the graph depicted in Figure 4.1. To separate this example from the rest of the proof we use text boxes.

Let V^* be the set of looped vertices in H, i.e. $V^* = \bigcup_{i=0}^{Q} K_i$ and let $X = \{x_v \mid v \in V^* \setminus \{p_0\}\}$ be a set of Boolean variables. We fix an ordering "<" on the vertices of V^* with two properties: (1) In K_i, p_i is the smallest vertex and p_{i+1} is the largest; (2)

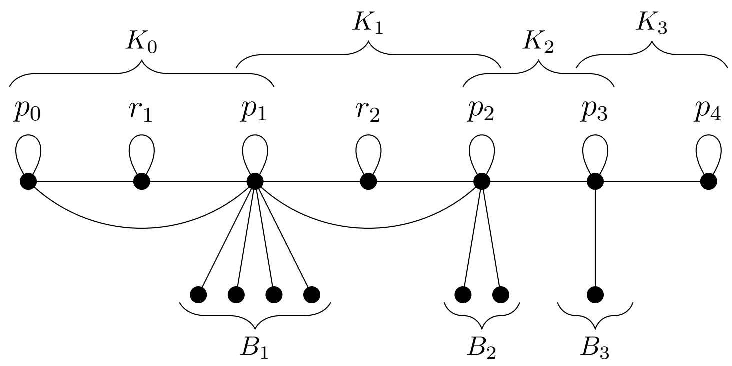


Figure 4.1: Example graph from $\mathcal{H}_{\mathrm{BIS}}$ for $Q = 3$. Note that $|B_1| = 4 = (|K_0| - 1) \cdot (|K_1| - 1)$, $|B_2| = 2 = (|K_1| - 1) \cdot (|K_2| - 1)$ and $|B_3| = 1 = (|K_2| - 1) \cdot (|K_3| - 1)$.

The order of the K_i's is respected in the sense that, for any pair of distinct vertices $u, v \in V^*$, if there is an $i < j$ such that $u \in K_i$ and $v \in K_j$, then $u < v$. We define $U = \{\mathrm{Imp}(x_u, x_v) \mid u > v\}$.

Consider $I_{\mathrm{v}}^* = (X, U)$ and $I_{\mathrm{e}}^* = (X, U)$. The graph $H_{I_{\mathrm{v}}^*, I_{\mathrm{e}}^*}$ is simply a reflexive path on $|V^*|$ vertices. We will construct I_{v} from I_{v}^* by choosing a subset C_{v} of U — this allows the creation of bristles. Similarly we will construct I_{e} from I_{e}^* by choosing a subset C_{e} of U — this creates reflexive cliques amongst the vertices of the reflexive path in $H_{I_{\mathrm{v}}^*, I_{\mathrm{e}}^*}$. In order to define C_{v} and C_{e}, for $i \in \{0, \dots, Q\}$, we define the sets of constraints $D_e(i)$ (the constraints that we will delete from U to define C_{e}) as follows.

$$D_e(i) = \{\mathrm{Imp}(x_u, x_v) \in U \mid u, v \in K_i \setminus \{p_i\}\}. \tag{4.1}$$

For $i \geq 1$ (i.e. for $i \in [Q]$), we define the sets of constraints $D_v(i)$ (the constraints we will delete from U to define C_{v}). The definition of $D_v(i)$ is a bit more involved and uses the following sets:

$$A(i) = \{\mathrm{Imp}(x_u, x_v) \in U \mid u \in K_i \setminus \{p_i\},\ v \in K_{i-1} \setminus \{p_{i-1}\}\}. \tag{4.2}$$

In order to model the set of bristles B_i we will "delete" exactly $|B_i|$ constraints that belong to $A(i)$ from C_{v}. As a means to specify which constraints will be deleted we define an order on the constraints in U (which uses the order $<$ on the vertices in V^* which we fixed previously). There are several orders which would work.

Definition 4.3. We define an order "$\preceq$" on U. Let $\mathrm{Imp}(x_u, x_v), \mathrm{Imp}(x_{u'}, x_{v'}) \in U$. Then $\mathrm{Imp}(x_u, x_v) \preceq \mathrm{Imp}(x_{u'}, x_{v'})$ if one of the following holds:

- $u < u'$.
- $u = u'$ and $v \geq v'$.

If $\mathrm{Imp}(x_u, x_v) \preceq \mathrm{Imp}(x_{u'}, x_{v'})$ and the ordered pair (u, v) is distinct from the ordered pair (u', v') then $\mathrm{Imp}(x_u, x_v) \prec \mathrm{Imp}(x_{u'}, x_{v'})$.

Now let $D_v(i)$ be the $|B_i|$ smallest elements of $A(i)$ with respect to $\preceq$ as given in Definition 4.3. Note that by the definition of $A(i)$ (Equation (4.2)) and the bound on $|B_i|$ (Definition 4.2) this is well-defined since

$$|B_i| \leq (|K_{i-1}| - 1) \cdot (|K_i| - 1) = |A(i)|.$$

Finally, we define C_v and C_e as follows.

$$C_\mathrm{v} = U \setminus \left(\bigcup_{i=1}^{Q} D_v(i) \right) \quad \text{and} \quad C_\mathrm{e} = U \setminus \left(\bigcup_{i=0}^{Q} D_e(i) \right). \tag{4.3}$$

In our running example, order the variables in V^* from left to right. We have $X = \{x_{r_1}, x_{p_1}, x_{r_2}, x_{p_2}, x_{p_3}, x_{p_4}\}$ as the set of variables of I_v and I_e. Since $|B_1| = 4$ the set $D_v(1)$ contains the 4 smallest elements of $A(1) = \{\mathrm{Imp}(x_{r_2}, x_{p_1}), \mathrm{Imp}(x_{r_2}, x_{r_1}), \mathrm{Imp}(x_{p_2}, x_{p_1}), \mathrm{Imp}(x_{p_2}, x_{r_1})\}$, which means $D_v(1) = A(1)$. Similarly, since $|B_2| = 2$, the set $D_v(2)$ contains the 2 smallest elements of $A(2) = \{\mathrm{Imp}(x_{p_3}, x_{p_2}), \mathrm{Imp}(x_{p_3}, x_{r_2})\}$, which means $D_v(2) = A(2)$. Finally, since $|B_3| = 1$, we have $D_v(3) = A(3) = \{\mathrm{Imp}(x_{p_4}, x_{p_3})\}$. Thus,

$$\begin{aligned} C_\mathrm{v} = \{&\mathrm{Imp}(x_{p_4}, x_{p_2}), \mathrm{Imp}(x_{p_4}, x_{r_2}), \mathrm{Imp}(x_{p_4}, x_{p_1}), \mathrm{Imp}(x_{p_4}, x_{r_1}), \\ &\mathrm{Imp}(x_{p_3}, x_{p_1}), \mathrm{Imp}(x_{p_3}, x_{r_1}), \mathrm{Imp}(x_{p_2}, x_{r_2}), \mathrm{Imp}(x_{p_1}, x_{r_1})\}. \end{aligned} \tag{4.4}$$

Regarding the edge constraints we have $D_e(0) = \{\mathrm{Imp}(x_{p_1}, x_{r_1})\}$, $D_e(1) = \{\mathrm{Imp}(x_{p_2}, x_{r_2})\}$, $D_e(2) = \emptyset$ and $D_e(3) = \emptyset$ and hence

$$C_\mathrm{e} = U \setminus \{\mathrm{Imp}(x_{p_1}, x_{r_1}), \mathrm{Imp}(x_{p_2}, x_{r_2})\}. \tag{4.5}$$

Recall that the satisfying assignments of I_v correspond to the vertices of $H_{I_\mathrm{v}, I_\mathrm{e}}$. For $v \in V^*$ let $\sigma_v \colon X \to \{0, 1\}$ be the assignment with

$$\sigma_v(x_u) = \begin{cases} 1, & \text{if } u \leq v \\ 0, & \text{otherwise,} \end{cases}$$

where $u \in V^* \setminus \{p_0\}$ (i.e. $x_u \in X$). Note that σ_{p_0} is the all-zero assignment since p_0 is the minimum vertex in V^*. The reason that we did not introduce a variable for p_0 in the definition of X is that its role is captured by the all-zero assignment to X. We will call assignments of the form σ_v *path assignments*. The path assignments inherit an order from the order on the set V^* that we fixed, i.e. $\sigma_u < \sigma_v$ if and only if $u < v$.

Lemma 4.4. *All path assignments satisfy $I_\mathrm{v} = (X, C_\mathrm{v})$. If $\sigma_{p_0}, \ldots, \sigma_{p_{Q+1}}$ are the path assignments ordered by $<$, then they form a reflexive path in $H_{I_\mathrm{v}, C_\mathrm{v}}$. The reflexive path is not necessarily induced by its vertices.*

Proof. This proof merely requires that $C_\mathrm{v} \subseteq U$ and $C_\mathrm{e} \subseteq U$ — it does not use the detailed definitions of C_v and C_e.

First, note that the assignments σ_v satisfy the #CSP({Imp})-instance (X, U). Thus, they will still be satisfying assignments if we delete constraints from U.

We now investigate the edges between the vertices $\sigma_{p_0}, \dots, \sigma_{p_{Q+1}}$ of $H_{I_\mathrm{v},I_\mathrm{e}}$. From Definition 3.10 recall that, given any satisfying assignments σ and σ' of I_v, there is an edge $\{\sigma, \sigma'\}$ in $H_{I_\mathrm{v},I_\mathrm{e}}$ if and only if the following holds:

$$\text{For every constraint } \mathrm{Imp}(x, y) \text{ in } C_\mathrm{e}, \text{ we have } \sigma(x) \implies \sigma'(y) \text{ and } \sigma'(x) \implies \sigma(y). \tag{4.6}$$

Using (4.6) it can easily be checked that for $C_\mathrm{e} = U$ (and therefore for all $C_\mathrm{e} \subseteq U$), the vertices $\sigma_{p_0}, \dots, \sigma_{p_{Q+1}}$ form a reflexive path, i.e. these vertices are looped and for $v \in V^* \setminus \{p_0\}$ and $v' = \max\{u \in V^* \mid u < v\}$ we have that $\{\sigma_{v'}, \sigma_v\}$ is an edge in $H_{I_\mathrm{v},I_\mathrm{e}}$. □

Lemma 4.4 shows that even if we were to enforce all constraints in U both in the modelling of vertices ($I^*_\mathrm{v} = (X, U)$) and of edges ($I^*_\mathrm{e} = (X, U)$), the graph $H_{I^*_\mathrm{v},I^*_\mathrm{e}}$ always contains a reflexive path on $|V^*|$ vertices. As noted earlier, $H_{I^*_\mathrm{v},I^*_\mathrm{e}}$ is precisely this reflexive path. Our definition of C_e, given in (4.3), ensures that $C_\mathrm{e} \subseteq U$ so it uses a subset of the constraints, which leads to the possibilitiy of more edges in $H_{I_\mathrm{v},I_\mathrm{e}}$. The idea behind the construction is that a vertex $v \in V^*$ will be modelled by the satisfying assignment σ_v, which is a vertex in $H_{I_\mathrm{v},I_\mathrm{e}}$. The previous lemma shows that these assignments form a reflexive path. We now show how the clique structure is modelled. Intuitively, the deleted constraints allow shortcuts that form cliques along the underlying path of path assignments. The following observation follows immediately from the definition of $D_e(i)$ in (4.1).

Observation 4.5. *For all $i \in [Q]$, $v \in V^*$ with $p_i < v$ and all $k \in \{0, \dots, Q\}$, $\mathrm{Imp}(x_v, x_{p_i}) \notin D_e(k)$ and consequently $\mathrm{Imp}(x_v, x_{p_i}) \in C_\mathrm{e}$ (since $\mathrm{Imp}(x_v, x_{p_i}) \in U$).*

Lemma 4.6. *Two vertices $u, v \in V^*$ are adjacent in H if and only if σ_u is adjacent to σ_v in $H_{I_\mathrm{v},I_\mathrm{e}}$.*

Proof. First we consider the case where $u, v \in V^*$ are adjacent. Then there exists an index i such that $u, v \in K_i$. Consequently $p_i \le u \le p_{i+1}$ and $p_i \le v \le p_{i+1}$. Therefore, for $w \in V^*$, if $w \le p_i$ then $\sigma_u(x_w) = 1$ and $\sigma_v(x_w) = 1$, and if $w > p_{i+1}$ then $\sigma_u(x_w) = 0$ and $\sigma_v(x_w) = 0$. Let $\mathrm{Imp}(x_t, x_s)$ be some constraint in C_e. Then $\mathrm{Imp}(x_t, x_s) \notin D_e(i)$ and, by the Definition of $D_e(i)$ in (4.1), we have at least one of $s \le p_i$ or $t > p_{i+1}$. Using these facts, we can verify (4.6) for $\sigma = \sigma_u$ and $\sigma' = \sigma_v$ which shows that σ_u is adjacent to σ_v in $H_{I_\mathrm{v},I_\mathrm{e}}$.

Now assume that u and v are not adjacent. Then there exist indices $i \neq j$ from $\{0, \dots, Q\}$ such that $u \in K_i \setminus \{p_{i+1}\}$, $v \notin K_i$, $v \in K_j \setminus \{p_{j+1}\}$ and $u \notin K_j$. Without loss of generality assume $i < j$. Then $u < p_{i+1} < v$ so $\sigma_v(x_v) = 1$ and $\sigma_u(x_{p_{i+1}}) = 0$. However, by Observation 4.5, $\mathrm{Imp}(x_v, x_{p_{i+1}}) \in C_\mathrm{e}$ and consequently σ_u and σ_v are not adjacent in $H_{I_\mathrm{v},I_\mathrm{e}}$. □

In our example, the vertices $\sigma_{p_0}, \sigma_{r_1}, \sigma_{p_1}, \sigma_{r_2}, \sigma_{p_2}, \sigma_{p_3}, \sigma_{p_4}$ form a reflexive path (as guaranteed by Lemma 4.4). Furthermore, $\{\sigma_{p_0}, \sigma_{p_1}\}$ is an edge since $\mathrm{Imp}(x_{p_1}, x_{r_1}) \notin C_{\mathrm{e}}$ by (4.5), and $\{\sigma_{p_1}, \sigma_{p_2}\}$ is an edge since $\mathrm{Imp}(x_{p_2}, x_{r_2}) \notin C_{\mathrm{e}}$ by (4.5). The subgraph of $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$ induced by the path assignments is as depicted in Figure 4.2 and is isomorphic to $H[V^*] = H[\{p_0, r_1, p_1, r_2, p_2, p_3, p_4\}]$ (as is guaranteed by Lemma 4.6).

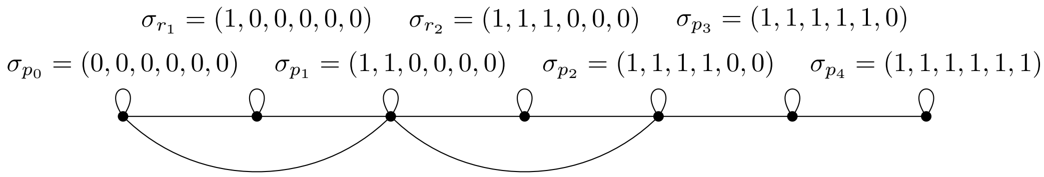


Figure 4.2: The subgraph of $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$ induced by the path assignments. If a vertex corresponds to a path assignment σ_v of I_{v} then its label is of the form $\sigma_v = (\sigma_v(x_{r_1}), \sigma_v(x_{p_1}), \sigma_v(x_{r_2}), \sigma_v(x_{p_2}), \sigma_v(x_{p_3}), \sigma_v(x_{p_4}))$.

With Lemma 4.6 we have established the clique structure of the looped vertices. Next we will model the bristles that are attached to the vertices $p_1, \dots, p_Q$. To do this, we consider satisfying assignments of I_{v} that are not path assignments, i.e. that are not of the form σ_v. These assignments are called *bristle assignments*.

Definition 4.7. An assignment $\beta\colon X \to \{0,1\}$ is a *bristle assignment* if and only if there exist $u, v \in V^* \setminus \{p_0\}$ with $u < v$ such that $\beta(x_u) = 0$ and $\beta(x_v) = 1$.

Definition 4.8. For $i \in [Q]$, $a \in K_{i-1} \setminus \{p_{i-1}\}$ and $b \in K_i \setminus \{p_i\}$ let $\beta_i[a,b]\colon X \to \{0,1\}$ be the assignment with

$$\beta_i[a,b](x_u) = \begin{cases} 1, & \text{if } u < a \\ 0, & \text{if } a \leq u \leq p_i \\ 1, & \text{if } p_i < u \leq b \\ 0, & \text{otherwise (if } b < u\text{)}, \end{cases} \tag{4.7}$$

where $u \in V^* \setminus \{p_0\}$. Hence $\beta_i[a,b]$ is of the following form.

x_u		$\dots$		x_a	$\dots$	x_{p_i}		$\dots$	x_b		$\dots$	$x_{p_{Q+1}}$
$\beta_i[a,b](x_u)$	1	$\dots$	1	0	$\dots$	0	1	$\dots$	1	0	$\dots$	0

Note that a is the minimum index for which $\beta_i[a,b]$ takes value 0, and b is the maximum index for which $\beta_i[a,b]$ takes value 1. We say that a bristle assignment is *good* if it is of the form $\beta_i[a,b]$.

Observation 4.9. *Since $C_{\mathrm{v}} \subseteq U$ the following are equivalent:*

(I) $\beta_i[a,b]$ is a satisfying assignment of I_{v}.

(II) For all $s, t \in V^* \setminus \{p_0\}$ with $a \leq s \leq p_i$ and $p_i < t \leq b$ it holds that $\mathrm{Imp}(x_t, x_s) \notin C_{\mathrm{v}}$.

Lemma 4.10. *Every bristle assignment that satisfies $I_{\mathrm{v}} = (X, C_{\mathrm{v}})$ is good.*

Proof. The property of C_{v} that is used in this proof is that $U \setminus \bigcup_{i\in[Q]} A(i) \subseteq C_{\mathrm{v}}$ and therefore if a constraint from U is not in C_{v} it has to be in one of the sets $A(i)$.

Let β be a bristle assignment that satisfies I_{v}. Let a be the minimum index with $\beta(x_a) = 0$ and let b be the maximum index with $\beta(x_b) = 1$. Since β is a bristle assignment, $b > a$. Then, since β is a satisfying assignment, $\mathrm{Imp}(x_b, x_a) \notin C_{\mathrm{v}}$. Therefore $\mathrm{Imp}(x_b, x_a) \in A(i)$ for some $i \in [Q]$, and therefore (by the definition of $A(i)$), $b \in K_i \setminus \{p_i\}$ and $a \in K_{i-1} \setminus \{p_{i-1}\}$. This is consistent with Definition 4.8. We will show that $\beta = \beta_i[a, b]$. Let $u \in V^* \setminus \{p_0\}$. We investigate the value of $\beta(x_u)$ depending on u to show that β takes values as given in (4.7):

- If $u < a$, by the minimality of a we have $\beta(x_u) = 1$.
- If $u = a$, then $\beta(x_u) = \beta(x_a) = 0$ by the choice of a.
- If $a < u \leq p_i$, then $u \notin K_i \setminus \{p_i\}$ which implies $\mathrm{Imp}(x_u, x_a) \notin A(i)$ and consequently, since $\mathrm{Imp}(x_u, x_a)$ is in U, we have $\mathrm{Imp}(x_u, x_a) \in C_{\mathrm{v}}$ (also using the fact that $\mathrm{Imp}(x_u, x_a)$ cannot be in any of the other sets $A(k)$ since $a \in K_{i-1} \setminus \{p_{i-1}\}$). Therefore it holds that $\beta(x_u) = 0$.
- If $p_i < u < b$, then $u \notin K_{i-1}$ which implies $\mathrm{Imp}(x_b, x_u) \notin A(i)$ and consequently, since $\mathrm{Imp}(x_b, x_u)$ is in U, we have $\mathrm{Imp}(x_b, x_u) \in C_{\mathrm{v}}$ (also using the fact that $\mathrm{Imp}(x_b, x_u)$ cannot be in any of the other sets $A(k)$ since $b \in K_i \setminus \{p_i\}$). Therefore it holds that $\beta(x_u) = 1$.
- If $u = b$, then $\beta(x_u) = \beta(x_b) = 1$ by the choice of b.
- If $b < u$, by the maximality of b we have $\beta(x_u) = 0$.

□

We can now check which of the (good) bristle assignments satisfy I_{v} in the running example. First, consider the set $D_v(1)$. We already showed, in the text box containing (4.4), that $D_v(1) = \{\mathrm{Imp}(x_{r_2}, x_{p_1}), \mathrm{Imp}(x_{r_2}, x_{r_1}), \mathrm{Imp}(x_{p_2}, x_{p_1}), \mathrm{Imp}(x_{p_2}, x_{r_1})\}$. From the definition of C_{v} (Equation (4.3)),

(i) $\mathrm{Imp}(x_{r_2}, x_{p_1}) \notin C_{\mathrm{v}}$.
(ii) $\mathrm{Imp}(x_{r_2}, x_{r_1}) \notin C_{\mathrm{v}}$.
(iii) $\mathrm{Imp}(x_{p_2}, x_{p_1}) \notin C_{\mathrm{v}}$.
(iv) $\mathrm{Imp}(x_{p_2}, x_{r_1}) \notin C_{\mathrm{v}}$.

The good bristle assignments of the form $\beta_1[a, b]$ have $a \in K_0 \setminus \{p_0\} = \{r_1, p_1\}$ and $b \in K_1 \setminus \{p_1\} = \{r_2, p_2\}$ so they are $\beta_1[p_1, r_2]$, $\beta_1[r_1, r_2]$, $\beta_1[p_1, p_2]$ and $\beta_1[r_1, p_2]$, which are as follows:

x_u	x_{r_1}	x_{p_1}	x_{r_2}	x_{p_2}	x_{p_3}	x_{p_4}
$\beta_1[p_1, r_2](x_u)$	**1**	**0**	**1**	**0**	0	0
$\beta_1[r_1, r_2](x_u)$	**0**	**0**	**1**	**0**	0	0
$\beta_1[p_1, p_2](x_u)$	**1**	**0**	**1**	**1**	0	0
$\beta_1[r_1, p_2](x_u)$	**0**	**0**	**1**	**1**	0	0

We now apply Observation 4.9. From (i) it follows that $\beta_1[p_1, r_2]$ satisfies I_{v}. Similarly, from (i) and (ii) it follows that $\beta_1[r_1, r_2]$ satisfies I_{v}. From (i) and (iii) it follows that $\beta_1[p_1, p_2]$ satisfies I_{v}. Finally, from (i), (ii), (iii) and (iv) it follows that $\beta_1[r_1, p_2]$ satisfies I_{v}. The four bristle assignments that we have checked correspond to the four bristles in B_1 in Figure 4.1. Similarly, one can check, from the definitions of $D_v(2)$ and $D_v(3)$, that $\beta_2[p_2, p_3]$, $\beta_2[r_2, p_3]$ and $\beta_3[p_3, p_4]$ are the only other satisfying bristle assignments.

Lemma 4.11. *For every bristle assignment β that satisfies $I_{\mathrm{v}} = (X, C_{\mathrm{v}})$ there exists $i \in [Q]$ such that σ_{p_i} is the only neighbour of β in $H_{I_{\mathrm{v}}, I_{\mathrm{e}}}$.*

Proof. Here we will use the fact that $U \setminus \bigcup_{i \in [Q]} A(i) \subseteq C_{\mathrm{v}}$ and the fact that $C_{\mathrm{e}} = U \setminus \left(\bigcup_{k=0}^{Q} D_e(k)\right)$.

Let β be a bristle assignment that satisfies I_{v}. By Lemma 4.10, β is good, i.e. there exist $i \in [Q]$, $a \in K_{i-1} \setminus \{p_{i-1}\}$ and $b \in K_i \setminus \{p_i\}$ such that $\beta = \beta_i[a, b]$. We will show that σ_{p_i} is the only neighbour of $\beta = \beta_i[a, b]$ in $H_{I_{\mathrm{v}}, I_{\mathrm{e}}}$.

Let ψ be some satisfying assignment of I_{v} which is adjacent to $\beta_i[a, b]$. Note that $\beta_i[a, b](x_{p_i}) = 0$. This fact together with Observation 4.5 (which states that for all $u \in V^*$ with $p_i < u$ we have $\mathrm{Imp}(x_u, x_{p_i}) \in C_{\mathrm{e}}$) implies that, for all $u > p_i$, $\psi(x_u) = 0$.

Let v be the minimum vertex with $p_i < v$. Then $p_i < v \leq b$ and therefore $\beta_i[a, b](x_v) = 1$. The choice of v ensures that for all vertices $u \in V^*$, $u \leq p_i$ iff $u < v$. Therefore, if $u \leq p_i$ we have $\mathrm{Imp}(x_v, x_u) \in C_{\mathrm{e}}$ (since $v \in K_i \setminus \{p_i\}$ but $u \notin K_i \setminus \{p_i\}$ and hence $\mathrm{Imp}(x_v, x_u) \notin D_e(i)$) and consequently $\psi(x_u) = 1$. Summarising, we obtain

$$\text{for all } u \leq p,\ \psi(x_u) = 1 \text{ and for all } u > p_i,\ \psi(x_u) = 0.$$

Thus, $\psi = \sigma_{p_i}$.

It remains to check that $\beta_i[a,b]$ and σ_{p_i} are in fact adjacent. To this end we verify (4.6):

Claim: If $\mathrm{Imp}(x_t, x_s) \in C_\mathrm{e}$ then $\beta_i[a,b](x_t) \Rightarrow \sigma_{p_i}(x_s)$.

Proof of the claim: We check for possible violations. The only relevant s and t are those for which $s < t$, $\beta_i[a,b](x_t) = 1$ and $\sigma_{p_i}(x_s) = 0$, i.e., all s and t satisfying $p_i < s < t \leq b \leq p_{i+1}$. However, constraints of this form are in $D_e(i)$ and hence are not in C_e. *(End of the proof of the claim.)*

Claim: If $\mathrm{Imp}(x_t, x_s) \in C_\mathrm{e}$ then $\sigma_{p_i}(x_t) \Rightarrow \beta_i[a,b](x_s)$.

Proof of the claim: Again we check for violations. The only relevant s and t are those for which $s < t$, $\sigma_{p_i}(x_t) = 1$ and $\beta_i[a,b](x_s) = 0$, i.e., all s and t satisfying $p_{i-1} < a \leq s < t \leq p_i$. However, constraints of this form are in $D_e(i-1)$ and hence are not in C_e. *(End of the proof of the claim.)* □

Lemma 4.12. *For each $i \in [Q]$ there are exactly $|B_i|$ good bristle assignments that satisfy I_v and are adjacent to σ_{p_i} in $H_{I_\mathrm{v},I_\mathrm{e}}$.*

Proof. Every good bristle assignment is of the form $\beta_i[a,b]$ for some $i \in [Q]$, $a \in K_{i-1} \setminus \{p_{i-1}\}$ and $b \in K_i \setminus \{p_i\}$. In the proof of Lemma 4.11 we have shown that the only neighbour of $\beta_i[a,b]$ is σ_{p_i}. Then the following claim completes the proof of the lemma:

Claim: $\beta_i[a,b]$ satisfies I_v if and only if $\mathrm{Imp}(x_b, x_a)$ is among the $|B_i|$ smallest elements of $A(i)$.

Proof of the claim: Since $a \in K_{i-1} \setminus \{p_{i-1}\}$ and $b \in K_i \setminus \{p_i\}$ we have $\mathrm{Imp}(x_b, x_a) \in A(i)$.

First consider the case where $\mathrm{Imp}(x_b, x_a)$ is one of the $|B_i|$ smallest elements of $A(i)$. From Observation 4.9 we know that $\beta_i[b,a]$ satisfies I_v if and only if for all $s, t \in V^* \setminus \{p_0\}$ with $a \leq s \leq p_i$ and $p_i < t \leq b$ it holds that $\mathrm{Imp}(x_t, x_s) \notin C_\mathrm{v}$. Note that each such $\mathrm{Imp}(x_t, x_s)$ is in $A(i)$ and by Definition 4.3, $\mathrm{Imp}(x_t, x_s) \preceq \mathrm{Imp}(x_b, x_a)$. Thus, $\mathrm{Imp}(x_t, x_s) \in D_v(i)$ and we obtain $\mathrm{Imp}(x_t, x_s) \notin C_\mathrm{v}$ as required.

Now consider the remaining case where $\mathrm{Imp}(x_b, x_a)$ is not among the the $|B_i|$ smallest elements of $A(i)$. Then, by our choice of $D_v(i)$, $\mathrm{Imp}(x_b, x_a) \notin D_v(i)$. Moreover, for all $k \neq i$, $\mathrm{Imp}(x_b, x_a) \notin D_v(k)$ since $\mathrm{Imp}(x_b, x_a) \in A(i)$, $D_v(k) \subseteq A(k)$ and $A(k) \cap A(i) = \emptyset$. Hence $\mathrm{Imp}(x_b, x_a) \in C_\mathrm{v}$ and consequently $\beta_i[b,a]$ does not satisfy I_v by Observation 4.9. *(End of the proof of the claim.)* □

Lemma 4.13. *Let $I_\mathrm{v} = (X, C_\mathrm{v})$ and $I_\mathrm{e} = (X, C_\mathrm{e})$. Then $H_{I_\mathrm{v},I_\mathrm{e}}$ is isomorphic to H.*

Proof. Here we collect the previous results. By Lemma 4.4 the path assignments satisfy I_v (and hence are vertices of $H_{I_\mathrm{v},I_\mathrm{e}}$). Lemma 4.6 shows that the subgraph of $H_{I_\mathrm{v},I_\mathrm{e}}$ induced by the path assignments is isomorphic to $H[V^*]$. By Lemma 4.10 all other satisfying assignments are good bristle assignments and by Lemma 4.11 each of these good bristle assignments has a unique neighbour, which is among $\sigma_{p_1}, \ldots, \sigma_{p_Q}$. Then Lemma 4.12 shows that there are exactly $|B_i|$ good bristle assignments adjacent to σ_{p_i}. □

We can now prove Theorem 1.12, which we re-state at this point for the convenience of the reader.

Theorem 1.12. *Let H be a graph in $\mathcal{H}_{\mathrm{BIS}}$. Then approximately counting retractions to H is* #BIS*-equivalent under approximation-preserving reductions.*

Proof. Let $H \in \mathcal{H}_{\mathrm{BIS}}$. The #BIS-easiness part of Theorem 1.12 follows directly from Lemmas 4.13 and 4.1. The #BIS-hardness part follows from $\#\textsc{Hom}(H) \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$ (Observation 1.18) together with the fact that $\#\mathrm{BIS} \leq_{\mathrm{AP}} \#\textsc{Hom}(H)$ for all connected graphs H other than reflexive cliques and irreflexive stars [62, Theorem 1]. □

In our running example we conclude by showing in Figure 4.3 how H is encoded as $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$. We have already demonstrated which assignments are satisfying and thus are vertices of $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$. Using (4.5) and (4.6) it is straightforward to verify (with some work) that each satisfying bristle assignment of the form $\beta_i[a,b]$ is in fact adjacent only to σ_{p_i}.

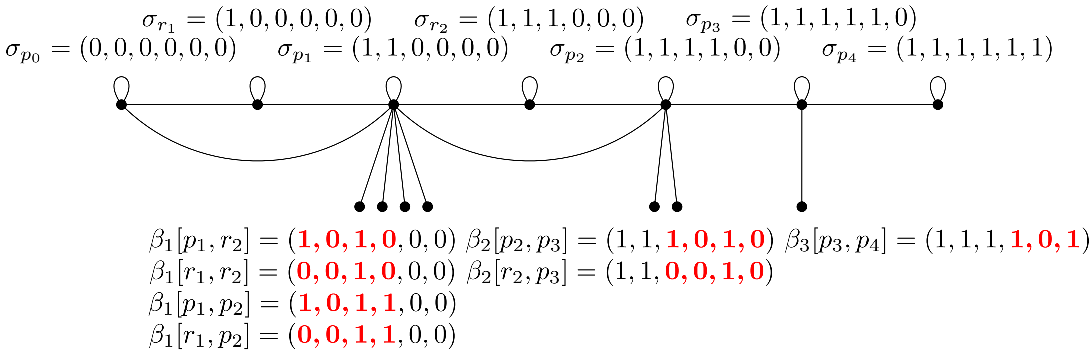


Figure 4.3: The graph $H_{I_{\mathrm{v}},I_{\mathrm{e}}}$. If a vertex corresponds to a satisfying assignment ρ of I_{v} then its label is of the form $\rho = (\rho(x_{r_1}), \rho(x_{p_1}), \rho(x_{r_2}), \rho(x_{p_2}), \rho(x_{p_3}), \rho(x_{p_4}))$.

4.3 #SAT-Hardness Results

From previous results we already know the following:

- Theorem 1.11 classifies the complexity of approximately counting retractions to H for all graphs H that are both square-free and triangle-free (i.e. have girth at least 5).
- For irreflexive H, the proof of Theorem 1.11 does not use triangle-freeness. This gives a trichotomy for approximately counting retractions to the class of irreflexive square-free graphs.

Therefore, we investigate square-free graphs that contain at least one triangle and at least one looped vertex. It turns out that we have to work through a number of technical cases to cover all #SAT-hard graphs with these properties (and hence all #SAT-hard square-free graphs).

For a positive integer q the graph WR_q is a looped star on $q+1$ vertices (the underlying star has q degree-1 vertices). The *net* is a looped triangle where each vertex of the triangle has an additional degree-2 neighbour. (The net is depicted in Section 4.3.5, Figure 4.15.) Here is an overview of the cases that we consider:

- In Section 4.3.2 we show that mixed triangles induce #SAT-hardness.
- In Section 4.3.3 we show in which cases the neighbourhood of a looped vertex induces #SAT-hardness.
- In Sections 4.3.4, 4.3.5 and 4.3.6 we show #SAT-hardness for square-free graphs with an induced WR_3, with an induced net and with an induced reflexive cycle of length at least 5, respectively. Essentially, these three sections deal with the graphs from the excluded subgraph characterisation of reflexive proper interval graphs (see [63, Section 1 and Appendix A] for the details about this characterisation).

4.3.1 Retractions and Neighbourhoods

Definition 4.14. For a graph H and a vertex $v \in V(H)$ we define the *(distance-1) neighbourhood* of v as $\Gamma_H(v) = \{u \in V(H) \mid \{u,v\} \in E(H)\}$. (In particular, this might include v itself.) Then $\deg_H(v) = |\Gamma_H(v)|$ is the *degree* of v. More generally, the *distance-k neighbourhood* of v is defined as $\Gamma_H^k(v) = \{u \in V(H) \mid$ There is a walk $W = u, w_1, \ldots, w_{k-1}, v$ (on k edges) in $H\}$. Let U be a subset of $V(H)$. Then $\Gamma_H(U) = \bigcap_{v \in U} \Gamma_H(v)$ is the *set of common neighbours* of the vertices in U.

The following well-known and simple observation shows that, for approximately counting retractions, hardness carries over from subgraphs that are induced by the neighbourhood of a vertex.

Observation 4.15. *Let H be a graph and let u be a vertex of H. Then $\#\textsc{Ret}(H[\Gamma_H(u)]) \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$.*

Proof. Let $(G, \mathbf{S})$ be an input to $\#\textsc{Ret}(H[\Gamma_H(u)])$, let $v_1, \ldots, v_n$ be the vertices of G and $\mathbf{S} = \{S_v \mid v \in V(G)\}$. Let w be a vertex distinct from the vertices in G. Then we construct the graph G' with vertices $V(G') = V(G) \cup \{w\}$ and edges $E(G') = E(G) \cup \{\{w, v_i\} \mid i \in [n]\}$. We set $\mathbf{S}' = \{S'_v \mid v \in V(G')\}$, where

$$S'_v = \begin{cases} \{u\}, & \text{if } v = w \\ S_v, & \text{if } v \in V(G) \text{ and } |S_v| = 1 \\ V(H), & \text{otherwise.} \end{cases}$$

Then $N\big((G, \mathbf{S}) \to H[\Gamma_H(u)]\big) = N\big((G', \mathbf{S}') \to H\big)$. □

4.3.2 Square-Free Graphs with Mixed Triangles

Lemma 4.16. *Let H be a square-free graph which contains a triangle with exactly two looped and one unlooped vertex. Then $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$.*

Proof. Let b_1, b_2, r be a triangle in H, where b_1 and b_2 are looped and r is unlooped. Consider the neighbourhood $\Gamma_H(b_1) \cap \Gamma_H(b_2)$. Since H is square-free, $H[\Gamma_H(b_1) \cap \Gamma_H(b_2)]$ is precisely the triangle b_1, b_2, r. The problem $\#\textsc{Hom}(H[\{b_1, b_2, r\}])$ corresponds to counting independent sets where vertices not in the independent set have a weight of 2 and vertices in the independent set have weight 1. It is well-known that approximately counting weighted independent sets is #SAT-hard, see for instance [108, Lemma 2.3]. This gives $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Hom}(H[\Gamma_H(b_1) \cap \Gamma_H(b_2)])$. From Observation 1.18 it follows immediately that $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H[\Gamma_H(b_1) \cap \Gamma_H(b_2)])$.

Finally, one can easily observe that $\#\textsc{Ret}(H[\Gamma_H(b_1) \cap \Gamma_H(b_2)]) \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$: Let $(G, \mathbf{S})$ be an input to $\#\textsc{Ret}(H[\Gamma_H(b_1) \cap \Gamma_H(b_2)])$ and let $\mathbf{S} = \{S_v \mid v \in V(G)\}$. Let w_1 and w_2 be vertices distinct from the vertices in G. Then we construct the graph G' with vertices $V(G') = V(G) \cup \{w_1, w_2\}$ and edges $E(G') = E(G) \cup \big(V(G) \times \{w_1, w_2\}\big)$. We set $\mathbf{S}' = \{S'_v \mid v \in V(G')\}$, where

$$S'_v = \begin{cases} \{b_1\}, & \text{if } v = w_1 \\ \{b_2\}, & \text{if } v = w_2 \\ S_v, & \text{if } v \in V(G) \text{ and } |S_v| = 1 \\ V(H), & \text{otherwise.} \end{cases}$$

Then $N\big((G, \mathbf{S}) \to H[\Gamma_H(b_1) \cap \Gamma_H(b_2)]\big) = N\big((G', \mathbf{S}') \to H\big)$. □

Lemma 4.17. *Let H be a square-free graph which contains a triangle with exactly two unlooped and one looped vertex. Then $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$.*

Proof. Let H_T be a triangle in H with vertices b, r_1 and r_2, where b is looped and both r_1 and r_2 are unlooped. Let $H' = H[\Gamma_H(b)]$. By Observation 4.15 it holds that $\#\textsc{Ret}(H') \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$. Suppose we can show that $\#\textsc{Ret}(H'[\{b, r_1\}]) \leq_{\mathrm{AP}} \#\textsc{Ret}(H')$. Then $H'[\{b, r_1\}]$ is a single edge with one looped (b) and one unlooped vertex (r_1) and it is well-known that counting homomorphisms to this graph corresponds to counting independent sets, which in turn is known to be #SAT-hard ([37, Theorem 3]). Summarising we have

$$\begin{aligned} \#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Hom}(H'[\{b, r_1\}]) &\leq_{\mathrm{AP}} \#\textsc{Ret}(H'[\{b, r_1\}]) \\ &\leq_{\mathrm{AP}} \#\textsc{Ret}(H') \leq_{\mathrm{AP}} \#\textsc{Ret}(H), \end{aligned}$$

where the second reduction is from Observation 1.18. This proves the lemma. It remains to prove the following claim.

Claim: $\#\mathbf{Ret}(H'[\{b, r_1\}]) \leq_{\mathrm{AP}} \#\mathbf{Ret}(H')$.

Proof of the claim: For $u \in V(H')$ let $w(u)$ be the number of common neighbours of r_1 and u in H'. Then $w(b) = 2$ since r_1 and b have two common neighbours: r_2 and b, and these are their only common neighbours as H' is square-free. Similarly, $w(r_1) = 2$ as the "common" neighbours in this case are simply the neighbours of r_1, which are only b and r_2 (since H' is square-free). Now let $u \in V(H') \setminus \{b, r_1\}$. The vertex b is a common neighbour of u and r_1 since every vertex in H' is a neighbour of b. It turns out that b is the only common neighbour of u and r_1: Suppose there exists a vertex $u' \neq b$ in H' which is a common neighbour of u and r_1. If $u' = u$ (see Figure 4.4 on the left) then u is adjacent to r_1. Additionally, u is then looped and hence $u \neq r_2$. Then

Figure 4.4: Contradictions to the square-freeness of the graph H'.

u, r_1, r_2, b is a square. If otherwise $u' \neq u$ then u, u', r_1, b is a square (see Figure 4.4 on the right), both cases give a contradiction. So we have shown that, for $u \in V(H')$,

$$w(u) = 2 \text{ if } u \in \{b, r_1\}, \text{ and } w(u) = 1 \text{ otherwise.} \tag{4.8}$$

Intuitively, we will use this fact to "boost" the vertices b and r_1 and make them exponentially more likely to be used by a homomorphism to H'.

Let q be the number of vertices of H'. Now let $(G, \mathbf{S})$ be an n-vertex input to $\#\textsc{Ret}(H'[\{b, r_1\}])$ and let ε be the desired precision. As usual, from $(G, \mathbf{S})$ we define an input $(G', \mathbf{S}')$ to $\#\textsc{Ret}(H')$. We introduce a vertex p distinct from the vertices of G that will serve as a pin to the vertex r_1 in H'. Then, for each $v \in V(G)$ we introduce an independent set on s vertices all of which are connected only to p and v. The parameter s will depend on the input size, specifically we set $s = n^2$. Intuitively it is clear that this gadget introduces a weight equal to $w(u)^s$ for each vertex $u \in V(H')$. For sufficiently large s, the image of v is likely to be b or r_1. This implies the statement of the lemma. The reader that trusts this intuition can skip reading the following calculations.

We give the full details for the sake of completeness: For each $v \in V(G)$, let I_v be an independent set of size s with vertices distinct from the remaining vertices of G'. Then G' is the graph with vertices $V(G') = V(G) \cup \{p\} \cup \bigcup_{v \in V(G)} I_v$ and edges $E(G') = E(G) \cup \bigcup_{v \in V(G)} \big(\{v, p\} \times I_v\big)$. We set $\mathbf{S}' = \{S'_v \mid v \in V(G')\}$, where

$$S'_v = \begin{cases} \{r_1\}, & \text{if } v = p \\ S_v, & \text{if } v \in V(G) \text{ and } |S_v| = 1 \\ V(H'), & \text{otherwise.} \end{cases}$$

We say that a homomorphism $h \in \mathcal{H}((G', \mathbf{S}'), H')$ is *full* if $h(V(G)) \subseteq \{b, r_1\}$. Let Z^* be the number of full homomorphisms from $(G', \mathbf{S}')$ to H'. Let Z_0 be the number of non-full homomorphisms from $(G', \mathbf{S}')$ to H'. Then

$$N\big((G', \mathbf{S}') \to H'\big) = Z^* + Z_0. \tag{4.9}$$

For $h \in \mathcal{H}((G,\mathbf{S}),H')$, let $Z(h)$ be the number of homomorphisms $h' \in \mathcal{H}((G',\mathbf{S}'),H')$ for which $h = h'|_{V(G)}$. By the construction of G', every vertex $v \in V(G)$ with $h(v) \in \{b, r_1\}$ contributes a factor of 2^s to $Z(h)$, whereas a vertex $v \in V(G)$ with $h(v) \notin \{b, r_1\}$ contributes a factor of 1 to $Z(h)$. It follows that

$$Z^* = \sum_{h\in\mathcal{H}((G,\mathbf{S}),H'),\ h \text{ full}} Z(h) = 2^{sn} \cdot N\big((G,\mathbf{S}) \to H'[\{b,r_1\}]\big), \tag{4.10}$$

and

$$Z_0 = \sum_{h\in\mathcal{H}((G,\mathbf{S}),H'),\ h \text{ non-full}} Z(h) \leq 2^{s(n-1)} \cdot N\big((G,\mathbf{S}) \to H'\big) \leq 2^{s(n-1)} \cdot q^n.$$

Therefore,

$$Z_0/2^{sn} \leq 2^{-s} \cdot q^n \leq 1/4, \tag{4.11}$$

where the last inequality holds for sufficiently large n by the choice $s = n^2$. Summarising, by (4.9) and (4.10), we have

$$N\big((G,\mathbf{S}) \to H'[\{b,r_1\}]\big) = \frac{Z^*}{2^{sn}} \leq \frac{N\big((G',\mathbf{S}') \to H'\big)}{2^{sn}}$$

and, using (4.9), (4.10) as well as (4.11), we obtain

$$\frac{N\big((G',\mathbf{S}') \to H'\big)}{2^{sn}} = \frac{Z^*}{2^{sn}} + \frac{Z_0}{2^{sn}} \leq N\big((G,\mathbf{S}) \to H'[\{b,r_1\}]\big) + 1/4.$$

Hence $N\big((G',\mathbf{S}') \to H'\big)/2^{sn} \in \big[N\big((G,\mathbf{S}) \to H'[\{b,r_1\}]\big), N\big((G,\mathbf{S}) \to H'[\{b,r_1\}]\big) + 1/4\big]$. Let Q be the solution returned by an oracle call to #Ret(H') with input $((G',\mathbf{S}'),\varepsilon/21)$, i.e. an approximation of $N\big((G',\mathbf{S}') \to H'\big)$. Then the output $\lfloor Q/2^{sn} \rfloor$ approximates $N\big((G,\mathbf{S}) \to H'[\{b,r_1\}]\big)$ with the desired precision as was shown in [37, Proof of Theorem 3]. *(End of the proof of the claim.)* □

4.3.3 Square-Free Neighbourhoods of a Looped Vertex

Now we consider graphs of the form $X(k_1,k_2,k_3)$ (see Figure 4.5). Why are we interested in these graphs? Let H be a square-free graph with a looped vertex b and let H not contain any mixed triangle as a subgraph. Then consider $H[\Gamma_H(b)]$, the graph induced by the neighbourhood of b. Since H does not contain mixed triangles, the unlooped neighbours of b do not have any neighbours in $H[\Gamma_H(b)]$ apart from b. Since H is square-free, $H[\Gamma_H(b)]$ is square-free as well and therefore a looped neighbour $u \neq b$ of b can have at most one additional neighbour apart from b and u itself (within $H[\Gamma_H(b)]$). It follows that $H[\Gamma_H(b)]$ is of the form $X(k_1,k_2,k_3)$. Note that $X(k_1,k_2,0)$ does not contain any cycles. Therefore the hardness results for graphs of this form come from the classification for graphs of girth at least 5 (Theorem 1.11). The remaining cases ($k_3 \geq 1$) are covered in this chapter. As an overview in advance, we will obtain #SAT $\leq_{\mathrm{AP}}$ #Ret($X(k_1,k_2,k_3)$) in the following cases (where in most cases we actually show the stronger result #SAT $\leq_{\mathrm{AP}}$ #Hom($X(k_1,k_2,k_3)$) — #SAT-hardness for #Ret($X(k_1,k_2,k_3)$) then follows from Observation 1.18.):

- $k_3 = 0$ and
  - $k_2 = 0$ and $k_1 \geq 1$ (Theorem 1.11)
  - $k_2 = 1$ and $k_1 \geq 1$ (Theorem 1.11)
  - $k_2 = 2$ and $k_1 \geq 2$ (Theorem 1.11)
- $k_3 = 1$ and
  - $k_2 = 0$ and $k_1 \geq 1$ (Lemma 4.30)
  - $k_2 = 1$ and $k_1 \geq 3$ (Lemma 4.31)
- $k_3 = 2$, $k_2 = 0$ and $k_1 \geq 5$ (Lemma 4.32)
- $k_2 + k_3 \geq 3$ (Lemma 4.38)

Following the classification for graphs of the form $X(k_1, k_2, k_3)$ we give a hardness result (Lemma 4.33) which uses properties of the distance-2 neighbourhood of a looped vertex b in H.

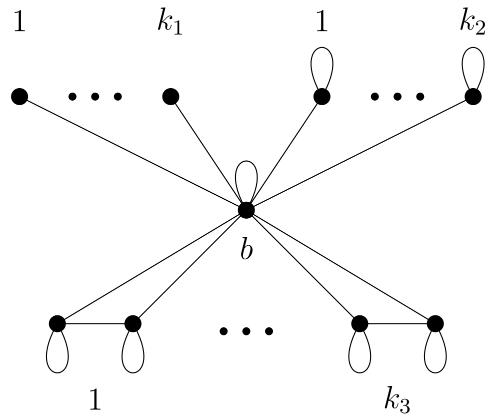


Figure 4.5: The graph $X(k_1, k_2, k_3)$.

A useful and well-known tool for proving hardness results for approximate counting problems are gadgets based on complete bipartite graphs where two states dominate (see, e.g., [37, Lemma 25], [78, Section 5] and [108, Lemma 5.1]). Let $F(H) = \{u \in V(H) \mid \Gamma_H(u) = V(H)\}$. One can use the described tool to show that, under certain conditions, a homomorphism from a complete bipartite graph to H will typically map one side to $F(H)$ and the other to $V(H)$. In this case it is then easy to reduce from counting independent sets to obtain #SAT-hardness. Formally, we use the version stated by Kelk [108]:

Lemma 4.18 ([108, Lemma 5.1])**.** *Let H be a graph with $\emptyset \subsetneq F(H) \subsetneq V(H)$. Suppose that, for every pair (S, T) with $\emptyset \subseteq S, T \subseteq V(H)$ satisfying $S \subseteq \Gamma_H(T)$ and $T \subseteq \Gamma_H(S)$, at least one of the following holds:*

(1) $S = F(H)$.

(2) $T = F(H)$.

(3) $|S| \cdot |T| < |F(H)| \cdot |V(H)|$.

Then $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Hom}(H)$.

In order to prove Lemmas 4.30 and 4.31 we will use Lemma 4.18 and, in addition, a reduction from the problem of counting large cuts, which is formally defined as follows: A *cut* of a graph G is a partition of $V(G)$ into two subsets (the order of this pair is ignored) and the size of a cut is the number of edges that have exactly one endpoint in each of these two subsets.

Name: #LargeCut.
Input: An integer $K \geq 1$ and a connected graph G in which every cut has size at most K.
Output: The number of size-K cuts in G.

The full details of the proof involve analysing different types of homomorphisms. The most important part of the results leading up to the proof of Lemmas 4.30 and 4.31 are the Tables 4.1 and 4.2, respectively. These tables show which types of homomorphisms represent a significant share of the overall number of homomorphisms that we are interested in. The crucial question is whether we can ensure that the right types of homomorphisms dominate this number. We desire two properties. First, the number of homomorphisms should be dominated by homomorphisms of two distinct types. Second, these two types should interact in an "anti-ferromagnetic" way.

We are going to use the graph $J(p, q, t)$ (see Figure 4.6) as a vertex gadget. This gadget was originally introduced in [37]. In general it is a good candidate when looking for gadgets to prove reductions from #LargeCut. Here is the formal definition.

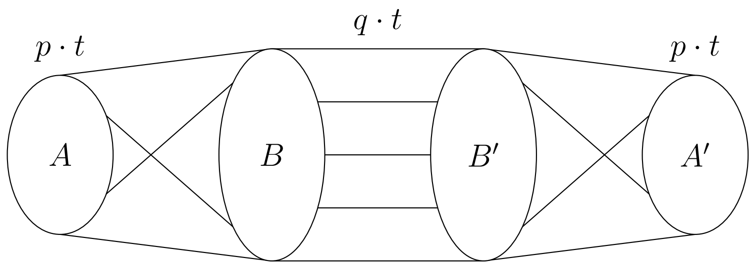


Figure 4.6: The graph $J(p, q, t)$.

Let p, q and t be positive integers. Let A and A' be independent sets of size $p \cdot t$ and let B and B' be independent sets of size $q \cdot t$. The set of edges M between B and B' forms a perfect matching. Then $J(p, q, t)$ is the graph for which the vertex set is the union of A, B, B' and A'. The edges are $(A \times B) \cup M \cup (B' \times A')$.

Let H be a graph and let h be a homomorphism from $J(p, q, t)$ to H. Let $h(B, B') = \{(h(u), h(v)) \mid u \in B, v \in B', \{u, v\} \in E(J(p, q, t))\}$. Intuitively, $h(B, B')$ contains (a directed version of) the edges of H that are "used" by h to cover the matching between B and B'. We say that h has *type* $(h(A), h(B, B'), h(A'))$. In general, a tuple $T = (T_1, T_2, T_3)$ is an H-*type* if $T_1, T_3 \subseteq V(H)$ and $T_2 \subseteq \{(x, y) \mid \{x, y\} \in E(H)\}$. Let $A(T) = T_1$, $B(T) = \{x \mid \exists y\, (x, y) \in T_2\}$, $B'(T) = \{y \mid \exists x\, (x, y) \in T_2\}$ and $A'(T) = T_3$.

Constraint. When we consider H-types (with respect to $J(p,q,t)$) in the following proofs we always formally require — without repeatedly stating it — that the parameter t is sufficiently large with respect to the number of vertices and edges in H, say $t \geq \max\{|V(H)|, 2|E(H)|\}$.

An H-type T is *non-empty* (with respect to $J(p,q,t)$) if there exists a homomorphism from $J(p,q,t)$ to H that has type T. Otherwise, T is called an *empty H-type*. From the definition of $J(p,q,t)$ we observe the following.

Observation 4.19. *Let H be a graph. An H-type T is non-empty if and only if*

(1) T_1, T_2 and T_3 are non-empty,

(2) $T_1 \times B(T) \subseteq E(H)$, and

(3) $B'(T) \times T_3 \subseteq E(H)$.

Let T and T' be H-types. We write $T \subseteq T'$ if, for $i \in [3]$, we have $T_i \subseteq T'_i$. An H-type T is *maximal* if it is non-empty and every H-type T' with $T' \neq T$, $T \subseteq T'$ is empty.

Lemma 4.20. *Let H be a graph and let $T = (T_1, T_2, T_3)$ be a maximal H-type. Then T is completely defined by $B(T)$ and $B'(T)$ since*

(1) $T_1 = \Gamma_H(B(T))$, $T_2 = E(B(T), B'(T))$ and $T_3 = \Gamma_H(B'(T))$.

(2) $B(T) = \Gamma_H(\Gamma_H(B(T)))$ and $B'(T) = \Gamma_H(\Gamma_H(B'(T)))$.

Proof. Since T is maximal it is also non-empty by definition.
Proof of (1)**:** We first show that $T_1 = \Gamma_H(B(T))$. Since T is non-empty, item (2) in Observation 4.19 implies that $T_1 \subseteq \Gamma_H(B(T))$. For the other direction consider the type $T^* = (\Gamma_H(B(T)), T_2, T_3)$. Clearly, $T \subseteq T^*$. Furthermore, from Observation 4.19 and the fact that T is non-empty, it follows that T^* is non-empty. Since T is maximal, we have $T = T^*$. Analogously, we obtain that $T_3 = \Gamma_H(B'(T))$ (where we use item (3) of Observation 4.19 rather than item (2)).

Showing $T_2 = E(B(T), B'(T))$ is similar: From the definitions of $B(T)$ and $B(T')$ we directly obtain that $T_2 \subseteq E(B(T), B'(T))$. To prove the other direction consider the type $T^* = (T_1, E(B(T), B'(T)), T_3)$. Clearly, $T \subseteq T^*$. Furthermore, $T_1^* = T_1$, $B(T^*) = B(T)$, $B'(T^*) = B'(T)$ and $T_3^* = T_3$. Hence, by Observation 4.19 and the fact that T is non-empty, it holds that T^* is non-empty. Since T is maximal, we have $T = T^*$.
Proof of (2)**:** Clearly, $B(T) \subseteq \Gamma_H(\Gamma_H(B(T)))$. To prove the other direction consider the type $T^* = (T_1, E(\Gamma_H(\Gamma_H(B(T))), \Gamma_H(\Gamma_H(B'(T)))), T_3)$. Since we have shown that $T_2 = E(B(T), B'(T))$, it is clear that $T \subseteq T^*$. In order to show that T^* is non-empty we verify the properties stated in Observation 4.19. T_1^*, T_2^* and T_3^* are non-empty by the fact that $T \subseteq T^*$ and T is non-empty. We have already established that $T_1 = \Gamma_H(B(T))$. Consequently, $T_1 \times \Gamma_H(\Gamma_H(B(T))) \subseteq E(H)$. Analogously, we obtain $\Gamma_H(\Gamma_H(B'(T))) \times T_3 \subseteq E(H)$, which shows that T^* is non-empty.

So, since T is maximal we have $T = T^*$. Then, $B(T) = B(T^*) = \Gamma_H(\Gamma_H(B(T)))$ and $B'(T) = B'(T^*) = \Gamma_H(\Gamma_H(B'(T)))$. □

Given an H-type $T = (T_1, T_2, T_3)$ we define $N(T)$ to be the number of homomorphisms in $\mathcal{H}(J(p,q,t), H)$ that have type T. We also set $\widehat{N}(T) = |T_1|^{pt}|T_2|^{qt}|T_3|^{pt}$. In Lemma 4.23, we will show that, for non-empty T, $\widehat{N}(T)$ is a close approximation to $N(T)$. In order to show this we need the following preliminaries.

Definition 4.21. Let $\left\{{a \atop b}\right\}$ be the number of surjective functions from a set of a elements to a set of b elements, i.e., $b! \cdot S(a,b)$, where $S(a,b)$ is the Stirling number of the second kind.[1]

Lemma 4.22 ([37, Lemma 18])**.** *If a and b are positive integers and $a \geq 2b \ln b$, then*

$$b^a\left(1 - \exp\left(-\frac{a}{2b}\right)\right) \leq \left\{{a \atop b}\right\} \leq b^a.$$

Lemma 4.23. *Let H be a graph. Let p and q be positive integers. There exists a positive integer t_0 such that for all $t \geq t_0$ and all H-types T that are non-empty with respect to $J(p,q,t)$, it holds that*

$$\frac{\widehat{N}(T)}{2} \leq N(T) \leq \widehat{N}(T).$$

Proof. Let $T = (T_1, T_2, T_3)$ be a non-empty type. Then

$$N(T) = \left\{{p \cdot t \atop |T_1|}\right\} \cdot \left\{{q \cdot t \atop |T_2|}\right\} \cdot \left\{{p \cdot t \atop |T_3|}\right\}. \tag{4.12}$$

For fixed p and q and sufficiently large t_0 we know from Lemma 4.22 that for all $t \geq t_0$ we have

$$1 - \exp\left(-\frac{p \cdot t}{2|T_1|}\right) \geq (1/2)^{1/3},$$

an analogous bound holds for the other two factors in Equation (4.12). The statement of the lemma then directly follows from Lemma 4.22. □

Lemma 4.24. *Let H be a connected graph with at least 2 vertices. Let T be a non-empty H-type that is not maximal. Then there exists a non-empty H-type T^* such that $\widehat{N}(T) \leq \left(\frac{2|E(H)|-1}{2|E(H)|}\right)^t \widehat{N}(T^*)$.*

Proof. Let $T = (T_1, T_2, T_3)$ be a non-empty type that is not maximal. Then there exists a non-empty type $T^* = (T_1^*, T_2^*, T_3^*)$ with $T^* \neq T$ and $T_i \subseteq T_i^*$ for $i \in [3]$. Since $T^* \neq T$ there exists an index $i \in [3]$ such that $T_i \subsetneq T_i^*$, i.e. $|T_i| \leq |T_i^*| - 1$. Then (using the fact that $p, q \geq 1$)

$$\frac{\widehat{N}(T)}{\widehat{N}(T^*)} = \frac{|T_1|^{pt}|T_2|^{qt}|T_3|^{pt}}{|T_1^*|^{pt}|T_2^*|^{qt}|T_3^*|^{pt}} \leq \left(\frac{|T_i^*| - 1}{|T_i^*|}\right)^t \leq \left(\frac{2|E(H)| - 1}{2|E(H)|}\right)^t,$$

where the last inequality holds since $|T_i^*| \leq 2|E(H)|$. □

Figure 4.7: The graphs $X(k_1, 0, 1)$ (on the left) and $X(k_1, 1, 1)$ (on the right).

Table 4.1: Maximal types of the homomorphisms from $J(p, q, t)$ to $X(k_1, 0, 1)$, where the vertices of $X(k_1, 0, 1)$ are labelled as in Figure 4.7 (on the left). Each line i corresponds to a type $T_i = \Big(A(T_i), E\big(B(T_i), B'(T_i)\big), A'(T_i)\Big)$. To shorten the notation we set $\mathcal{G} = \{g_j \mid j \in [k_1]\}$.

	$A(T)$	$B(T)$	$B'(T)$	$A'(T)$	$\widehat{N}(T)$
T_1	$\{r_1, r_2, b\} \cup \mathcal{G}$	$\{b\}$	$\{b\}$	$\{r_1, r_2, b\} \cup \mathcal{G}$	$(3 + k_1)^{pt} \cdot 1^{qt} \cdot (3 + k_1)^{pt}$
T_2	$\{r_1, r_2, b\} \cup \mathcal{G}$	$\{b\}$	$\{r_1, r_2, b\}$	$\{r_1, r_2, b\}$	$(3 + k_1)^{pt} \cdot 3^{qt} \cdot 3^{pt}$
T_3	$\{r_1, r_2, b\} \cup \mathcal{G}$	$\{b\}$	$\{r_1, r_2, b\} \cup \mathcal{G}$	$\{b\}$	$(3 + k_1)^{pt} \cdot (3 + k_1)^{qt} \cdot 1^{pt}$
T_4	$\{r_1, r_2, b\}$	$\{r_1, r_2, b\}$	$\{r_1, r_2, b\}$	$\{r_1, r_2, b\}$	$3^{pt} \cdot 9^{qt} \cdot 3^{pt}$
T_5	$\{r_1, r_2, b\}$	$\{r_1, r_2, b\}$	$\{r_1, r_2, b\} \cup \mathcal{G}$	$\{b\}$	$3^{pt} \cdot (9 + k_1)^{qt} \cdot 1^{pt}$
T_6	$\{b\}$	$\{r_1, r_2, b\} \cup \mathcal{G}$	$\{r_1, r_2, b\} \cup \mathcal{G}$	$\{b\}$	$1^{pt} \cdot (9 + 2k_1)^{qt} \cdot 1^{pt}$

Let $T = \big(A(T), E(B(T), B'(T)), A'(T)\big)$ be an H-type. Then we call T *symmetric* to the H-type $T' = \big(A'(T), E(B'(T), B(T)), A(T)\big)$. Clearly, $\widehat{N}(T) = \widehat{N}(T')$.

Lemma 4.25. *Let $H = X(k_1, 0, 1)$. Then all maximal H-types are listed in Table 4.1 (apart from those that are symmetric to a listed H-type). For each listed H-type T the last column of the table gives the corresponding value $\widehat{N}(T)$.*

Proof. Let $H = X(k_1, 0, 1)$ and let T be a maximal H-type. We claim that

$$B(T), B'(T) \in \{\{b\}, \{r_1, r_2, b\}, \{r_1, r_2, b, g_1, \dots, g_{k_1}\}\}$$

for the following reasons (we give the arguments for $B(T)$, they are identical for $B'(T)$):

- Since b is a neighbour of every vertex in H, from item (2) of Lemma 4.20 we obtain $b \in B(T)$.
- If, for some $i \in [k_1]$, we have $g_i \in B(T)$ then $\Gamma_H(B(T)) = \{b\}$ as b is the only neighbour of g_i. By item (2) of Lemma 4.20 it follows that $B(T) = \{r_1, r_2, b, g_1, \dots, g_{k_1}\}$.

[1]Note that in Chapters 2 and 3 we used $\left\{ {a \atop b} \right\}$ for the Stirling number itself.

- If $B(T) = \{r_1, b\}$, then $\Gamma_H(B(T)) = \{r_1, r_2, b\}$. By item (2) of Lemma 4.20 this gives $B(T) = \{r_1, r_2, b\}$, a contradiction.

- $B(T) = \{r_2, b\}$ gives a contradiction in the same way.

Table 4.1 then lists all possible combinations of sets $B(T)$ and $B'(T)$. From Lemma 4.20 it follows that these sets determine T completely (and $A(T), A'(T)$ are given accordingly). By definition, $T_1 = A(T)$ and $T_3 = A'(T)$. From item (1) of Lemma 4.20 it has to hold that $T_2 = E(B(T), B'(T))$. Then $\widehat{N}(T) = |T_1|^{pt}|T_2|^{qt}|T_3|^{pt}$ can be computed from the given sets in each row. □

Table 4.2: Maximal types of the homomorphisms from $J(p, q, t)$ to $X(k_1, 1, 1)$, where the vertices of $X(k_1, 1, 1)$ are labelled as in Figure 4.7 (on the right). Each line i corresponds to a type $T_i = \Big(A(T_i), E\big(B(T_i), B'(T_i)\big), A'(T_i)\Big)$. To shorten the notation we set $\mathcal{G} = \{g_j \mid j \in [k_1]\}$.

	$A(T)$	$B(T)$	$B'(T)$	$A'(T)$	$\widehat{N}(T)$
T_1	$\{r_1, r_2, b, c\} \cup \mathcal{G}$	$\{b\}$	$\{b\}$	$\{r_1, r_2, b, c\} \cup \mathcal{G}$	$(4 + k_1)^{pt} \cdot 1^{qt} \cdot (4 + k_1)^{pt}$
T_2	$\{r_1, r_2, b, c\} \cup \mathcal{G}$	$\{b\}$	$\{b, c\}$	$\{b, c\}$	$(4 + k_1)^{pt} \cdot 2^{qt} \cdot 2^{pt}$
T_3	$\{r_1, r_2, b, c\} \cup \mathcal{G}$	$\{b\}$	$\{r_1, r_2, b\}$	$\{r_1, r_2, b\}$	$(4 + k_1)^{pt} \cdot 3^{qt} \cdot 3^{pt}$
T_4	$\{r_1, r_2, b, c\} \cup \mathcal{G}$	$\{b\}$	$\{r_1, r_2, b, c\} \cup \mathcal{G}$	$\{b\}$	$(4 + k_1)^{pt} \cdot (4 + k_1)^{qt} \cdot 1^{pt}$
T_5	$\{b, c\}$	$\{b, c\}$	$\{b, c\}$	$\{b, c\}$	$2^{pt} \cdot 4^{qt} \cdot 2^{pt}$
T_6	$\{b, c\}$	$\{b, c\}$	$\{r_1, r_2, b\}$	$\{r_1, r_2, b\}$	$2^{pt} \cdot 4^{qt} \cdot 3^{pt}$
T_7	$\{b, c\}$	$\{b, c\}$	$\{r_1, r_2, b, c\} \cup \mathcal{G}$	$\{b\}$	$2^{pt} \cdot (6 + k_1)^{qt} \cdot 1^{pt}$
T_8	$\{r_1, r_2, b\}$	$\{r_1, r_2, b\}$	$\{r_1, r_2, b\}$	$\{r_1, r_2, b\}$	$3^{pt} \cdot 9^{qt} \cdot 3^{pt}$
T_9	$\{r_1, r_2, b\}$	$\{r_1, r_2, b\}$	$\{r_1, r_2, b, c\} \cup \mathcal{G}$	$\{b\}$	$3^{pt} \cdot (10 + k_1)^{qt} \cdot 1^{pt}$
T_{10}	$\{b\}$	$\{r_1, r_2, b, c\} \cup \mathcal{G}$	$\{r_1, r_2, b, c\} \cup \mathcal{G}$	$\{b\}$	$1^{pt} \cdot (12 + 2k_1)^{qt} \cdot 1^{pt}$

Lemma 4.26. *Let $H = X(k_1, 1, 1)$. Then all maximal H-types are listed in Table 4.2 (apart from those that are symmetric to a listed H-type). For each listed H-type T the last column of the table gives the corresponding value $\widehat{N}(T)$.*

Proof. Let $H = X(k_1, 1, 1)$ and let T be a maximal H-type. We claim that

$$B(T), B'(T) \in \{\{b\}, \{b, c\}, \{r_1, r_2, b\}, \{r_1, r_2, b, c, g_1, \ldots, g_{k_1}\}\}.$$

The remainder of the proof is analogous to that of Lemma 4.25 with only one additional argument:

- If $r_1 \in B(T)$ and $c \in B(T)$, then $\Gamma_H(B(T)) = \{b\}$. By item (2) of Lemma 4.20 it follows that $B(T) = \{r_1, r_2, b, c, g_1, \ldots, g_{k_1}\}$. The same is true if both $r_2 \in B(T)$ and $c \in B(T)$.

□

Lemma 4.27. *Let $k_1 \in [7]$. Consider the types T_i, $i \in [6]$ given by Table 4.1. Then there is a $\gamma \in (0,1)$ and positive integers p and q such that, for all $i \in [6]$, $i \neq 5$ and all positive integers t, we have $\widehat{N}(T_i) \leq \gamma^t \widehat{N}(T_5)$.*

Proof. Let

$$\mathcal{L} = \left\{ \frac{\log\left(\frac{(3+k_1)^2}{3}\right)}{\log(9+k_1)}, \frac{\log(3+k_1)}{\log\left(\frac{9+k_1}{3}\right)}, \frac{\log\left(\frac{3+k_1}{3}\right)}{\log\left(\frac{9+k_1}{3+k_1}\right)}, \frac{\log 3}{\log\left(\frac{9+k_1}{9}\right)} \right\}$$

and

$$R = \frac{\log 3}{\log\left(\frac{9+2k_1}{9+k_1}\right)}.$$

For each of the seven possible values of k_1 we can check (for example by computer) that every member of $\mathcal{L}$ is less than R. Thus, we can choose p and q so that

$$\forall L \in \mathcal{L},\ L < \frac{q}{p} < R. \tag{4.13}$$

We check the sought-for bound for each $i \in [6]$, $i \neq 5$:

T_1: $\frac{\widehat{N}(T_1)}{\widehat{N}(T_5)} = ((3+k_1)^2/3)^{pt}(1/(9+k_1))^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $((3+k_1)^2/3)^p < (9+k_1)^q$ which is equivalent to $\log\big((3+k_1)^2/3\big)/\log(9+k_1) < q/p$. This is true by (4.13).

T_2: $\frac{\widehat{N}(T_2)}{\widehat{N}(T_5)} = (3+k_1)^{pt}(3/(9+k_1))^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $(3+k_1)^p < ((9+k_1)/3)^q$. This is true by (4.13).

T_3: $\frac{\widehat{N}(T_3)}{\widehat{N}(T_5)} = ((3+k_1)/3)^{pt}((3+k_1)/(9+k_1))^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $((3+k_1)/3)^p < ((9+k_1)/(3+k_1))^q$. This is true by (4.13).

T_4: $\frac{\widehat{N}(T_4)}{\widehat{N}(T_5)} = 3^{pt}(9/(9+k_1))^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $3^p < ((9+k_1)/9)^q$. This is true by (4.13).

T_6: $\frac{\widehat{N}(T_6)}{\widehat{N}(T_5)} = (1/3)^{pt}((9+2k_1)/(9+k_1))^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $((9+2k_1)/(9+k_1))^q < 3^p$. This is true by (4.13).

□

Lemma 4.28. *Let $k_1 \in \{3,4,5,6\}$. Consider the types T_i, $i \in [10]$ given by Table 4.2. Then there is a $\gamma \in (0,1)$ and positive integers p and q such that, for all $i \in [10]$, $i \neq 9$ and all positive integers t, we have $\widehat{N}(T_i) \leq \gamma^t \widehat{N}(T_9)$.*

Proof. Let

$$\mathcal{L} = \left\{ \frac{\log\left(\frac{(4+k_1)^2}{3}\right)}{\log(10+k_1)}, \frac{\log(4+k_1)}{\log\left(\frac{10+k_1}{3}\right)}, \frac{\log\left(\frac{4+k_1}{3}\right)}{\log\left(\frac{10+k_1}{4+k_1}\right)}, \frac{\log 3}{\log\left(\frac{10+k_1}{9}\right)} \right\}$$

and

$$R = \frac{\log 3}{\log\left(\frac{12+2k_1}{10+k_1}\right)}.$$

For each of the four possible values of k_1 we can check (for example by computer) that every member of $\mathcal{L}$ is less than R. Thus, we can choose p and q so that

$$\forall L \in \mathcal{L},\ L < \frac{q}{p} < R. \tag{4.14}$$

Suppose that T and T' are types listed in Table 4.2 which are distinct from T_9 and have the property that $\widehat{N}(T') \leq \widehat{N}(T)$ for all $k_1 \in \{3,4,5,6\}$. Then the sought-for bound automatically holds for T' if it holds for T.

We check the sought-for bound for each $i \in [10]$, $i \neq 9$:

T_1: $\frac{\widehat{N}(T_1)}{\widehat{N}(T_9)} = ((4+k_1)^2/3)^{pt}(1/(10+k_1))^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $((4+k_1)^2/3)^p < (10+k_1)^q$. This is true by (4.14).

T_2: $\widehat{N}(T_2) \leq \widehat{N}(T_3)$.

T_3: $\frac{\widehat{N}(T_3)}{\widehat{N}(T_9)} = (4+k_1)^{pt}(3/(10+k_1))^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $(4+k_1)^p < ((10+k_1)/3)^q$. This is true by (4.14).

T_4: $\frac{\widehat{N}(T_4)}{\widehat{N}(T_9)} = ((4+k_1)/3)^{pt}((4+k_1)/(10+k_1))^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $((4+k_1)/3)^p < ((10+k_1)/(4+k_1))^q$. This is true by (4.14).

T_5: $\widehat{N}(T_5) \leq \widehat{N}(T_8)$.

T_6: $\widehat{N}(T_6) \leq \widehat{N}(T_8)$, for all $k \in \{3,4,5,6\}$.

T_7: $\frac{\widehat{N}(T_7)}{\widehat{N}(T_9)} = (2/3)^{pt}((6+k_1)/(10+k_1))^{qt} < \gamma^t$ is fulfilled for $2/3 \cdot (6+k_1)/(10+k_1) < \gamma < 1$.

T_8: $\frac{\widehat{N}(T_8)}{\widehat{N}(T_9)} = 3^{pt}(9/(10+k_1))^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $3^p < ((10+k_1)/9)^q$. This is true by (4.14).

T_{10}: $\frac{\widehat{N}(T_{10})}{\widehat{N}(T_9)} = (1/3)^{pt}((12+2k_1)/(10+k_1))^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $((12+2k_1)/(10+k_1))^q < 3^p$. This is true by (4.14).

□

Remark 4.29. We point out that the proofs of Lemmas 4.27 and 4.28 break for larger k_1. (For larger k_1 there exists some lower bound on p/q which exceeds some upper bound on that ratio.) Lemma 4.28 also breaks for $k_1 = 1$ and $k_1 = 2$. This matches the results from Section 4.2 which show that approximately counting retractions to $X(k_1, 1, 1)$ is actually #BIS-easy for these values of k_1.

Lemma 4.30. *If $k_1 \geq 1$, then* $\#\text{SAT} \leq_{\text{AP}} \#\textsc{Hom}(X(k_1, 0, 1))$.

Proof. We make a case distinction depending on k_1. The first case is the main work of the proof and we use the dominance of the type T_5 from Table 4.1 for $k_1 \leq 6$ as shown in Lemma 4.27. The second case ($k_1 \geq 7$) then follows from Lemma 4.18.

Case 1: $\boldsymbol{k_1} \in [\mathbf{6}]$. We use a reduction from #LargeCut, which is known to be #SAT-hard (see [37]). Let G and K be an input to #LargeCut, n be the number of vertices of G and $\varepsilon \in (0, 1)$ be the parameter of the desired precision. To shorten notation let $H = X(k_1, 0, 1)$. From G we construct an input G' to $\#\textsc{Hom}(H)$ by introducing vertex and edge gadgets. We assume that the vertices of H are labelled as in Figure 4.7 (on the left).

Let p, q be positive integers that fulfil (4.13). Note that p and q only depend on k_1 which is a parameter of H and therefore does not depend on the input G. We define the parameter t of the gadget graph $J(p, q, t)$ to be $t = n^4$. We also define a new parameter $s = n + 2$.

For each vertex $v \in V(G)$ we introduce a vertex gadget G'_v which is a graph $J(p, q, t)$ as given in Figure 4.6. We denote the corresponding sets A, B, B', A' by A_v, B_v, B'_v and A'_v, respectively. We say that two gadgets G'_u and G'_v are adjacent if u and v are adjacent in G.

For every edge $e = \{u, v\} \in E(G)$ we introduce an edge gadget as follows. We introduce two size-s independent sets, denoted by S_e and S'_e. As shown in Figure 4.8 we construct the set of edges

$$E'_e = (B_u \times S_e) \cup (B'_u \times S'_e) \cup (B_v \times S'_e) \cup (B'_v \times S_e).$$

Putting the pieces together, G' is the graph with

$$V(G') = \bigcup_{v \in V(G)} V(G'_v) \cup \bigcup_{e \in E(G)} (S_e \cup S'_e) \quad \text{and} \quad E(G') = \bigcup_{v \in V(G)} E(G'_v) \cup \bigcup_{e \in E(G)} E'_e.$$

Let h be a homomorphism from G to H, v be some vertex of G and G'_v be the corresponding vertex gadget. Then $h|_{V(G'_v)}$ corresponds to a homomorphism from $J(p, q, t)$ to H and therefore has an H-type.

We say that a homomorphism from G' to H is *full* if its restriction to each vertex gadget is either of type T_5 (from Table 4.1) or of its symmetric type (let us call it T'_5). The cut corresponding to a full homomorphism h partitions $V(G)$ into those vertices v for which $h|_{G'_v}$ has type T_5 and those for which $h|_{G'_v}$ has type T'_5. We say that a full homomorphism is K*-large* if the size of the corresponding cut is equal to K, otherwise we say that the homomorphism is K*-small*. Consider a full homomorphism h from G' to H.

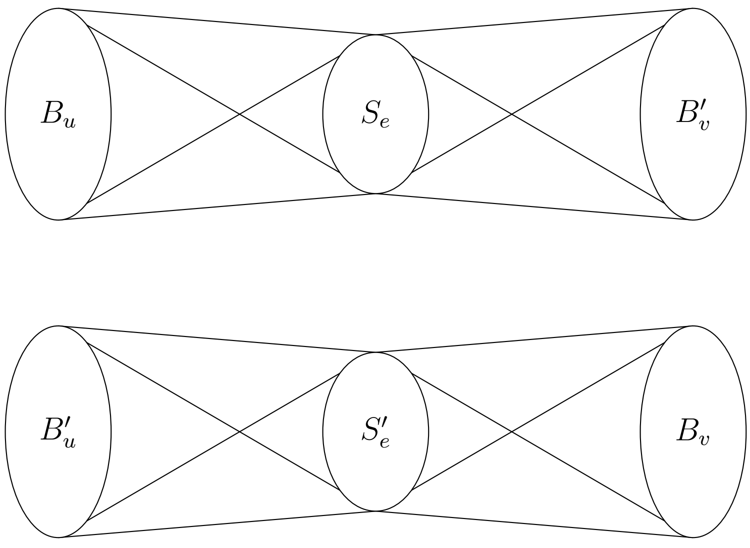


Figure 4.8: The edge gadget for the edge $e = \{u, v\}$.

- For an edge $e = \{u, v\}$ of G suppose $h|_{G'_u}$ has type T_5 and $h|_{G'_v}$ has type T'_5. Note that by the definition of the edge gadget, we have $h(S_e) \subseteq \Gamma_H(h(B_u)) \cap \Gamma_H(h(B'_v))$. Then the vertices in S_e can be mapped to any of $\{r_1, r_2, b\}$, whereas all vertices in S'_e have to be mapped to b (the sole common neighbour of r_1, r_2, b and the vertices in $\mathcal{G}$).

- Suppose instead that $h|_{G'_u}$ and $h|_{G'_v}$ have the same type T_5 or T'_5. Then the homomorphism h has to map the vertices in both S_e and S'_e to b.

Thus, every pair of adjacent gadgets of different types contributes a factor of 3^s to the number of full homomorphisms, whereas every pair of adjacent gadgets of the same type only contributes a factor of 1. Recall the definition of $N(T)$ as the number of homomorphisms from $J(p, q, t)$ to H that have type T. Then for $\ell \geq 1$ every size-ℓ cut of G arises in $2 \cdot N(T_5)^n \cdot 3^{s\ell}$ ways as a full homomorphism from G' to H.

Let L be the number of solutions to #LargeCut with input G and K (our goal is to approximate this number). We partition the homomorphisms from G' to H into three different sets. Z^* is the number of K-large (full) homomorphisms, Z_1 is the number of homomorphisms that are full but K-small and Z_2 is the number of non-full homomorphisms. Then we have $L = Z^*/(2N(T_5)^n 3^{sK})$ and $N\big(G' \to H\big) = Z^* + Z_1 + Z_2$. Thus it remains to show that $(Z_1 + Z_2)/(2N(T_5)^n 3^{sK}) \leq 1/4$ for our choice of p, q, t and s. Under this assumption we then have $N\big(G' \to H\big)/(2N(T_5)^n 3^{sK}) \in [L, L + 1/4]$ and a single oracle call to determine $N\big(G' \to H\big)$ with precision $\delta = \varepsilon/21$ suffices to determine L with the sought-for precision as demonstrated in [37].

Now we prove $(Z_1 + Z_2)/(2N(T_5)^n 3^{sK}) \leq 1/4$. As there are at most 2^n ways to assign a type T_5 or T'_5 to the n vertex gadgets in G' we have $Z_1 \leq 2^n \cdot N(T_5)^n \cdot 3^{s(K-1)}$. Then we obtain the following bound since $s = n + 2$:

$$\frac{Z_1}{2N(T_5)^n 3^{sK}} \leq \frac{2^n N(T_5)^n 3^{s(K-1)}}{2N(T_5)^n 3^{sK}} = \frac{2^n}{2 \cdot 3^s} \leq \frac{1}{8}.$$

We obtain a similar bound for Z_2: From Lemmas 4.24, 4.25 and 4.27 we know that for our choice of p and q there exists $\gamma \in (0,1)$ such that for every H-type T that is not T_5 or T_5' we have $\widehat{N}(T) \leq \gamma^t \widehat{N}(T_5)$. Using Lemma 4.23 this gives $N(T) \leq 2\gamma^t N(T_5)$ for sufficiently large t with respect to p, q and k_1 (which do not depend on the input G). Since $t = n^4$ we can assume that t is sufficiently large with respect to p and q as otherwise the input size is bounded by a constant (in which case we can solve #LargeCut in constant time).

For each H-type $T = (T_1, T_2, T_3)$, the cardinality of each set T_i is bounded above by $\max\{|V(H)|, 2|E(H)|\} = 12 + 2k_1$ and hence there are at most $\left(2^{12+2k_1}\right)^3$ different types. Furthermore, as H has $3 + k_1$ vertices, there are at most $(3+k_1)^{2sn^2}$ possible functions from the at most $2sn^2$ vertices in $\bigcup_{e \in E(G)}(S_e \cup S_e')$ to vertices in H. Since $t = n^4$ and $s = n+2$ we obtain

$$\begin{aligned} \frac{Z_2}{2N(T_5)^n 3^{sK}} &\leq \frac{\left(2^{12+2k_1}\right)^{3n} \cdot N(T_5)^{n-1} \cdot 2\gamma^t N(T_5) \cdot (3+k_1)^{2sn^2}}{2N(T_5)^n 3^{sK}} \\ &= \gamma^t \cdot \frac{\left(2^{12+2k_1}\right)^{3n} (3+k_1)^{2sn^2}}{3^{sK}} \leq \frac{1}{8}. \end{aligned}$$

The last inequality holds for sufficiently large n as

$$\frac{\left(2^{12+2k_1}\right)^{3n} (3+k_1)^{2sn^2}}{3^{sK}} \leq C^{n^3}$$

for some constant C that only depends on H, but not on the input G, whereas $t = n^4$.
(End of Case 1)

Case 2: $k_1 \geq 7$. We will show that in this case we can apply Lemma 4.18 to obtain #SAT $\leq_{\mathrm{AP}}$ #Hom(H). We have $F(H) = \{b\}$. Therefore, $|F(H)| \cdot |V(H)| = |V(H)| = 3 + k_1 \geq 10$. Let (S,T) be a pair with $\emptyset \subseteq S, T \subseteq V(H)$ satisfying $S \subseteq \Gamma_H(T)$, $T \subseteq \Gamma_H(S)$ and both $S \neq \{b\}$ and $T \neq \{b\}$ to meet the requirements of Lemma 4.18. We have to show that $|S| \cdot |T| < |F(H)| \cdot |V(H)| = 3 + k_1$. Note that for every vertex $u \neq b$ of $H = X(k_1, 0, 1)$ it holds that $|\Gamma_H(u)| \leq 3$. Therefore, $|S| \leq |\Gamma_H(T)| \leq 3$ and analogously $|T| \leq |\Gamma_H(S)| \leq 3$. Hence

$$|S| \cdot |T| \leq 9 < 10 \leq 3 + k_1.$$

(End of Case 2) □

Lemma 4.31. *If $k_1 \geq 3$, then* #SAT $\leq_{\mathrm{AP}}$ #Hom$(X(k_1,1,1))$.

Proof. This proof is very similar to the proof of Lemma 4.30. We give the details for the sake of completeness. As before, we make a case distinction depending on k_1. The first case is about the dominance of the type T_9 from Table 4.2 for $k_1 \in \{3,4,5,6\}$ as shown in Lemma 4.28. Otherwise, we use Lemma 4.18.

Case 1: $k_1 \in \{3,4,5,6\}$. We use a reduction from #LargeCut (#SAT-hard by [37]). Let G and K be an input to #LargeCut, n be the number of vertices of

G and $\varepsilon \in (0,1)$ be the parameter of the desired precision. To shorten notation let $H = X(k_1, 1, 1)$. We assume that the vertices of H are labelled as in Figure 4.7 (on the right).

Let p, q be positive integers that fulfil (4.14). Note that p and q only depend on k_1 which is a parameter of H and therefore does not depend on the input G. We will define the parameter t of the gadget graph J to be $t = n^4$. We also define a new parameter $s = n + 2$.

For each vertex $v \in V(G)$ we introduce a vertex gadget G'_v which is a graph $J(p, q, t)$ as given in Figure 4.6. We denote the corresponding sets A, B, B', A' by A_v, B_v, B'_v and A'_v, respectively. We say that two gadgets G'_u and G'_v are adjacent if u and v are adjacent in G. For every edge $e = \{u, v\} \in E(G)$ we use exactly the same edge gadget as in the proof of Lemma 4.30 (see Figure 4.8). G' is the graph with

$$V(G') = \bigcup_{v \in V(G)} V(G'_v) \cup \bigcup_{e \in E(G)} (S_e \cup S'_e) \quad \text{and} \quad E(G') = \bigcup_{v \in V(G)} E(G'_v) \cup \bigcup_{e \in E(G)} E'_e.$$

Let h be a homomorphism from G to H, v be some vertex of G and G'_v be the corresponding vertex gadget. Then $h|_{V(G'_v)}$ corresponds to a homomorphism from $J(p, q, t)$ to H and therefore has an H-type.

We say that a homomorphism from G' to H is *full* if its restriction to each vertex gadget is either of type T_9 (from Table 4.2) or of its symmetric type (let us call it T'_9). The cut corresponding to a full homomorphism h partitions $V(G)$ into those vertices v for which $h|_{G'_v}$ has type T_9 and those for which $h|_{G'_v}$ has type T'_9. We say that a full homomorphism is *K-large* if the size of the corresponding cut is equal to K, otherwise we say that the homomorphism is *K-small*. Consider a full homomorphism h from G' to H.

- For an edge $e = \{u, v\}$ of G suppose $h|_{G'_u}$ has type T_9 and $h|_{G'_v}$ has type T'_9. Note that by the definition of the edge gadget, we have $h(S_e) \subseteq \Gamma_H(h(B_u)) \cap \Gamma_H(h(B'_v))$. Then the vertices in S_e can be mapped to any of $\{r_1, r_2, b\}$, whereas all vertices in S'_e have to be mapped to b (the sole common neighbour of r_1, r_2, b, c and the vertices in $\mathcal{G}$).
- Suppose instead that $h|_{G'_u}$ and $h|_{G'_v}$ have the same type T_9 or T'_9. Then the homomorphism h has to map the vertices in both S_e and S'_e to b.

Thus, every pair of adjacent gadgets of different types contributes a factor of 3^s to the number of full homomorphisms, whereas every pair of adjacent gadgets of the same type only contributes a factor of 1. Recall the definition of $N(T)$ as the number of homomorphisms from $J(p, q, t)$ to H that have type T. Then for $\ell \geq 1$ every size-ℓ cut of G arises in $2 \cdot N(T_9)^n \cdot 3^{s\ell}$ ways as a full homomorphism from G' to H.

Let L be the number of solutions to #LargeCut with input G and K (our goal is to approximate this number). We partition the homomorphisms from G' to H into three different sets. Z^* is the number of K-large (full) homomorphisms, Z_1 is the number of homomorphisms that are full but K-small and Z_2 is the number of non-full homomorphisms. Then we have $L = Z^*/(2N(T_9)^n 3^{sK})$ and $N\big(G' \to H\big) = Z^* + Z_1 + Z_2$.

Thus it remains to show that $(Z_1 + Z_2)/(2N(T_9)^n 3^{sK}) \leq 1/4$ for our choice of p, q, t and s. Under this assumption we then have $N\big(G' \to H\big)/(2N(T_9)^n 3^{sK}) \in [L, L+1/4]$ and a single oracle call to determine $N\big(G' \to H\big)$ with precision $\delta = \varepsilon/21$ suffices to determine L with the sought-for precision as demonstrated in [37].

Now we prove $(Z_1 + Z_2)/(2N(T_9)^n 3^{sK}) \leq 1/4$. As there are at most 2^n ways to assign a type T_9 or T_9' to the n vertex gadgets in G' we have $Z_1 \leq 2^n \cdot N(T_9)^n \cdot 3^{s(K-1)}$. Then we obtain the following bound since $s = n + 2$:

$$\frac{Z_1}{2N(T_9)^n 3^{sK}} \leq \frac{2^n N(T_9)^n 3^{s(K-1)}}{2N(T_9)^n 3^{sK}} = \frac{2^n}{2 \cdot 3^s} \leq \frac{1}{8}.$$

We obtain a similar bound for Z_2: From Lemmas 4.24, 4.26 and 4.28 we know that for our choice of p and q there exists $\gamma \in (0,1)$ such that for every H-type T that is not T_9 or T_9' we have $\widehat{N}(T) \leq \gamma^t \widehat{N}(T_9)$. Using Lemma 4.23 this gives $N(T) \leq 2\gamma^t N(T_9)$ for sufficiently large t with respect to p, q and k_1. Since $t = n^4$ we can assume that t is sufficiently large with respect to p and q as otherwise the input size is bounded by a constant (in which case we can solve #LargeCut in constant time).

For each H-type $T = (T_1, T_2, T_3)$, the cardinality of each set T_i is bounded above by $\max\{|V(H)|, 2|E(H)|\} = 16 + 2k_1$ and hence there are at most $\big(2^{16+2k_1}\big)^3$ different types. Furthermore, as H has $4 + k_1$ vertices, there are at most $(4 + k_1)^{2sn^2}$ possible functions from the at most $2sn^2$ vertices in $\bigcup_{e \in E(G)}(S_e \cup S_e')$ to vertices in H. Since $t = n^4$ and $s = n + 2$ we obtain

$$\begin{aligned}\frac{Z_2}{2N(T_9)^n 3^{sK}} &\leq \frac{\big(2^{16+2k_1}\big)^{3n} \cdot N(T_9)^{n-1} \cdot 2\gamma^t N(T_9) \cdot (4 + k_1)^{2sn^2}}{2N(T_9)^n 3^{sK}} \\ &= \gamma^t \cdot \frac{\big(2^{16+2k_1}\big)^{3n} (4 + k_1)^{2sn^2}}{3^{sK}} \leq \frac{1}{8}.\end{aligned}$$

The last inequality holds for sufficiently large n as

$$\frac{\big(2^{16+2k_1}\big)^{3n} (4 + k_1)^{2sn^2}}{3^{sK}} \leq C^{n^3}$$

for some constant C that only depends on H, but not on the input G, whereas $t = n^4$. **(End of Case 1)**

Case 2: $k_1 \geq 6$. We will show that in this case we can apply Lemma 4.18 to obtain #SAT $\leq_{\mathrm{AP}}$ #Hom(H). We have $F(H) = \{b\}$. Therefore, $|F(H)| \cdot |V(H)| = |V(H)| = 4 + k_1 \geq 10$. Let (S, T) be a pair with $\emptyset \subseteq S, T \subseteq V(H)$ satisfying $S \subseteq \Gamma_H(T)$, $T \subseteq \Gamma_H(S)$ and both $S \neq \{b\}$ and $T \neq \{b\}$ to meet the requirements of Lemma 4.18. We have to show that $|S| \cdot |T| < |F(H)| \cdot |V(H)| = 4 + k_1$. Note that for every vertex $u \neq b$ of $H = X(k_1, 1, 1)$ it holds that $|\Gamma_H(u)| \leq 3$. Therefore, $|S| \leq |\Gamma_H(T)| \leq 3$ and analogously $|T| \leq |\Gamma_H(S)| \leq 3$. Hence

$$|S| \cdot |T| \leq 9 < 10 \leq 4 + k_1.$$

(End of Case 2) □

Lemma 4.32. *If $k_1 \geq 5$, then* $\#\text{SAT} \leq_{\text{AP}} \#\text{Hom}(X(k_1, 0, 2))$.

Proof. Let $k_1 \geq 5$. To shorten notation let $H = X(k_1, 0, 2)$. Again we use Lemma 4.18 to obtain $\#\text{SAT} \leq_{\text{AP}} \#\text{Hom}(H)$. We have $F(H) = \{b\}$ and $|F(H)| \cdot |V(H)| = |V(H)| = 5 + k_1 \geq 10$. The remainder of the proof is identical to Case 2 in the proof of Lemma 4.31. □

Lemma 4.33. *Let H be a graph and $b \in V(H)$ be a looped vertex with an unlooped neighbour $g \in V(H)$. If $|\Gamma_H(g)| \geq 2$ (g has at least 2 neighbours in H) and for all $u \in \Gamma_H(b) \setminus \{g\}$ we have $|\Gamma_H(u) \cap \Gamma_H(g)| = 1$ (b is the only common neighbour of g and u), then* $\#\text{SAT} \leq_{\text{AP}} \#\text{Ret}(H)$.

Proof. Let H' be the graph obtained by replacing the vertex g in $H[\Gamma_H(b)]$ by an independent set I of size $|\Gamma_H(g)|^s$, where $s = 2\lceil \log_{|\Gamma_H(g)|}(|\Gamma_H(b)|) \rceil$. This is well-defined as $|\Gamma_H(g)| > 1$. The choice of s will become clear in a moment. Within the graph $H[\Gamma_H(b)]$, g is adjacent only to b by the assumption that $|\Gamma_H(b) \cap \Gamma_H(g)| = 1$. Therefore, each vertex in I shares an edge only with b. The transformation is depicted in Figure 4.9.

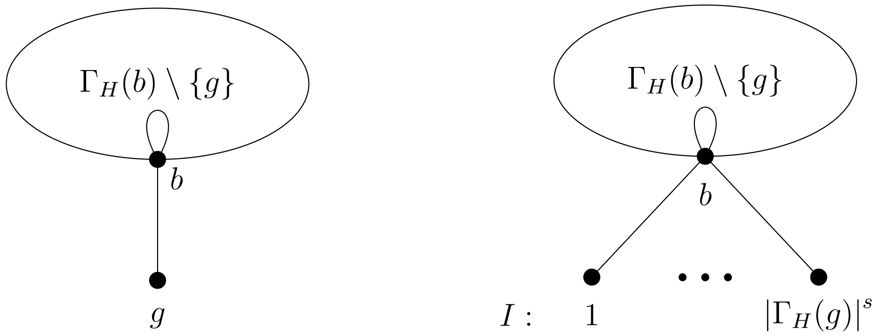


Figure 4.9: $H[\Gamma_H(b)]$ on the left and H' on the right.

First, we will show #SAT-hardness for $\#\text{Ret}(H')$ and we will apply Lemma 4.18 to achieve this. Let us check that the requirements are met. First note that $F(H') = \{b\}$. Now we have to show that, for every pair (S, T) with $\emptyset \subseteq S, T \subseteq V(H')$ satisfying $S \subseteq \Gamma_{H'}(T)$, $T \subseteq \Gamma_{H'}(S)$ and both $S \neq \{b\}$ and $T \neq \{b\}$ it holds that $|S| \cdot |T| < |F(H')| \cdot |V(H')| = |V(H')|$. First note that if there exists a vertex $u \in I \cap S$ then $T = \{b\}$ as b is the only neighbour of u in H'. Hence $I \cap S = \emptyset$. By the same reasoning it holds that $I \cap T = \emptyset$. Then $|S|, |T| \leq |\Gamma_{H'}(b) \setminus I| \leq |\Gamma_H(b)|$ and, by our choice $s = 2\lceil \log_{|\Gamma_H(g)|}(|\Gamma_H(b)|) \rceil$, we can conclude that

$$|S| \cdot |T| \leq |\Gamma_H(b)|^2 \leq |\Gamma_H(g)|^s < |V(H')| = |F(H')| \cdot |V(H')|.$$

This proves $\#\text{SAT} \leq_{\text{AP}} \#\text{Ret}(H')$.

To complete the prove of the lemma we show the following claim.

Claim: $\#\text{Ret}(H') \leq_{\text{AP}} \#\text{Ret}(H)$.

Proof of the claim:

Let $(G, \mathbf{S})$ be an input to $\#\text{RET}(H')$. Let w be a weight function on the vertices in $\Gamma_H(b)$ with $w(g) = |\Gamma_H(g)|^s$ and $w(u) = 1$ for all $u \in \Gamma_H(b) \setminus \{g\}$. By the construction of H' it is standard that

$$N\big((G, \mathbf{S}) \to H'\big) = \sum_{h \in \mathcal{H}((G,\mathbf{S}),H[\Gamma_H(b)])} \prod_{v \in V(G)} w(h(v)). \tag{4.15}$$

Now consider the vertices in $\Gamma_H(b)$. For $u \in \Gamma_H(b)$, let $w'(u)$ be the number of common neighbours of u and g in H. (It is essential that we regard all neighbours in H, not just the neighbours in $\Gamma_H(b)$.) By definition $w'(u) = |\Gamma_H(g)|$ if $u = g$ and, by the assumptions of this lemma, $w(u) = 1$ if $u \in \Gamma_H(b) \setminus \{g\}$. Hence, for all $u \in \Gamma_H(b)$ we have

$$w(u) = w'(u)^s. \tag{4.16}$$

Intuitively, we will use the fact that g has larger "weight" w' compared to the other vertices in $\Gamma_H(b)$ to "boost" the vertex g and make it more likely (by a factor of $|\Gamma_H(g)|^s$) to be used in a homomorphism to $H[\Gamma_H(b)]$.

Here are the details: Let $(G, \mathbf{S})$ be an input to $\#\text{RET}(H')$. We construct a graph G' from G in the following way. Let β and γ be vertices that are distinct from the vertices in G. Intuitively, β and γ will serve as "pins" to b and g, respectively. In addition, for each $v \in V(G)$, we introduce a independent set I_v of size s, see Figure 4.10. Then G' is the graph with vertices $V(G') = V(G) \cup \{\beta, \gamma\} \cup \bigcup_{v \in V(G)} I_v$ and edges $E(G') = E(G) \cup \big(V(G) \times \{\beta\}\big) \cup \bigcup_{v \in V(G)} \big(I_v \times \{v, \gamma\}\big)$.

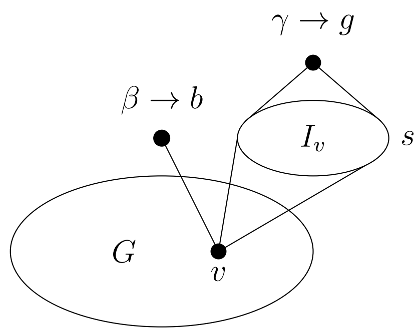


Figure 4.10: The vertex gadget used in the construction of G'.

Consider a homomorphism h from $(G', \mathbf{S}')$ to H. Then, since every vertex in $V(G)$ is a neighbour of β and $h(\beta) = b$, $h|_{V(G)}$ is a homomorphism from $(G, \mathbf{S})$ to $H[\Gamma_H(b)]$. Furthermore, by the construction of G', for each $v \in V(G)$ and each $u \in I_v$, it holds that $h(u)$ is a common neighbour of $h(v)$ and g in H. Recall that there are $w'(h(v))$ such common neighbours. Thus, using (4.15) and (4.16), we conclude

$$\begin{aligned} N\big((G', \mathbf{S}') \to H\big) &= \sum_{h \in \mathcal{H}((G,\mathbf{S}),H[\Gamma_H(b)])} \prod_{v \in V(G)} w'(h(v))^s \\ &= \sum_{h \in \mathcal{H}(G,H[\Gamma_H(b)])} \prod_{v \in V(G)} w(h(v)) \\ &= N\big((G, \mathbf{S}) \to H'\big). \end{aligned}$$

□

4.3.4 Square-Free Graphs with an Induced WR_3

This section can be seen as an extension of Section 4.3.3 as it essentially shows #SAT-hardness for graphs of the form $X(k_1, k_2, k_3)$ where $k_2 + k_3 \geq 3$. Consider a square-free graph H with an induced WR_3. Suppose that there is an induced WR_3 such that the neighbourhood of its center b does not contain any triangles. Then $H[\Gamma_H(b)]$ is subject to Theorem 1.11 which shows $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H[\Gamma_H(b)])$. Then, $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$ by Observation 4.15.

However, when considering square-free graphs as opposed to graphs of girth at least 5, $H[\Gamma_H(b)]$ might contain triangles. The smallest open case is displayed in Figure 4.11.

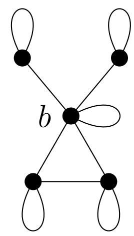


Figure 4.11: Smallest square-free graph with induced WR_3 for which it remains to prove hardness.

The goal of this section is to prove Lemma 4.38. We will use the following corollary of Lemma 4.22. (Note that here we assume $a \geq 2b \ln 2b$, rather than $a \geq 2b \ln b$.)

Corollary 4.34. *If a and b are positive integers with $a \geq 2b \ln 2b$, then*

$$\frac{b^a}{2} \leq \left\{ \begin{matrix} a \\ b \end{matrix} \right\} \leq b^a.$$

The proof of Lemma 4.38 uses the same general idea as the proof of [73, Theorem 2] — namely that approximating the partition function of the q-state ferromagnetic Potts model is #SAT-equivalent if $q \geq 3$ and, in addition, we are allowed to specify that certain vertices have to have a specific spin. Crucially, #SAT-hardness is known only if this single-vertex "pinning" is allowed. In general, the complexity of approximating the partition function of the Potts model is still unresolved and an important open problem. The approach of simulating ferromagnetic Potts with "pinning" to obtain hardness results has been used before, for instance in the proofs of [76, Lemma 3.6] and our Lemma 3.6. The gadgets we use here to accomplish the reduction are tailored to the specific problem and different from the gadgets used in similar reductions.

As in the proof of [73, Theorem 2] we use a reduction from the problem of counting so-called multiterminal cuts. We introduce the corresponding definitions from [73]. A *multiterminal cut* of a graph G with distinguished vertices $\tau_1, \dots, \tau_q$ (called *terminals*) is a set of edges $E' \subseteq E(G)$ that disconnects the terminals (i.e. ensures that there is no path in $(V(G), E(G) \setminus E')$ that connects any two distinct terminals). The *size* of a multiterminal cut is its cardinality. We consider the following computational problem.

Name: #MULTITERMINALCUT(q).
Input: A connected irreflexive graph G with q distinct terminals $\tau_1, \ldots, \tau_q \in V(G)$ and a positive integer K. The input has the property that every multiterminal cut has size at least K.
Output: The number of size-K multiterminal cuts of G with terminals $\tau_1, \ldots, \tau_q$.

Lemma 4.35 ([73, Section 4]). *Let $q \geq 3$. Then #MULTITERMINALCUT(q) $\equiv_{\mathrm{AP}}$ #SAT.*

Definition 4.36. Let $I = (G, \tau_1, \ldots, \tau_q, K)$ be an instance of #MULTITERMINALCUT(q). $\Phi(I) = \{\phi \colon V(G) \to [q] \mid \phi(\tau_i) = i,\ i \in [q]\}$ is the set of *separating functions* from $V(G)$ to $[q]$. For $\phi \in \Phi(I)$ let $\mathrm{Cut}(\phi) = \{\{u, v\} \in E(G) \mid \phi(u) \neq \phi(v)\}$ and, for $i \in [q]$, let $\mathrm{Mon}_i(\phi) = \{\{u, v\} \in E(G) \mid \phi(u) = \phi(v) = i\}$. Finally, let $\Phi^*(I) = \{\phi \in \Phi(I) \mid |\mathrm{Cut}(\phi)| = K\}$.

Observation 4.37. *Let $I = (G, \tau_1, \ldots, \tau_q, K)$ be an instance of #MULTITERMINALCUT(q). For each $\phi \in \Phi(I)$, $\mathrm{Cut}(\phi)$ is a multiterminal cut of I. On the other hand, each size-K multiterminal cut splits the graph G into exactly q connected components (as otherwise there would exist a multiterminal cut of size less than K). Hence each size-K multiterminal cut corresponds exactly to the function $\phi \in \Phi(I)$ for which $\phi(v) = i$ if v is in the same connected component as τ_i. Thus, $\Phi^*(I)$ is the subset of functions in $\Phi(I)$ that correspond to size-K multiterminal cuts. Let $T(I)$ be the number of size-K multiterminal cuts of the instance I. Then $T(I) = |\Phi^*(I)|$.*

Now we have all the tools at hand to prove the main lemma of this section.

Lemma 4.38. *Let H be a square-free graph. If H contains a* WR_3 *as an induced subgraph then #SAT $\leq_{\mathrm{AP}}$ #RET(H).*

Proof. Suppose that H contains a mixed triangle as an induced subgraph, then the statement of this lemma follows from Lemmas 4.16 and 4.17. Hence, for the remainder of this proof let H be a square-free graph with an induced WR_3 without any induced mixed triangle subgraphs. We choose a vertex b such that b is the center of an induced WR_3. We consider the graph $H[\Gamma_H(b)]$ which is the subgraph of H that is induced by the neighbourhood of b. For ease of notation we set $H_b = H[\Gamma_H(b)]$. Let U be the set of unlooped neighbours of b. Since H does not contain any mixed triangles, for each $u \in U$, b is the only neighbour of u in H_b. Since H is square-free, so is H_b. Therefore every looped neighbour $w \neq b$ of b has degree $\deg_{H_b}(w) \in \{2, 3\}$. By the choice of b, b has at least 4 neighbours including itself, i.e. we have $\deg_{H_b}(b) = \deg_H(b) \geq 4$. Let $x_1, \ldots, x_k$ be the looped neighbours of degree 2 and $x_{k+1}, y_{k+1}, \ldots, x_q, y_q$ be the looped neighbours of degree 3, where for each $i \in \{k+1, \ldots, q\}$ we have $\{x_i, y_i\} \in E(H_b)$ (where we use that looped vertices can only have looped neighbours since b is the sole neighbour of vertices in U). The graph H_b is depicted in Figure 4.12.

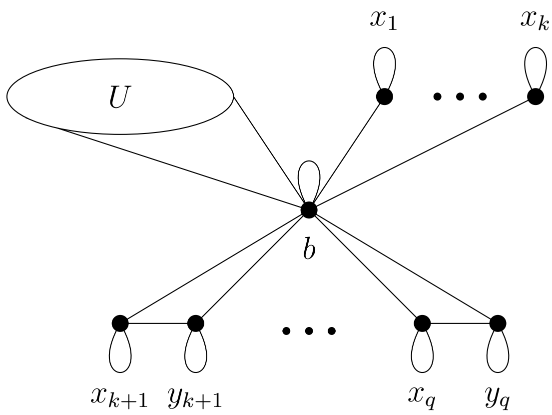


Figure 4.12: The graph H_b.

We will give a reduction from #MULTITERMINALCUT(q) to #RET(H_b). By the choice of b we have $q \geq 3$. This gives the desired reduction since

$$\#\text{SAT} \leq_{\text{AP}} \#\text{MULTITERMINALCUT(q)} \leq_{\text{AP}} \#\text{RET}(H_b) \leq_{\text{AP}} \#\text{RET}(H),$$

where the first reduction is from Lemma 4.35 and the last reduction is from Observation 4.15.

Let $I = (G, \tau_1, \ldots, \tau_q, K)$ be an instance of #MULTITERMINALCUT(q) and let $\varepsilon \in (0, 1)$ be the desired precision bound. Let $n = |V(G)|$ and $m = |E(G)|$. From the instance I we construct an instance $(J, \mathbf{S})$ of #RET(H_b). We will need some parameters whose relevance will become clear later in the proof. Let $s = n^5$ and $t = n^2$.

The intuition behind the gadgets that will be used in this proof is the following. For every vertex v in G we introduce a huge clique C_v. The image of such a clique under a homomorphism to H_b tends to be a reflexive clique, i.e. tends to be of the form $\Gamma_{H_b}(x_i)$. There are q such neighbourhoods. These will correspond to the q different states that a vertex $v \in V(G)$ can be in. We will have to add some attachments to the clique C_v to balance out the fact that $\Gamma_{H_b}(x_i)$ is a clique on 2 vertices if $i \leq k$, whereas it is a clique on 3 vertices if $i > k$. For each edge $\{u, v\} \in E(G)$ we introduce a gadget that favours the case where u and v have identical states (i.e. the corresponding cliques have the same image under homomorphisms to H_b).

Here are the details. First we define the graph J. We introduce q distinct vertices $p_1, \ldots, p_q$ which will serve as "pins" to the vertices $x_1, \ldots, x_q$. In the first part of the construction we will only use the vertices $p_1, \ldots, p_k$. For every vertex $v \in V(G)$ we introduce a graph J_v (the "vertex gadget") as follows. Let C_v be a clique on s vertices. For each vertex w in this clique we introduce k distinct vertices $\{w_1, \ldots, w_k\}$. Then J_v is the graph with vertices

$$V(J_v) = \{p_1, \ldots, p_k\} \cup V(C_v) \cup \bigcup_{w \in V(C_v)} \{w_1, \ldots, w_k\}$$

and edges

$$E(J_v) = E(C_v) \cup \bigcup_{w \in V(C_v)} \bigcup_{i \in [k]} \{\{w, w_i\}, \{w_i, p_i\}\}.$$

The graph J_v is depicted in Figure 4.13. Note that the vertices $p_1, \ldots p_k$ are identical over all vertex gadgets whereas the remaining vertices are distinct for each v.

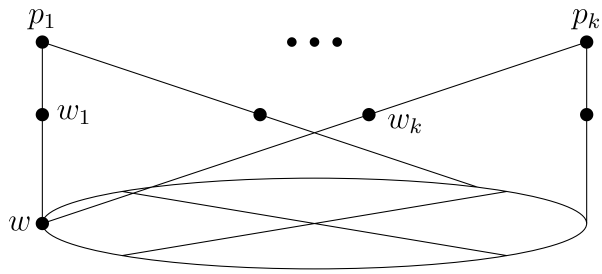


Figure 4.13: The graph J_v for a vertex v.

For every edge $e = \{u, v\} \in E(G)$ we introduce a graph J_e together with a set of edges E_e (the "edge gadget"). The graph J_e is defined in precisely the same way as J_v but uses the parameter t instead of s. We denote the corresponding clique by C_e. Further, we set $E_e = (V(C_u) \cup V(C_v)) \times V(C_e)$. The edge gadget is depicted in Figure 4.14.

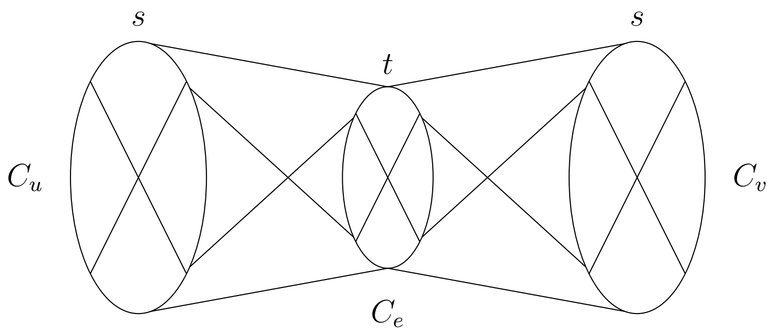


Figure 4.14: The edge gadget for an edge $e = \{u, v\}$. The edges to the vertices $\{w_1, \ldots, w_k \mid w \in C_u \cup C_e \cup C_v\}$ and $\{p_1, \ldots, p_k\}$ are omitted.

Finally, J is the graph with vertices

$$V(J) = \{p_{k+1}, \ldots, p_q\} \cup \bigcup_{v \in V(G)} V(J_v) \cup \bigcup_{e \in E(G)} V(J_e)$$

and edges

$$E(J) = \bigcup_{i \in [q]} \big(V(C_{\tau_i}) \times \{p_i\}\big) \cup \bigcup_{v \in V(G)} E(J_v) \cup \bigcup_{e \in E(G)} \big(E(J_e) \cup E_e\big).$$

Note that the first set in the union is a set of edges for each terminal τ_i. The purpose of these edges will be to ensure that the corresponding graph J_{τ_i} is in the right "state" (the one corresponding to i). Here we now use all of the p_i, not just the first k as we did in the construction of J_v and J_e.

Next we define the lists $\mathbf{S} = \{S_v \subseteq V(H_b) \mid v \in V(J)\}$. We set

$$S_v = \begin{cases} \{x_i\}, & \text{if } v = p_i,\ i \in [q] \\ V(H_b), & \text{otherwise.} \end{cases}$$

For a homomorphism $h \in \mathcal{H}((J, \mathbf{S}), H_b)$ and a vertex $v \in V(G)$ we say that the image $h(V(C_v))$ is the *state* of v (under h).

A *pinned configuration* is a tuple $(z, z_1, \ldots, z_k)$ of vertices of H_b such that, for each $i \in [k]$, $\{z, z_i\}$ and $\{z_i, x_i\}$ are edges of H_b. Note that the vertices $(w, w_1, \ldots, w_k)$ of C_v (see Figure 4.13) have to map to a pinned configuration under a homomorphism from $(J, \mathbf{S})$ to H_b. For $z \in V(H_b)$ let $f(z)$ be the number of pinned configurations $(z, z_1, \ldots, z_k)$. We have

$$f(b) = 2^k \qquad \text{(All } z_j \text{ can be either } x_j \text{ or } b.) \tag{4.17}$$

$$f(x_i) = 2\ (\forall i \in [k]) \qquad (z_i \text{ can be either } x_i \text{ or } b, \text{ all other } z_j \text{ have to be } b.) \tag{4.18}$$

$$f(x_i) = f(y_i) = 1\ (\forall i \in \{k+1, \ldots, q\}) \qquad \text{(All } z_j \text{ have to be } b.) \tag{4.19}$$

$$f(u) = 1\ (\forall u \in U) \qquad \text{(All } z_j \text{ have to be } b.) \tag{4.20}$$

We say that a vertex $v \in V(G)$ is *full* (under h) if the following conditions are met:

- There exists $i \in [q]$ such that $h(V(C_v)) = \Gamma_{H_b}(x_i)$.
- For every element $z \in h(V(C_v))$ and every pinned configuration $(z, z_1, \ldots, z_k)$ there is a vertex w in the clique C_v such that $h(w, w_1, \ldots, w_k) = (z, z_1, \ldots, z_k)$ (elementwise).

We call an edge $e = \{u, v\} \in E(G)$ *monochromatic* (under h) if u and v have the same state. Otherwise, we say that e is *dichromatic*. We say that the homomorphism h is *full* if every vertex $v \in V(G)$ is full under h. We say that a full homomorphism h is *K-small* if there are at most K dichromatic edges under h, otherwise we say that it is *K-large*.

Let Z^* be the number of full homomorphisms that are K-small. Further, let Z_1 be the number of full homomorphisms that are K-large and let Z_2 be the number of non-full homomorphisms. Then

$$N\big((J, \mathbf{S}) \to H_b\big) = Z^* + Z_1 + Z_2.$$

Let T be the sought-for number of size-K multiterminal cuts of the instance $I = (G, \tau_1, \ldots, \tau_q, K)$. We will now investigate the way in which the number Z^* relates to T. Recall the definitions about separating functions from Definition 4.36. In particular, we will use the sets $\Phi(I)$, $\Phi^*(I)$ and $\mathrm{Cut}(\phi)$. To shorten notation within the scope of this proof, we write Φ when we mean $\Phi(I)$ and Φ^* when we mean $\Phi^*(I)$. From Observation 4.37 we know that $T = |\Phi^*|$.

For a function $\phi \in \Phi$ we say that a homomorphism $h \in \mathcal{H}((J, \mathbf{S}), H_b)$ *agrees* with ϕ if, for each vertex v of G, the state of v under h is $h(V(C_v)) = \Gamma_{H_b}(x_{\phi(v)})$. Note that, by the construction of J, under a full homomorphism a terminal τ_i has state $\Gamma_{H_b}(x_i)$. Therefore, each full homomorphism h agrees with exactly one $\phi \in \Phi$.

If h agrees with ϕ, then it follows that $\mathrm{Cut}(\phi)$ is exactly the set of dichromatic edges under h. Hence, each K-small full homomorphism agrees with exactly one function $\phi \in \Phi^*$ and each K-large full homomorphism agrees with exactly one function $\phi \in \Phi \setminus \Phi^*$. Let Z_ϕ be the number of full homomorphisms that agree with $\phi \in \Phi$. Then

$$Z^* = \sum_{\phi \in \Phi^*} Z_\phi \quad \text{and} \quad Z_1 = \sum_{\phi \in \Phi \setminus \Phi^*} Z_\phi. \tag{4.21}$$

Let $\phi \in \Phi$. We are interested in the number Z_ϕ. What are the possible full homomorphisms h that agree with ϕ?

Observation A Let $v \in V(G)$. We consider possible images of the vertices of J_v. For h to agree with ϕ, the state of v is fixed to be $\Gamma_{H_b}(x_{\phi(v)})$, where this set can be either of the form $\{b, x_i\}$ or of the form $\{b, x_i, y_i\}$. From Equations (4.17) and (4.18) it follows that there are a total of $2^k + 2$ pinned configurations $(z, z_1, \ldots, z_k)$ with $z \in \{b, x_i\}$. Similarly, from Equations (4.17) and (4.19) it follows that there are a total of 2^k+1+1 pinned configurations with $z \in \{b, x_i, y_i\}$. As h is full, each possible pinned configuration has to be used at least once by the s vertices in C_v. As a consequence each vertex $v \in V(G)$ contributes a factor of $\left\{ {s \atop 2^k+2} \right\}$ to Z_ϕ.

Observation B Let $e = \{u, v\}$. What are the possible images of the vertices of J_e? We make a case distinction depending on e.

- Let $e = \{u, v\} \in \mathrm{Cut}(\phi)$. Then, as h is full, $h(V(C_u))$ and $h(V(C_v))$ are different states from the set $\{\Gamma_{H_b}(x_i) \mid i \in [q]\}$. By the definition of J we have that $h(V(C_e)) \subseteq h(V(C_u)) \cap h(V(C_v))$. It follows that $h(V(C_e)) = \{b\}$. There are 2^k pinned configurations with $z = b$. Each of the t vertices of C_e can have any of these 2^k pinned configurations. Therefore, e contributes a factor of 2^{kt} to Z_ϕ.
- Let $e = \{u, v\} \notin \mathrm{Cut}(\phi)$. Then, as h is full, $h(V(C_u)) = h(V(C_v)) = \Gamma_{H_b}(x_i)$ for some $i \in [q]$. Then $h(C_e) \subseteq \Gamma_{H_b}(x_i)$, where $\Gamma_{H_b}(x_i)$ is of the form $\{b, x_i\}$ or of the form $\{b, x_i, y_i\}$. As before there are $2^k + 2$ corresponding pinned configurations. Therefore, e contributes a factor of $(2^k + 2)^t$ to Z_ϕ.

Summarising, we obtain

$$Z_\phi = \left\{ {s \atop 2^k + 2} \right\}^n (2^k)^{t|\mathrm{Cut}(\phi)|} (2^k + 2)^{t(m - |\mathrm{Cut}(\phi)|)}. \tag{4.22}$$

For each $\phi \in \Phi^*$ we have $|\text{Cut}(\phi)| = K$. Then, using the fact that $|\Phi^*| = T$, we obtain

$$\begin{aligned} Z^* &= \sum_{\phi \in \Phi^*} Z_\phi \\ &= \sum_{\phi \in \Phi^*} \left\{ \begin{matrix} s \\ 2^k + 2 \end{matrix} \right\}^n (2^k)^{tK} (2^k+2)^{t(m-K)} \\ &= \left\{ \begin{matrix} s \\ 2^k + 2 \end{matrix} \right\}^n (2^k)^{tK} (2^k+2)^{t(m-K)} \cdot T. \end{aligned}$$

To shorten the notation let

$$L = \left\{ \begin{matrix} s \\ 2^k + 2 \end{matrix} \right\}^n (2^k)^{tK} (2^k+2)^{t(m-K)}.$$

We want to approximately compute the value T, where we have shown that $T = Z^*/L$. Assume for now that we have

$$N\big((J, \mathbf{S}) \to H_b\big)/L \in [Z^*/L, Z^*/L + 1/4] = [T, T + 1/4].$$

Then consider the algorithm which makes a $\#\text{RET}(H_b)$ oracle call with input $((J, \mathbf{S}), \varepsilon/21)$ to obtain a value Q and returns $\lfloor Q/L \rfloor$. This algorithm approximates T with the desired error bound ε as is shown in [37, Proof of Theorem 3]. It remains to show the following claim.

Claim: $\boldsymbol{N\big((J, \mathrm{S}) \to H_b\big)/L \in [Z^*/L, Z^*/L + 1/4]}$**.**

Proof of the claim: Recall that $N\big((J, \mathbf{S}) \to H_b\big) = Z^* + Z_1 + Z_2$. Clearly, we have $N\big((J, \mathbf{S}) \to H_b\big)/L \geq Z^*/L$. We will show $Z_1/L \leq 1/8$ and $Z_2/L \leq 1/8$ to prove $N\big((J, \mathbf{S}) \to H_b\big)/L \leq Z^*/L + 1/4$.

Recall that Z_1 is the number of K-large full homomorphisms. Using (4.21) and (4.22) and the fact that for each $\phi \in \Phi \setminus \Phi^*$ we have $|\text{Cut}(\phi)| \geq K + 1$ we obtain

$$\begin{aligned} Z_1 &\leq \sum_{\phi \in \Phi \setminus \Phi^*} \left\{ \begin{matrix} s \\ 2^k + 2 \end{matrix} \right\}^n (2^k)^{t(K+1)} (2^k+2)^{t(m-K-1)} \\ &\leq q^n \left\{ \begin{matrix} s \\ 2^k + 2 \end{matrix} \right\}^n (2^k)^{t(K+1)} (2^k+2)^{t(m-K-1)} \end{aligned}$$

where the last inequality follows from the fact that there are q^n functions in Φ. Then

$$\frac{Z_1}{L} \leq \left(\frac{2^k}{(2^k+2)} \right)^t q^n \leq 1/8,$$

where the last inequality holds for sufficiently large n by our choice of $t = n^2$ and the fact that $2^k/(2^k+2) < 1$.

Recall that Z_2 is the number of homomorphisms that are not full. How many non-full homomorphisms h are there? In general, there are at most $2^{|V(H_b)|}$ possible states

$h(V(C_v))$ for any vertex $v \in V(G)$. By the same arguments as given in Observation A, each full vertex under h contributes a factor of $\left\{{s \atop 2^k+2}\right\}$ to Z_2. Since $s = n^5$ the requirements of Corollary 4.34 are met for sufficiently large n and we obtain

$$(2^k+2)^s/2 \le \left\{{s \atop 2^k+2}\right\}. \tag{4.23}$$

We will use this bound shortly. If h is not full, there has to exist at least one vertex $v \in V(G)$ which is not full under h, and consequently there are at most $n-1$ full vertices under h.

Now assume that $v \in V(G)$ is not full under h. Then either v has state $h(V(C_v)) = \Gamma_{H_b}(x_i)$ for some $i \in [q]$ but not all of the 2^k+2 possible pinned configurations are used, or $h(V(C_v)) \notin \{\Gamma_{H_b}(x_i) \mid i \in [q]\}$ (which means that either $h(V(C_v)) \subsetneq \Gamma_{H_b}(x_i)$ or $h(V(C_v)) = \{b, u\}$ for some $u \in U$, by the fact that C_v is a large clique). In both cases the s vertices in C_v can each use at most 2^k+1 different pinned configurations. Hence, each non-full vertex contributes a factor of at most $(2^k+1)^s$ to Z_2. In particular this factor is smaller (for sufficiently large n) than the factor contributed by full vertices (see (4.23)). Finally, for each edge e there are at most $|V(H_b)|^{(k+1)t}$ mappings from the $(k+1)\cdot t$ vertices in $V(J_e) \setminus \{p_1, \ldots, p_k\}$ to $V(H_b)$. Therefore,

$$Z_2 \le 2^{|V(H_b)|n} \cdot \left\{{s \atop 2^k+2}\right\}^{n-1} \cdot (2^k+1)^s \cdot |V(H_b)|^{(k+1)tm}.$$

Recall that $L = \left\{{s \atop 2^k+2}\right\}^n (2^k)^{tK} (2^k+2)^{t(m-K)} \ge \left\{{s \atop 2^k+2}\right\}^n$. Then

$$\frac{Z_2}{L} \le \frac{2^{|V(H_b)|n} \cdot (2^k+1)^s \cdot |V(H_b)|^{(k+1)tm}}{\left\{{s \atop 2^k+2}\right\}} \le \left(\frac{2^k+1}{2^k+2}\right)^s \cdot 2 \cdot 2^{|V(H_b)|n} \cdot |V(H_b)|^{(k+1)n^4} \le \frac{1}{8},$$

where the second inequality follows from (4.23) and the last inequality holds for sufficiently large n by our choice of $s = n^5$. This proves the claim and completes the proof. *(End of the proof of the claim.)* □

4.3.5 Square-Free Graphs with an Induced Net

The goal of this section is to prove #SAT-hardness for square-free graphs with an induced net (see Figure 4.15). Note that the subgraphs of the net that are induced by a distance-1 neighbourhood of some vertex u of the net are of two forms. Either the corresponding subgraph is a looped edge (if $u \notin \{w_i \mid i \in [3]\}$) or it is isomorphic to a looped triangle where one vertex in the triangle has a single additional looped neighbour (if $u \in \{w_i \mid i \in [3]\}$). Approximately counting retractions to either of these two graphs is #BIS-easy (see Theorem 1.12). Therefore we cannot use these subgraphs in our hardness proof, so we need to work harder.

In our proof (Lemma 4.40) we use the same general approach that we used to prove Lemma 4.38. To make the approach work we have to find new gadgets tailored to the net. In one part of the reduction we will need to approximate real values by integers. To achieve this we use Dirichlet's approximation lemma, which has been used frequently in this line of research (see for instance [62]).

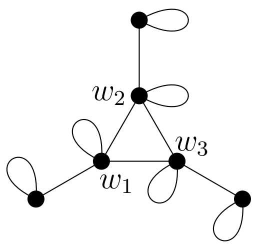


Figure 4.15: The *net*.

Lemma 4.39 ([131, p. 34]). *Let $\lambda_1, \ldots, \lambda_d > 0$ be real numbers and N be a natural number. Then there exist positive integers $t_1, \ldots, t_d, r$ with $r \leq N$ such that $|r\lambda_i - t_i| \leq 1/N^{1/d}$ for every $i \in [d]$.*

Lemma 4.40. *Let H be a square-free graph that has the net (as displayed in Figure 4.15) as an induced subgraph. Then #SAT $\leq_{\mathrm{AP}}$ #RET(H).*

Proof. Let H be a square-free graph with an induced net that is labelled as in Figure 4.15. Note that each of the vertices w_1, w_2 and w_3 might have additional neighbours in H. However, they cannot have further common neighbours as H is square-free. We use a reduction from #MULTITERMINALCUT(3) which is #SAT-hard under AP-reductions by Lemma 4.35. Let $I = (G, \tau_1, \tau_2, \tau_3, K)$ be an instance of #MULTITERMINALCUT(3) and $\varepsilon \in (0, 1)$ the desired precision bound. Let $n = |V(G)|$ and $m = |E(G)|$. We will construct an instance $(J, \mathbf{S})$ of #RET(H).

To construct this instance we will use a number of parameters which we introduce at this point. Let $s = n^2$. For $i \in [3]$ let $s_i = s \cdot \log_{|\Gamma_H(w_i)|/3} 2$. For $i \in [3]$, the value s_i is chosen such that

$$\left(\frac{|\Gamma_H(w_i)|}{3}\right)^{s_i} = 2^s. \tag{4.24}$$

It will become clear later in the proof why this is useful. (The important part is that the values $\left(\frac{|\Gamma_H(w_i)|}{3}\right)^{s_i}$, for $i \in [3]$, are identical and that the base of the exponent on the right-hand side is greater than 1.) We will now determine integers that approximate s_1, s_2 and s_3 using Dirichlet's approximation lemma. Let $\delta = \varepsilon/2$ and $\delta' = \log_{|V(H)|} e^{\delta}$ (the choices will become clear later in the proof). By Lemma 4.39 we obtain integers r, t_1, t_2 and t_3 of value at most $(m/\delta')^3 \in \mathrm{poly}(n, \varepsilon^{-1})$ such that $|rs_i - t_i| \leq \delta'/m$ for $i \in [3]$.

We go on to define the graph J. Intuitively, for $i \in [3]$, the terminal τ_i will serve as a "pin" to the vertex w_i. For each vertex $v \in V(G)$ we introduce a graph J_v which is simply a star with center v and leaves $\{\tau_1, \tau_2, \tau_3\}$. Formally, the vertices of J_v are $V(J_v) = \{v, \tau_1, \tau_2, \tau_3\}$. and the edges are $E(J_v) = \{v\} \times \{\tau_1, \tau_2, \tau_3\}$. For each edge $e = \{u, v\} \in E(G)$ we introduce a graph J_e which is defined as follows. For $i \in [3]$, let I_e^i be a disjoint independent set of size t_i. Let $t = \sum_{i \in [3]} t_i$ and let $I_e = \bigcup_{i \in [3]} I_e^i$ (I_e is an independent set of size t). Then J_e is depicted in Figure 4.16 and formally defined as the graph with vertices

$$V(J_e) = I_e \cup \{u, v, \tau_1, \tau_2, \tau_3\}$$

and edges

$$E(J_e) = (\{u, v\} \times I_e) \cup \bigcup_{i=1}^{3} \big(I_e^i \times \{\tau_i\}\big).$$

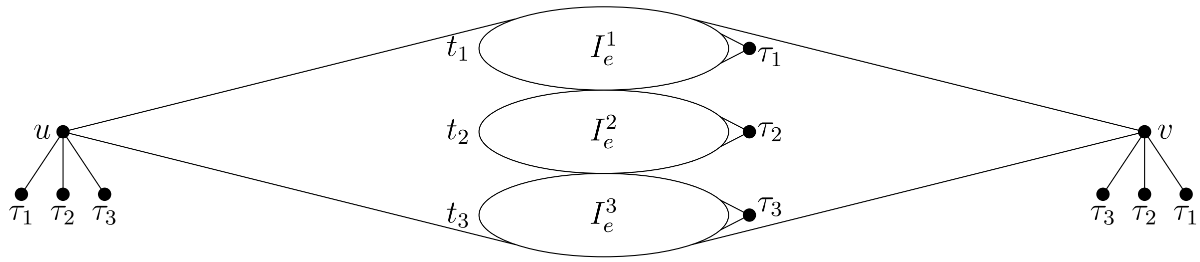


Figure 4.16: The graph J_e for an edge $e = \{u, v\}$.

Then J is the graph with vertices

$$V(J) = \bigcup_{v \in V(G)} V(J_v) \cup \bigcup_{e \in E(G)} V(J_e)$$

and edges

$$E(J) = \bigcup_{v \in V(G)} E(J_v) \cup \bigcup_{e \in E(G)} E(J_e).$$

The corresponding set of lists $\mathbf{S} = \{S_v \subseteq V(H) \mid v \in V(J)\}$ is defined by

$$S_v = \begin{cases} \{w_i\}, & \text{if } v = \tau_i,\ i \in [3] \\ V(H), & \text{otherwise.} \end{cases}$$

Let h be a homomorphism from $(J, \mathbf{S})$ to H. By the definition of J_v every vertex $v \in V(G)$ is also a vertex of J. An edge $e = \{u, v\} \in E(G)$ is called *monochromatic* under h if $h(u) = h(v)$. Otherwise, it is called *dichromatic* under h. We say that h is *K-small* if the number of dichromatic edges under h is at most K. Otherwise, h is called *K-large*. Let Z^* be the number of K-small homomorphisms from $(J, \mathbf{S})$ to H and let Z_1 be the number of K-large homomorphisms. Clearly,

$$N\big((J, \mathbf{S}) \to H\big) = Z^* + Z_1.$$

Recall the definitions of separating functions from Definition 4.36 and, to shorten notation, define the sets $\Phi = \Phi(I)$ and $\Phi^* = \Phi^*(I)$. T is the number of size-K multiterminal cuts of the instance $I = (G, \tau_1, \tau_2, \tau_3, K)$. Our goal is to approximate T. From Observation 4.37 we know that $T = |\Phi^*|$.

We say that a homomorphism $h \in \mathcal{H}((J, \mathbf{S}), H)$ *agrees* with $\phi \in \Phi$ if, for each $v \in V(G)$, we have $h(v) = w_{\phi(v)}$. By definition of the lists $\mathbf{S}$, for each $i \in [3]$, a homomorphism h from $(J, \mathbf{S})$ to H has to map τ_i to w_i. Furthermore, as v is adjacent to all three terminals and H is square-free we have $h(v) \in \{w_1, w_2, w_3\}$.

At this point we have introduced the gadget J_e and the graph J and have defined what it means for a homomorphism from J to H to agree with a function in Φ (which in turn corresponds to a multiterminal cut). All these definitions are heavily tailored to the graph H. The following steps, however, are very similar to those in the proof of Lemma 4.38. What complicates this proof in comparison to that of Lemma 4.38 is the fact that we need to use Dirichlet's approximation lemma to balance out the edge interactions. Here are the details.

Every homomorphism $h \in \mathcal{H}((J, \mathbf{S}), H)$ agrees with exactly one function $\phi \in \Phi$. In particular, if h agrees with ϕ, then the dichromatic edges of h are exactly the multiterminal cut $\mathrm{Cut}(\phi)$. It follows that every K-small homomorphism agrees with exactly one function $\phi \in \Phi^*$ and every K-large homomorphism agrees with exactly one function $\phi \in \Phi \setminus \Phi^*$. For $\phi \in \Phi$ let Z_ϕ be the number of homomorphisms from $(J, \mathbf{S})$ to H that agree with ϕ. Then

$$Z^* = \sum_{\phi \in \Phi^*} Z_\phi \quad \text{and} \quad Z_1 = \sum_{\phi \in \Phi \setminus \Phi^*} Z_\phi. \tag{4.25}$$

Let $\phi \in \Phi$. We are interested in the number Z_ϕ and investigate which homomorphisms h agree with ϕ. For each $v \in V(G)$ the image of J_v under h is fixed by the lists in $\mathbf{S}$ and the fact that $h(v) = \phi(v)$. Therefore, we only need to consider possible images of the graphs J_e. We make a case distinction depending on e.

- Let $e = \{u, v\} \in \mathrm{Cut}(\phi)$. (This means that e is dichromatic under h.) By the definition of J_e it follows that the image $h(I_e)$ is a subset of $\Gamma_H(h(u)) \cap \Gamma_H(h(v))$. The vertices $h(u)$ and $h(v)$ are distinct and are from $\{w_1, w_2, w_3\}$. As H is square-free it follows that $\Gamma_H(h(u)) \cap \Gamma_H(h(v)) = \{w_1, w_2, w_3\}$. In addition, each vertex of I_e is adjacent to one of the terminals. Since the images of these vertices are also in $\{w_1, w_2, w_3\}$ this does not bring any additional constraints. Summarising, since I_e has size t the edge e contributes a factor of 3^t to Z_ϕ.

- Let $e = \{u, v\} \in \mathrm{Mon}_i(\phi)$ for some $i \in [3]$. (This means that e is a monochromatic edge under h with $h(u) = h(v) = w_i$.) Then, for $j \in [3]$ with $j \neq i$, by the same arguments as before we have $h(I_e^j) \subseteq \{w_1, w_2, w_3\}$. However the vertices in I_e^i can be mapped to any neighbour of w_i. Therefore, each edge in $\mathrm{Mon}_i(\phi)$ contributes a factor of $|\Gamma_H(w_i)|^{t_i} \cdot 3^{t-t_i}$ to Z_ϕ.

Using this knowledge, for $\phi \in \Phi$, we have

$$Z_\phi = 3^{t|\mathrm{Cut}(\phi)|} \cdot \prod_{i \in [3]} \left(|\Gamma_H(w_i)|^{t_i} \cdot 3^{t-t_i}\right)^{|\mathrm{Mon}_i(\phi)|}.$$

Since $m = |\mathrm{Cut}(\phi)| + \sum_{i \in [3]} |\mathrm{Mon}_i(\phi)|$ we can simplify this expression to

$$Z_\phi = 3^{tm} \prod_{i \in [3]} (|\Gamma_H(w_i)|/3)^{t_i |\mathrm{Mon}_i(\phi)|}.$$

For $i \in [3]$, recall the fact that $|rs_i - t_i| \leq \delta'/m$, where $s_i = s \cdot \log_{|\Gamma_H(w_i)|/3} 2$. For an upper bound on Z_ϕ we use the fact that $t_i \leq rs_i + \delta'/m$. Also note that $|\Gamma_H(w_i)|$ is bounded above by $|V(H)|$ (which we use in the second inequality of the following expression). Furthermore, $\sum_{i\in[3]}|\mathrm{Mon}_i(\phi)| \leq m$ and $\delta' = \log_{|V(H)|} e^\delta$ (which we use in the third inequality of the following expression). Then

$$\begin{aligned} Z_\phi &\leq 3^{tm} \prod_{i\in[3]} (|\Gamma_H(w_i)|/3)^{(rs_i+\delta'/m)\cdot|\mathrm{Mon}_i(\phi)|} \\ &\leq 3^{tm} \cdot 2^{(\sum_{i\in[3]}|\mathrm{Mon}_i(\phi)|)\cdot rs} \cdot |V(H)|^{(\sum_{i\in[3]}|\mathrm{Mon}_i(\phi)|)\cdot \delta'/m} \\ &\leq 3^{tm} \cdot 2^{(\sum_{i\in[3]}|\mathrm{Mon}_i(\phi)|)\cdot rs} \cdot e^\delta. \end{aligned}$$

Analogously, for a lower bound on Z_ϕ we use the fact that $t_i \geq rs_i - \delta'/m$. We obtain $3^{tm} \cdot 2^{(\sum_{i\in[3]}|\mathrm{Mon}_i(\phi)|)\cdot rs} \cdot e^{-\delta} \leq Z_\phi$. Summarising, we have

$$3^{tm} \cdot 2^{(\sum_{i\in[3]}|\mathrm{Mon}_i(\phi)|)\cdot rs} \cdot e^{-\delta} \leq Z_\phi \leq 3^{tm} \cdot 2^{(\sum_{i\in[3]}|\mathrm{Mon}_i(\phi)|)\cdot rs} \cdot e^{\delta}. \tag{4.26}$$

Putting these bounds on Z_ϕ into the expression for Z^* in (4.25) gives

$$\sum_{\phi\in\Phi^*} 3^{tm} \cdot 2^{(\sum_{i\in[3]}|\mathrm{Mon}_i(\phi)|)\cdot rs} \cdot e^{-\delta} \leq Z^* \leq \sum_{\phi\in\Phi^*} 3^{tm} \cdot 2^{(\sum_{i\in[3]}|\mathrm{Mon}_i(\phi)|)\cdot rs} \cdot e^{\delta}.$$

Since $\sum_{i\in[3]}|\mathrm{Mon}_i(\phi)| = m - K$ for each $\phi \in \Phi^*$ and $|\Phi^*| = T$ we obtain

$$T \cdot 3^{tm} \cdot 2^{(m-K)\cdot rs} \cdot e^{-\delta} \leq Z^* \leq T \cdot 3^{tm} \cdot 2^{(m-K)\cdot rs} \cdot e^{\delta}.$$

We set $L = 3^{tm} 2^{(m-K)rs}$ to obtain

$$T \cdot e^{-\delta} \leq Z^*/L \leq T \cdot e^\delta. \tag{4.27}$$

Assume for now that $N\big((J, \mathbf{S}) \to H\big)/L \in [Z^*/L, Z^*/L + 1/4]$. Then the algorithm that makes a #Ret(H) oracle call with input $((J, \mathbf{S}), \varepsilon/42)$ returns a solution Q such that $Z^*/L \cdot e^{-\varepsilon/2} \leq \lfloor Q/L \rfloor \leq Z^*/L \cdot e^{\varepsilon/2}$ as was shown in [37, Proof of Theorem 3]. Using (4.27) and our choice of $\delta = \varepsilon/2$ this gives

$$T \cdot e^{-\varepsilon} \leq \lfloor Q/L \rfloor \leq T \cdot e^\varepsilon.$$

Therefore the output $\lfloor Q/L \rfloor$ approximates T with the desired precision. It remains to show the following claim.

Claim: $N\big((J, \mathrm{S}) \to H\big)/L \in [Z^*/L, Z^*/L + 1/4]$.

Proof of the claim: Recall that $N\big((J, \mathbf{S}) \to H\big) = Z^* + Z_1$. It is immediate that $N\big((J, \mathbf{S}) \to H\big)/L \geq Z^*/L$. It remains to show that $Z_1/L \leq 1/4$.

To obtain the following expression we first use (4.25) and (4.26). The third inequality then uses the fact that, for every $\phi \in \Phi \setminus \Phi^*$, we have $|\mathrm{Cut}(\phi)| \geq K + 1$

and hence $\sum_{i\in[3]}|\mathrm{Mon}_i(\phi)| \le m-(K+1)$. Finally, in the fourth inequality we use the fact that $|\Phi\setminus\Phi^*| \le |V(H)|^n$

$$\begin{aligned}
Z_1 &= \sum_{\phi\in\Phi\setminus\Phi^*} Z_\phi \\
&\le \sum_{\phi\in\Phi\setminus\Phi^*} 3^{tm}\cdot 2^{(\sum_{i\in[3]}|\mathrm{Mon}_i(\phi)|)\cdot rs}\cdot e^{\delta} \\
&\le \sum_{\phi\in\Phi\setminus\Phi^*} 3^{tm}\cdot 2^{(m-K-1)\cdot rs}\cdot e^{\delta} \\
&\le |V(H)|^n\cdot 3^{tm}\cdot 2^{(m-K-1)\cdot rs}\cdot e^{\delta}
\end{aligned}$$

Recall the definition $L=3^{tm}\cdot 2^{(m-K)\cdot rs}$. It follows that

$$Z_1/L \le \frac{|V(H)|^n\cdot e^{\delta}}{2^{rs}} \le 1/4,$$

where the last inequality holds for sufficiently large n by our choice of $s=n^2$ and the fact that $r\ge 1$. This proves the claim and completes the proof. *(End of the proof of the claim.)* □

4.3.6 Square-Free Graphs with an Induced Reflexive Cycle of Length at least 5

Let H be a connected square-free graph with an induced reflexive cycle of length at least 5. If all cycles in H have length at least 5, then H has girth at least 5 and the complexity of #RET(H) is classified by Theorem 1.11. In the special case where H is reflexive this classification is straightforward to see: Either there is just a single cycle in H, then H is a pseudotree and #SAT-hardness follows from NP-hardness for the decision problem [52, Corollary 4.2, Theorem 5.1] together with [37, Theorem 1] — or there are multiple cycles (all of which have length at least 5), then there exists an induced WR_3 (as H is connected) and hardness follows from Lemma 4.38.

Thus, it remains to show hardness if H contains both a cycle of length at least 5 as well as a cycle of length at most 5, i.e. (since H is square-free) it contains a triangle. The hardness proof we give in this section will handle the case were H includes triangles but will not rely on this fact (i.e. it will also cover the before-mentioned case where all cycles have length at least 5 without relying on hardness results for the decision problem).

As mentioned before, it is known that approximately counting list homomorphisms to reflexive graphs with an induced cycle of length at least 4 is #SAT-hard [63, Lemma 3.4]. That proof makes use of a certain set of two-vertex lists. In the proof of Lemma 4.46 we will use single-vertex lists to simulate these two-vertex lists.

As we have already shown #SAT-hardness results for square-free graphs with an induced WR_3 or an induced net in Sections 4.3.4 and 4.3.5 we now focus on graphs that do not contain such subgraphs. When considering reflexive graphs it turns out that this leaves a class of graphs that we call reflexive triangle-extended cycles. We will also make use of these reflexive triangle-extended cycles when considering square-free graphs H that are not necessarily reflexive. When we do this we will restrict to the (reflexive) subgraph induced by the looped vertices of H.

Definition 4.41. A *reflexive triangle-extended cycle* of length q consists of a reflexive cycle $c_0, \ldots, c_{q-1}$ together with a set $\mathcal{I} \subseteq \{0, \ldots, q-1\}$, and a reflexive triangle $d_i, c_i, c_{i+1 \bmod q}$ for each $i \in \mathcal{I}$. An example of a reflexive triangle-extended cycle is depicted in Figure 4.17.

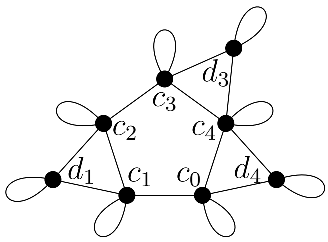


Figure 4.17: Reflexive triangle-extended cycle with $q = 5$ and $\mathcal{I} = \{1, 3, 4\}$.

Definition 4.42. Analogously to Definition 4.41, a *reflexive triangle-extended path* is a reflexive path $c_0, \ldots, c_{q-1}$ together with a set $\mathcal{I} \subseteq \{0, \ldots, q-2\}$, and a reflexive triangle d_i, c_i, c_{i+1} for each $i \in \mathcal{I}$.

Lemma 4.43. *Let H be a connected reflexive square-free graph that does not contain an induced* WR_3 *and also does not contain an induced net. If H contains an induced cycle of length at least* 5 *then H is a reflexive triangle-extended cycle of length at least* 5*. Otherwise it is a reflexive triangle-extended path.*

Proof. **Case 1: H contains an induced cycle $C = c_0, \ldots, c_{q-1}$ with $q \geq 5$.** If H is just the cycle C, then the statement of the lemma is true ($\mathcal{I} = \emptyset$). Otherwise, consider any $d \in V(H) \setminus V(C)$ with a neighbour $c \in V(C)$ (has to exist since H is connected).

Since H does not contain an induced WR_3 the vertex d is adjacent to a neighbour $c' \in V(C)$ of c. Let c^0 and c'' be the other neighbours of c and c' in C, respectively, i.e. $\Gamma_C(c) = \{c^0, c'\}$ and $\Gamma_C(c') = \{c, c''\}$. The vertices $\{c^0, c, c', c''\}$ are all distinct as C has length at least 5. As H is square-free we observe

$$\{d, c^0\} \notin E(H) \text{ and } \{d, c''\} \notin E(H). \tag{4.28}$$

The proof of the following claim directly proves that H is a reflexive triangle-extended cycle.

Claim: $\Gamma_H(d) = \{c, c'\}$.

Proof of the claim: Assume there exists a neighbour $d' \notin \{c, c'\}$ of d in H. By (4.28) we have $d' \notin \{c^0, c''\}$. Furthermore, since H is square-free, we obtain the following.

$$\text{There is no } u \neq d \text{ with } u \in \Gamma_H(c) \cap \Gamma_H(d') \text{ or } u \in \Gamma_H(c') \cap \Gamma_H(d'). \tag{4.29}$$

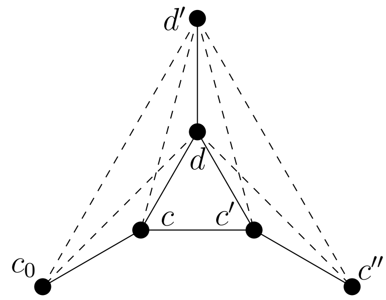


Figure 4.18: The graph H' induced by $\{c^0, c, c', c'', d, d'\}$. Loops are omitted. Dashed lines show edges that cannot exist by the fact that H is square-free.

Let H' be the subgraph of H induced by the vertices $\{c^0, c, c', c'', d, d'\}$, see Figure 4.18. Then H' is a net (cf. Figure 4.15 where $c = w_1$, $d = w_2$ and $c' = w_3$).

Because of (4.29) we have $\Gamma_{H'}(d') = \{d\}$. (The dashed edges incident to d' in Figure 4.18 cannot exist.) Because of (4.28) we have $\Gamma_{H'}(d) = \{d', c, c'\}$. (The dashed edges incident to d in Figure 4.18 cannot exist.) Finally, since C is an induced cycle, there are no edges between the vertices $\{c^0, c, c', c''\}$ outside of C. Therefore, H' is an *induced* net in H, which gives a contradiction. This proves the claim in Case 1. **(End of Case 1)**

Case 2: All induced cycles in H are triangles. This case is handled very similarly to the previous one: Let $P = c_0, \ldots, c_{q-1}$ be a maximal induced path in H. If H is just the path P, then the statement of the lemma is true ($\mathcal{I} = \emptyset$). Otherwise, let $d \in V(H) \setminus V(P)$ be a neighbour of $c \in V(P)$. We show that d is adjacent to a neighbour $c' \in V(P)$ of c:

- If c is an inner vertex of P then, since H does not contain an induced WR_3, d is also adjacent to a neighbour of c in P.
- If c is an endpoint of P, then d has to be adjacent to a vertex $c' \in V(P)$ with $c' \neq c$ as P is maximal induced. Without loss of generality assume that c' is the neighbour of d which is closest to c in P. Then c', c, d has to be a triangle (c' has to be a neighbour of c) as P is induced and all induced cycles in H are triangles.

Then the proof of the following claim shows that H is a reflexive triangle-extended path.

Claim: $\Gamma_H(d) = \{c, c'\}$.

Proof of the claim: Assume there exists a neighbour $d' \notin \{c, c'\}$ of d in H. We observe the following properties:

- The vertex d' does not have a neighbour in P: Assume the opposite and let $u \in P$ be a neighbour of d'. Without loss of generality let u be closer to c' than c in P. Furthermore, let u be the neighbour of d' in P which is closest to c'. Then the edge $\{d', u\}$ and the path d', d, c' close an induced cycle with P. If $u \neq c'$ this cycle has length greater than 3, a contradiction. If $u = c'$ we obtain a contradiction to the fact that H is square-free (see Figure 4.18).

- Both c and c' are inner points of P: Suppose, for contradiction, that c' is an end point of P. Since d' does not have a neighbour in P, replacing c' by d, d' in P gives an induced path P'. Moreover, P' is longer than P which is a contradiction to the maximality of P. This shows that c' cannot be an end point of P. Analogously c cannot be an end point of P.

Then let c^0 and c'' be the other neighbours of c and c' in P (they have to exist since c and c' are inner points of P). The remainder of the argument is analogous to the proof of the claim in Case 1. **(End of Case 2)** $\square$

The goal of the remainder of this section is to prove Lemma 4.46. In order to prove Lemma 4.46 we work with the following parameterised version of the list homomorphism counting problem. Let H be a graph and $\mathcal{L}$ be a set of subsets of $V(H)$.

Name: $\#\text{HOM}(H, \mathcal{L})$.
Input: An irreflexive graph G and a collection of lists $\mathbf{S} = \{S_v \in \mathcal{L} \mid v \in V(G)\}$.
Output: $N\big((G, \mathbf{S}) \to H\big)$.

We also use the following lemma.

Lemma 4.44 ([63, Proof of Lemma 3.4])**.** *Let H be a graph that contains an induced reflexive cycle $C = c_0, \dots, c_{q-1}$ on $q \geq 4$ vertices. Further, let $\mathcal{L} = \{\{c_0, c_1\}, \{c_0, c_2\}, \dots, \{c_0, c_{q-1}\}\}$. Then $\#\text{HOM}(H, \mathcal{L}) \equiv_{\text{AP}} \#\text{SAT}$.*

The key to proving Lemma 4.46 is the following result. It states that for certain graphs that contain a reflexive triangle-extended cycle we can simulate each size-2 list of vertices in the corresponding cycle C by gadgets using only single-vertex lists.

Lemma 4.45. *Let H be a square-free graph that does not contain any mixed triangle as an induced subgraph and let H^* be the graph induced by the looped vertices of H. Suppose that H^* contains a connected component H^{**} that is a reflexive triangle-extended cycle, where $C = c_0, \dots, c_{q-1}$ is the corresponding reflexive cycle as given by Definition 4.41 and the length of C is $q \geq 5$. Let $\mathcal{L}$ and $\mathcal{L}'$ be sets with*

$$\mathcal{L}' \subsetneq \mathcal{L} \subseteq \{\{c_0, c_1\}, \{c_0, c_2\}, \dots, \{c_0, c_{q-1}\}\} \quad \text{such that} \quad |\mathcal{L}'| = |\mathcal{L}| - 1.$$

Let $\mathcal{L}'' = \mathcal{L}' \cup \big\{S \subseteq V(H) \mid |S| \in \{1, |V(H)|\}\big\}$. Then

$$\#\text{HOM}(H, \mathcal{L}) \leq_{\text{AP}} \#\text{HOM}(H, \mathcal{L}'').$$

Proof. For the reflexive triangle-extended cycle H^{**} we use the notation (C, $\mathcal{I}$ and d_i for $i \in \mathcal{I}$) as given by Definition 4.41. Let $\mathcal{L}$, $\mathcal{L}'$ and $\mathcal{L}''$ be as given in the statement of the lemma. We have $\mathcal{L}' = \mathcal{L} \setminus \{\{c_0, c_\ell\}\}$ for some $\ell \in [q-1]$. Let $(G, \mathbf{S}^G)$ be an input to $\#\text{HOM}(H, \mathcal{L})$. Let $U = \{u \in V(G) \mid S^G_u = \{c_0, c_\ell\}\}$. Since $\{c_0, c_\ell\}$ is not part of $\mathcal{L}''$ the goal is to simulate $\{c_0, c_\ell\}$ using gadgetry and lists from $\mathcal{L}''$.

From $(G, \mathbf{S}^G)$ we define an instance $(J, \mathbf{S}^J)$ of $\#\text{HOM}(H, \mathcal{L}'')$. To this end we will define, for each $u \in U$, a vertex gadget J_u and a corresponding set of lists

$\mathbf{S}^u = \{S_v^u \in \mathcal{L}'' \mid v \in V(J_u)\}$. There are two distinct paths in C that connect c_0 and c_ℓ: $P_1 = c_0, c_1, \ldots, c_\ell$ and $P_2 = c_\ell, \ldots, c_{q-1}, c_0$. The graph J_u has two parts: a graph J_{P_1} and a graph J_{P_2}, which depend on the paths P_1 and P_2. We first define J_{P_1} and J_{P_2} (and the corresponding sets of lists $\mathbf{S}^1 = \{S_v^1 \in \mathcal{L}'' \mid v \in V(J_{P_1})\}$ and $\mathbf{S}^2 = \{S_v^2 \in \mathcal{L}'' \mid v \in V(J_{P_2})\}$) and then we describe the way in which they are connected to form J_u. The definition of J_{P_1} depends on ℓ, the number of edges of P_1:

- If ℓ is even, think of a path on $\ell/2$ edges. Let v^* be one of the end points of this path. We pin v^* to $c_{\ell/2}$ (the vertex in the "middle" of P_1). This graph is depicted in Figure 4.19 on the left. The graph J_{P_1} is then a modification of this graph where each vertex of the path is replaced by a clique of size 2 (apart from v^* which will be pinned to $c_{\ell/2}$ anyway). This modification will ensure that only looped vertices can be in the image of J_{P_1}. The graph J_{P_1} is depicted in Figure 4.19 (on the right) and is formally defined as follows: $V(J_{P_1}) = \{v_i, v_i' \mid i \in \{0, \ldots, \ell/2 - 1\}\} \cup \{v^*\}$, where all these vertices are distinct from the vertices of G (and distinct from the vertices used in other gadgets). $E(J_{P_1}) = \big\{\{v_i, v_i'\} \mid i \in \{0, \ldots, \ell/2 - 1\}\big\} \cup \big\{\{v_{i-1}, v_{i-1}'\} \times \{v_i, v_i'\} \mid i \in [\ell/2 - 1]\big\} \cup \big\{\{v_{\ell/2-1}, v^*\}, \{v_{\ell/2-1}', v^*\}\big\}$. We set $S_{v^*}^1 = \{c_{\ell/2}\}$ and $S_v^1 = V(H)$ for all $v \in V(J_{P_1}) \setminus \{v^*\}$
- If ℓ is odd, then J_{P_1} is defined very similarly to the previous case, see Figure 4.20. Formally, $V(J_{P_1}) = \{v_i, v_i' \mid i \in \{0, \ldots, \lfloor \ell/2 \rfloor\}\} \cup \{v_1^*, v_2^*\}$, where all these vertices are distinct from the vertices of G (and distinct from the vertices used in other gadgets). $E(J_{P_1}) = \big\{\{v_i, v_i'\} \mid i \in \{0, \ldots, \lfloor \ell/2 \rfloor\}\big\} \cup \big\{\{v_{i-1}, v_{i-1}'\} \times \{v_i, v_i'\} \mid i \in [\lfloor \ell/2 \rfloor]\big\} \cup \big\{\{v_{\lfloor \ell/2 \rfloor}, v_{\lfloor \ell/2 \rfloor}'\} \times \{v_1^*, v_2^*\}\big\}$. We set $S_{v_1^*}^1 = \{c_{\lceil \ell/2 \rceil}\}$, $S_{v_2^*}^1 = \{c_{\lfloor \ell/2 \rfloor}\}$ and $S_v^1 = V(H)$ for all $v \in V(J_{P_1}) \setminus \{v_1^*, v_2^*\}$.

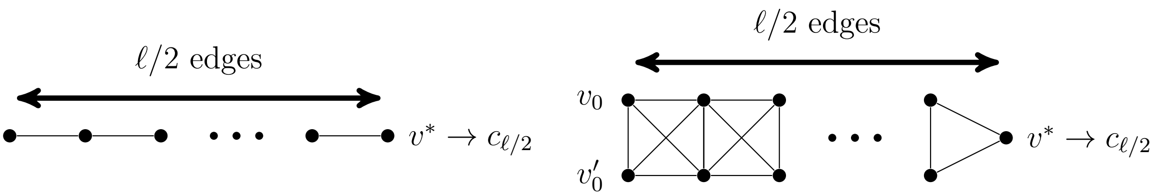


Figure 4.19: Construction of the graph J_{P_1} for even ℓ. The label of the form $v^* \to c_{\ell/2}$ means that the vertex $v^* \in V(J_{P_1})$ is "pinned" to $c_{\ell/2} \in V(H)$ since $S_{v^*}^1 = \{c_{\ell/2}\}$.

This completes the definition of J_{P_1}. J_{P_2} is defined analogously. However, the length of P_2 is $q - \ell$ instead of ℓ. Furthermore, if $q - \ell$ is even, note that the vertex in the "middle" of $P_2 = c_\ell, \ldots, c_{q-1}, c_0$ is $c_{(q+\ell)/2}$ rather than $c_{\ell/2}$. (Accordingly, if $q - \ell$ is odd, we use $c_{\lceil (q+\ell)/2 \rceil}$ and $c_{\lfloor (q+\ell)/2 \rfloor}$ to "pin" to.) Formally, J_{P_2} is defined as follows:

- If $q - \ell$ is even, we have $V(J_{P_2}) = \{w_i, w_i' \mid i \in \{0, \ldots, (q-\ell)/2 - 1\}\} \cup \{w^*\}$ and $E(J_{P_2}) = \big\{\{w_i, w_i'\} \mid i \in \{0, \ldots, (q-\ell)/2 - 1\}\big\} \cup \big\{\{w_{i-1}, w_{i-1}'\} \times \{w_i, w_i'\} \mid i \in [(q-\ell)/2 - 1]\big\} \cup \big\{\{w_{(q-\ell)/2-1}, w^*\}, \{w_{(q-\ell)/2-1}', w^*\}\big\}$. We set $S_{w^*}^2 = \{c_{(q+\ell)/2}\}$ and $S_w^2 = V(H)$ for all $w \in V(J_{P_2}) \setminus \{w^*\}$

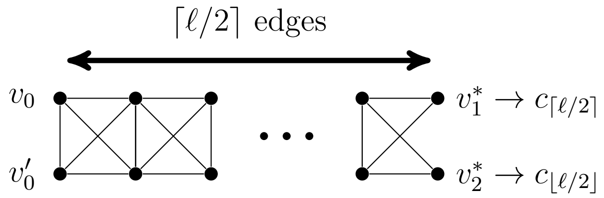


Figure 4.20: The graph J_{P_1} for odd ℓ. A label of the form $a \to b$ means that the vertex $a \in V(J_{P_1})$ is "pinned" to $b \in V(H)$ since $S^1_a = \{b\}$.

- If $q-\ell$ is odd, we have $V(J_{P_2}) = \{w_i, w'_i \mid i \in \{0, \dots, \lfloor (q-\ell)/2 \rfloor\}\} \cup \{w^*_1, w^*_2\}$ and $E(J_{P_2}) = \big\{\{w_i, w'_i\} \mid i \in \{0, \dots, \lfloor (q-\ell)/2 \rfloor\}\big\} \cup \big\{\{w_{i-1}, w'_{i-1}\} \times \{w_i, w'_i\} \mid i \in [\lfloor (q-\ell)/2 \rfloor]\big\} \cup \big\{\{w_{\lfloor (q-\ell)/2 \rfloor}, w'_{\lfloor (q-\ell)/2 \rfloor}\} \times \{w^*_1, w^*_2\}\big\}$. We set $S^2_{w^*_1} = \{c_{\lceil (q+\ell)/2 \rceil}\}$, $S^2_{w^*_2} = \{c_{\lfloor (q+\ell)/2 \rfloor}\}$ and $S^2_w = V(H)$ for all $w \in V(J_{P_2}) \setminus \{w^*_1, w^*_2\}$.

We can now define the graph J_u (for $u \in U$): J_u is the graph obtained from J_{P_1} and J_{P_2} by identifying v_0 with w_0 and v'_0 with w'_0. As an example, if P_1 has even length and P_2 has odd length, the graph J_u is depicted in Figure 4.21. Let $\mathcal{A}(u)$ denote the set that contains the two vertices $v_0 = w_0$ and $v'_0 = w'_0$. The lists $\mathbf{S}^u$ of the vertices in J_u are the union of $\mathbf{S}^1$ and $\mathbf{S}^2$. (Note that this is well-defined as $S^1_{v_0} = S^1_{v'_0} = S^2_{w_0} = S^2_{w'_0} = V(H)$.) This completes the definition of J_u and $\mathbf{S}^u$.

We can finally define the instance $(J, \mathbf{S}^J)$: J is the graph with vertices $V(J) = V(G) \setminus U \cup \bigcup_{u \in U} V(J_u)$ and edges

$$\begin{aligned} E(J) = &\big\{\{v, v'\} \mid \{v, v'\} \in E(G) \text{ and } v, v' \in V(G) \setminus U\big\} \\ &\cup \big\{\{v\} \times \mathcal{A}(v') \mid \{v, v'\} \in E(G) \text{ and } v \in V(G) \setminus U, v' \in U\big\} \\ &\cup \big\{\mathcal{A}(v) \times \mathcal{A}(v') \mid \{v, v'\} \in E(G) \text{ and } v, v' \in U\big\} \\ &\cup \bigcup_{u \in U} E(J_u). \end{aligned}$$

Finally, $\mathbf{S}^J = \{S^J_v \subseteq V(H) \mid v \in V(J)\}$ with

$$S^J_v = \begin{cases} S^G_v, & \text{if } v \in V(G) \setminus U \\ S^u_v & \text{if otherwise } v \in V(J_u) \text{ for some } u \in U. \end{cases}$$

Note that for each $v \in V(J)$ we have $S^J_v \in \mathcal{L}''$ and therefore $(J, \mathbf{S}^J)$ is a valid input to $\#\textsc{Hom}(H, \mathcal{L}'')$.

To show how homomorphisms from $(J, \mathbf{S}^J)$ to H relate to homomorphisms from $(G, \mathbf{S}^G)$ to H we determine some properties of the gadget J_u. Consider the case where ℓ, the number of edges of P_1, is even, and $q - \ell$, the number of edges of P_2, is odd. (The other cases of ℓ and $q - \ell$ even or odd will be analogous.) Then J_u is the gadget depicted in Figure 4.21. Let the vertices of J_u be labelled accordingly. Now let h be a homomorphism from $(J, \mathbf{S}^J)$ to H.

Claim 1: For all $i \in \{0, \dots, \ell/2-1\}$, the image $h(\{v_i, v'_i\})$ contains at least one looped vertex. For all $j \in \{0, \dots, \lfloor (q-\ell)/2 \rfloor\}$, the image $h(\{w_j, w'_j\})$ contains at least one looped vertex.

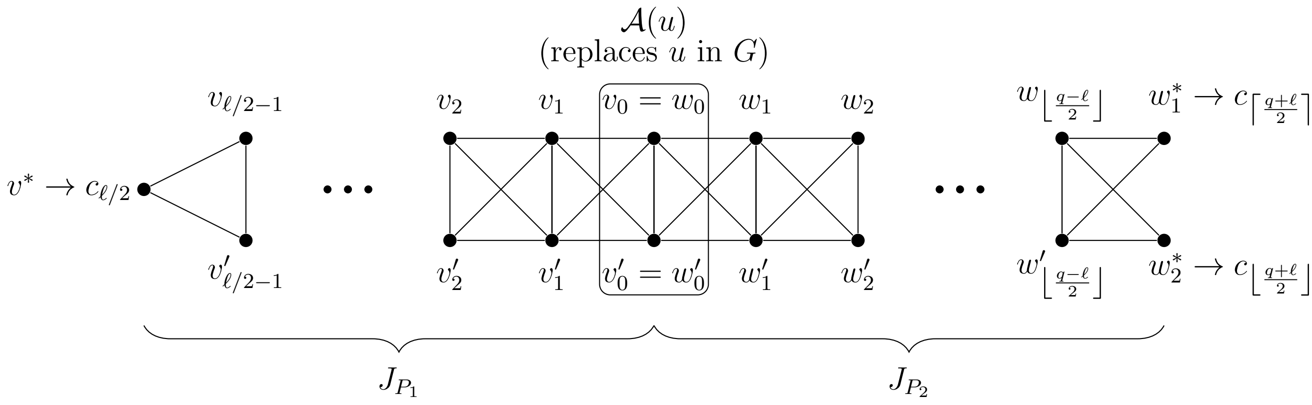


Figure 4.21: The graph J_u if ℓ (the number of edges of P_1) is even and $q-\ell$ (the number of edges of P_2) is odd.

Proof of Claim 1: First consider $h(\{v_{\ell/2-1}, v'_{\ell/2-1}\})$. Assume that both $h(v_{\ell/2-1})$ and $h(v'_{\ell/2-1})$ are unlooped vertices of H. Since $\{v_{\ell/2-1}, v'_{\ell/2-1}\}$ is an edge in J, $h(v_{\ell/2-1})$ and $h(v'_{\ell/2-1})$ have to be connected by an edge in H and therefore have to be different unlooped vertices. However, the vertices $v_{\ell/2-1}$ and $v'_{\ell/2-1}$ are also neighbours of v^* which is pinned to the looped vertex $c_{\ell/2}$. It follows that $h(v_{\ell/2-1}), h(v'_{\ell/2-1}), c_{\ell/2}$ form a mixed triangle in H, a contradiction. This argument can be repeated iteratively for each $i = \ell/2-2, \dots, 0$ (using the fact that $h(\{v_{i+1}, v'_{i+1}\})$ contains a looped vertex). The argument for $h(\{w_j, w'_j\})$ ($j \in \{0, \dots, \lfloor (q-\ell)/2 \rfloor\}$) is analogous.

Claim 2: There exists a vertex $v \in \mathcal{A}(u)$ with $h(v) \in \{c_0, c_\ell\}$.

Proof of Claim 2: Let $\mathcal{A}(u) = \{v, v'\}$ where $h(v)$ is looped (this can be assumed without loss of generality by Claim 1). We will show that $h(v) \in \{c_0, c_\ell\}$: By Claim 1 and the construction of J_{P_1} there exists a walk on $\ell/2$ edges in H which uses looped vertices only and goes from $c_{\ell/2}$ to $h(v)$, which by assumption is looped itself. As this walk is looped it is in H^*, and as it contains $c_{\ell/2}$ it is in H^{**}. Hence,

$$h(v) \in \Gamma_{H^{**}}^{\ell/2}(c_{\ell/2}). \tag{4.30}$$

Similarly, by the construction of J_{P_2} we obtain

$$h(v) \in \Gamma_{H^{**}}^{\lceil (q-\ell)/2 \rceil}(c_{\lfloor (q+\ell)/2 \rfloor}) \cap \Gamma_{H^{**}}^{\lceil (q-\ell)/2 \rceil}(c_{\lceil (q+\ell)/2 \rceil}). \tag{4.31}$$

Since H^{**} is a reflexive triangle-extended cycle we have

$$\Gamma_{H^{**}}^{\ell/2}(c_{\ell/2}) = \{c_0, \dots, c_\ell\} \cup \{d_i \mid i \in \mathcal{I} \cap \{0, \dots, \ell-1\}\}$$

and

$$\Gamma_{H^{**}}^{\lceil (q-\ell)/2 \rceil}(c_{\lfloor (q+\ell)/2 \rfloor}) \cap \Gamma_{H^{**}}^{\lceil (q-\ell)/2 \rceil}(c_{\lceil (q+\ell)/2 \rceil}) = \{c_\ell, \dots, c_{q-1}, c_0\} \cup \{d_i \mid i \in \mathcal{I} \cap \{\ell, \dots, q-1\}\}.$$

Therefore, from (4.30) and (4.31) it follows that $h(v) \in \{c_0, c_\ell\}$.

Claim 3: Let $v \in \mathcal{A}(u)$ and $h(v) \in \{c_0, c_\ell\}$. Then the image of the remaining vertices of J_u under h is determined completely. In particular, $h(\mathcal{A}(u)) = \{h(v)\}$.

Proof of Claim 3: Consider the case where $h(v) = c_0$ (the case $h(v) = c_\ell$ can be treated analogously). Since H^{**} is a reflexive triangle-extended cycle and a connected component of H^*, the walk $c_{\ell/2}, c_{\ell/2-1}, \ldots, c_0$ is the only $\ell/2$-edge walk on looped vertices in H that leads from $c_{\ell/2}$ to c_0 (since there are no reflexive shortcuts in C). Thus, by Claim 1 and the construction of J_{P_1}, we have $\forall i \in \{0, \ldots, \ell/2-1\}$, $c_i \in h(\{v_i, v_i'\})$. We assume without loss of generality (by renaming) that for each $i \in \{0, \ldots, \ell/2-1\}$ we have $h(v_i) = c_i$.

Now consider the image $h(v_i')$ for some $i \in [\ell/2-1]$. The vertex v_i' is a neighbour of v_{i-1}, v_i and v_{i+1} (or alternatively v^* if $i = \ell/2-1$) in J. Therefore, by the fact that for each $i \in \{0, \ldots, \ell/2-1\}$ we have $h(v_i) = c_i$ and by the pinning that ensures $h(v^*) = c_{\ell/2}$, we know that $h(v_i')$ has to be a neighbour of c_{i-1}, c_i and c_{i+1} in H. Since there is no edge between c_{i-1} and c_{i+1} we have $h(v_i') \notin \{c_{i-1}, c_{i+1}\}$. Then $h(v_i') = c_i$ as otherwise $c_{i-1}, c_i, c_{i+1}, h(v_i')$ would form a square in H which is a contradiction to the fact that H is square-free. So we have established that for each $i \in [\ell/2-1]$ it holds that $h(v_i) = h(v_i') = c_i$.

Similarly one establishes that for each $i \in [\lfloor (q-\ell)/2 \rfloor]$ it holds that $h(\{w_i, w_i'\}) = c_{q-i}$: Note that by the construction of J_{P_2} the homomorphism h has to map $w_{\lfloor (q-\ell)/2 \rfloor}$ and $w'_{\lfloor (q-\ell)/2 \rfloor}$ to common neighbours of $c_{\lfloor (q+\ell)/2 \rfloor}$ and $c_{\lceil (q+\ell)/2 \rceil}$. Since H^{**} is a reflexive triangle-extended cycle and a connected component of H^*, the walk $c_{\lceil (q+\ell)/2 \rceil} \ldots, c_{q-1}, c_0$ is the only $\lfloor (q-\ell)/2 \rfloor$-edge walk on looped vertices in H that leads from a common neighbour of $c_{\lfloor (q+\ell)/2 \rfloor}$ and $c_{\lceil (q+\ell)/2 \rceil}$ to c_0. Therefore, we have $c_0 \in h(\{w_0, w_0'\})$ and $\forall i \in [\lfloor (q-\ell)/2 \rfloor]$, $c_{q-i} \in h(\{w_i, w_i'\})$. Then by the same arguments as before we establish that for each $i \in [\lfloor (q-\ell)/2 \rfloor]$ it holds that $h(w_i) = h(w_i') = c_{q-i}$.

Finally, v_0' is a neighbour of w_1, $v_0 (= w_0)$ and v_1. Hence, $h(v_0')$ is a neighbour of $h(w_1) = c_{q-1}$, $h(v_0) = c_0$ and $h(v_1) = c_1$. Since there is no edge between c_{q-1} and c_1 we have $h(v_i') \notin \{c_{q-1}, c_1\}$. Then $h(v_0') = c_0$ as otherwise $c_{q-1}, c_0, c_1, h(v_0')$ would form a square in H which is a contradiction to the fact that H is square-free. We obtain $h(v_0') = h(v_0) = c_0$. This proves Claim 3.

From the construction of J together with Claim 2 and Claim 3 we directly obtain that $N\big((G, \mathbf{S}^G) \to H\big) = N\big((J, \mathbf{S}^J) \to H\big)$, which gives the sought-for reduction in the case where ℓ is even and $q-\ell$ is odd. All other cases of ℓ even or odd and $q-\ell$ even or odd can be treated analogously. □

Lemma 4.46. *Let H be a square-free graph. If H contains a reflexive cycle of length at least* 5 *as an induced subgraph then* $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\mathrm{RET}(H)$.

Proof. Suppose that H contains a mixed triangle as an induced subgraph, then the statement of this lemma follows from Lemmas 4.16 and 4.17. We can now assume that H does not contain any mixed triangle as an induced subgraph.

Let $C = c_0, \ldots, c_{q-1}$ be the reflexive cycle of length $q \geq 5$ in H. Let H^* be the graph induced by the looped vertices in H. If H (and hence H^*) contains an induced WR_3 or an induced net then $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\mathrm{RET}(H)$ by Lemmas 4.38 and 4.40, respectively. Otherwise, the connected component of H^* that contains the cycle C has to be a reflexive triangle-extended cycle by Lemma 4.43 and therefore H fulfills the requirements of Lemma 4.45.

Let $\mathcal{L} = \{\{c_0, c_1\}, \{c_0, c_2\}, \ldots, \{c_0, c_{q-1}\}\}$. Then $\#\mathrm{HOM}(H, \mathcal{L}) \equiv_{\mathrm{AP}} \#\mathrm{SAT}$ by Lemma 4.44. We can use Lemma 4.45 iteratively to obtain $\#\mathrm{HOM}(H, \mathcal{L}) \leq_{\mathrm{AP}} \#\mathrm{HOM}(H, \{S \subseteq V(H) \mid |S| \in \{1, |V(H)|\}\})$. Note that by the problem definitions we have $\#\mathrm{HOM}(H, \{S \subseteq V(H) \mid |S| \in \{1, |V(H)|\}\}) = \#\mathrm{RET}(H)$. Summarising,

$$\begin{aligned}\#\mathrm{SAT} &\equiv_{\mathrm{AP}} \#\mathrm{HOM}(H, \mathcal{L}) \\ &\leq_{\mathrm{AP}} \#\mathrm{HOM}(H, \{S \subseteq V(H) \mid |S| \in \{1, |V(H)|\}\}) = \#\mathrm{RET}(H).\end{aligned}$$

□

4.4 Putting the Pieces together

This section contains the proof of Theorem 1.13 (we restate it here for convenience). We will use the following theorem, which we re-state from Chapter 3.

Theorem 3.7. *Suppose that H is an irreflexive square-free graph.*

(i) If every connected component of H is a star, then $\#\mathrm{RET}(H)$ is in FP*.*

(ii) Otherwise, if every connected component of H is a caterpillar, then $\#\mathrm{RET}(H)$ approximation-equivalent to #BIS*.*

(iii) Otherwise, $\#\mathrm{RET}(H)$ approximation-equivalent to #SAT*.*

In the following lemma we collect the #SAT-hardness results which we use to prove Theorem 1.13.

Lemma 4.47. *Let H be a connected square-free graph other than a reflexive clique, a member of $\mathcal{H}_{\mathrm{BIS}}$, or an irreflexive caterpillar. Then $\#\mathrm{RET}(H)$ is approximation-equivalent to* #SAT*.*

Proof. If H is irreflexive then by assumption it is not a caterpillar. Thus $\#\mathrm{RET}(H)$ is approximation-equivalent to #SAT by Theorem 3.7.

If H is not irreflexive, i.e. if H has at least one loop, then we collect different #SAT-hardness results proved throughout this work to show hardness. If H contains a mixed triangle as induced subgraph, then $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\mathrm{RET}(H)$ by Lemmas 4.16 and 4.17. If H does not contain a mixed triangle as an induced subgraph but contains a WR_3, a net or a reflexive cycle of length at least 5 as an induced subgraph, then $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\mathrm{RET}(H)$ by Lemmas 4.38, 4.40 and 4.46, respectively. It remains to show $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\mathrm{RET}(H)$ if H is a graph with the following properties:

- H is connected and square-free.
- H has at least one looped vertex.

- H is not a reflexive clique.
- $H \notin \mathcal{H}_{\text{BIS}}$.
- H does not contain any of the following as an induced subgraph: a mixed triangle, a WR_3, a net, a reflexive cycle of length at least 5.

Let H^* be a connected component in the graph induced by the looped vertices in H. (It will turn out that H^* is actually the only connected component in this graph.) Then by the properties of H and Lemma 4.43 we know that H^* is a reflexive triangle-extended path. We recall the definition of a reflexive triangle-extended path from Definition 4.42: H^* is a reflexive path $c_0, \dots, c_{q-1}$ together with a set $\mathcal{I} \subseteq \{0, \dots, q-2\}$, and a reflexive triangle d_i, c_i, c_{i+1} for each $i \in \mathcal{I}$. Since H^* is not a reflexive clique it holds that $q - 1 \geq 2$.

Note that $H^* \in \mathcal{H}_{\text{BIS}}$ (where the set of bristles is empty and c_i corresponds to p_i). For all $i \in \mathcal{I}$ the clique K_i has size 3, for $i \notin \mathcal{I}$ it has size 2. Since $H \notin \mathcal{H}_{\text{BIS}}$ and H is connected, there exists a vertex u outside of H^* with a neighbour v in H^*. The vertex u has to be unlooped as otherwise it would be part of the reflexive connected component H^*. We consider four disjoint cases.

- If $\deg_H(u) \geq 2$, then consider two different cases:
  - If u is adjacent to a vertex $w \in \Gamma_H(v)$ with $w \neq v$, then $w \neq u$ since u is unlooped and u, v, w is a mixed triangle, a contradiction.
  - If v is the only neighbour of u in $\Gamma_H(v)$, then the requirements of Lemma 4.33 are met (with $b = v$ and $g = u$) and hence $\#\text{SAT} \leq_{\text{AP}} \#\text{RET}(H)$.
- If $\deg_H(u) = 1$ and $v \in \{d_i \mid i \in \mathcal{I}\}$, then $H[\Gamma_H(v)]$ is a graph of the form $X(k_1, 0, 1)$ where $k_1 \geq 1$ (cf. Figure 4.5) and therefore

  $$\begin{aligned}\#\text{SAT} &\leq_{\text{AP}} \#\text{HOM}(X(k_1,0,1)) \\ &\leq_{\text{AP}} \#\text{RET}(X(k_1,0,1)) = \#\text{RET}(H[\Gamma_H(v)]) \leq_{\text{AP}} \#\text{RET}(H),\end{aligned}$$

  by Lemma 4.30, Observation 1.18 and Observation 4.15 (in the order of the reductions used).
- If $\deg_H(u) = 1$ and $v \in \{c_0, c_{q-1}\}$. Without loss of generality (by renaming the vertices of H^*) let $v = c_0$. If $0 \in \mathcal{I}$ then c_0 is part of a reflexive triangle d_0, c_0, c_1 in H^* and $H[\Gamma_H(v)]$ is a graph of the form $X(k_1, 0, 1)$ where $k_1 \geq 1$. Then we have $\#\text{SAT} \leq_{\text{AP}} \#\text{RET}(H)$ by the same arguments used in the previous case. If otherwise $0 \notin \mathcal{I}$ then c_1 is the only neighbour of c_0 in H^* and $H[\Gamma_H(v)]$ is a graph of the form $X(k_1, 1, 0)$ where $k_1 \geq 1$. Then we use Theorem 1.11 to infer that $\#\text{SAT} \leq_{\text{AP}} \#\text{RET}(X(k_1, 1, 0))$ (since $X(k_1, 1, 0)$ has girth at least 5 and therefore is subject to Theorem 1.11). It follows that $\#\text{SAT} \leq_{\text{AP}} \#\text{RET}(H)$ by the same arguments as in the previous case.
- If for every pair u, v of adjacent vertices with $u \notin V(H^*)$ and $v \in V(H^*)$ we have $\deg_H(u) = 1$ (u is a so-called bristle) and $v \in \{c_1, \dots, c_{q-2}\}$, then H is

the triangle-extended path H^* together with a number of bristles (all of which are attached to a vertex in $\{c_1, \dots, c_{q-2}\}$). To match the notation of $\mathcal{H}_{\mathrm{BIS}}$ in Definition 4.2 we set $Q = q - 2$ and, for all $i \in \{0, \dots, Q+1\}$, we set $p_i = c_i$. Further, if $i \in \mathcal{I}$ then $K_i = \{p_i, d_i, p_{i+1}\}$ and otherwise $K_i = \{p_i, p_{i+1}\}$. Note that $|K_i| \in \{2, 3\}$ which we will use in a moment. For each $i \in [Q]$ let B_i be the set of unlooped neighbours (bristles) of p_i. By the fact that all unlooped vertices of H have degree 1 and a neighbour in $\{c_1, \dots, c_{q-2}\} = \{p_1, \dots, p_Q\}$ we have $V(H) = \bigcup_{i=0}^{Q} K_i \cup \bigcup_{i=1}^{Q} B_i$ and $E(H) = \bigcup_{i=0}^{Q}(K_i \times K_i) \cup \bigcup_{i=1}^{Q}(\{p_i\} \times B_i)$. Since H^* is a triangle-extended path we can also verify the properties $K_{i-1} \cap K_i = \{p_i\}$ (for $i \in [Q]$) and $K_i \cap K_j = \emptyset$ (for $i, j \in \{0, \dots, Q\}$ with $|j - i| > 1$).

Therefore, since $H \notin \mathcal{H}_{\mathrm{BIS}}$, there exists $i \in [Q]$ such that at least one of the following holds:

(i) $|K_{i-1}| = |K_i| = 2$ and $|B_i| \geq 2$.

(ii) $|K_{i-1}| = 2$ and $|K_i| = 3$ (or $|K_{i-1}| = 3$ and $|K_i| = 2$) and $|B_i| \geq 3$.

(iii) $|K_{i-1}| = |K_i| = 3$ and $|B_i| \geq 5$.

In all three cases we will show that the neighbourhood of p_i induces a #SAT-hard subgraph, i.e. that $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H[\Gamma_H(p_i)])$. Then, by Observation 4.15, we obtain $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$ which completes the proof of this case and with it the proof of the theorem.

- If item (i) holds, then $H[\Gamma_H(p_i)]$ is of the form $X(k_1, 2, 0)$ where $k_1 \geq 2$. The graph $X(k_1, 2, 0)$ has girth at least 5 and therefore is subject to Theorem 1.11. Since $X(k_1, 2, 0)$ with $k_1 \geq 2$ is a mixed graph but not a partially bristled reflexive path we obtain $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H[\Gamma_H(p_i)])$ by Theorem 1.11.
- If item (ii) holds, then $H[\Gamma_H(p_i)]$ is of the form $X(k_1, 1, 1)$ where $k_1 \geq 3$. Then $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H[\Gamma_H(p_i)])$ by Lemma 4.31.
- If item (iii) holds, then $H[\Gamma_H(p_i)]$ is of the form $X(k_1, 0, 2)$ where $k_1 \geq 5$. Then $\#\mathrm{SAT} \leq_{\mathrm{AP}} \#\textsc{Ret}(H[\Gamma_H(p_i)])$ by Lemma 4.32.

□

We will use the following remark to deal with graphs that have multiple connected components. It is a shorter version of Remark 3.3.

Remark 4.48. Let H be a graph with connected components $H_1, \dots, H_k$. On the one hand it holds that $\forall j \in [k], \#\textsc{Ret}(H_j) \leq_{\mathrm{AP}} \#\textsc{Ret}(H)$. On the other hand, given an oracle for each $\#\textsc{Ret}(H_j)$, we can construct a polynomial-time algorithm for $\#\textsc{Ret}(H)$.

Theorem 1.13. *Let H be a square-free graph.*

i) If every connected component of H is trivial then approximately counting retractions to H is in FP.

ii) Otherwise, if every connected component of H is

- *trivial,*
- *in the class* $\mathcal{H}_{\mathrm{BIS}}$*, or*
- *is an irreflexive caterpillar*

then approximately counting retractions to H is #BIS*-equivalent.*

iii) Otherwise, approximately counting retractions to H is #SAT*-equivalent.*

Proof. If H is a trivial graph then $\#\mathrm{LHom}(H) \in \mathrm{FP}$ by the result of Dyer and Greenhill [42] (see Theorem 7). From $\#\mathrm{Ret}(H) \leq_{\mathrm{AP}} \#\mathrm{LHom}(H)$ (Observation 1.18) it also follows that $\#\mathrm{Ret}(H) \in \mathrm{FP}$. Then item *i)* follows from Remark 4.48.

If H is a graph for which item *i)* does not hold, then H has a connected component H' that is not a trivial graph. By Remark 4.48 we have $\#\mathrm{Ret}(H') \leq_{\mathrm{AP}} \#\mathrm{Ret}(H)$. Then #BIS-hardness in item *ii)* follows from the reduction $\#\mathrm{Hom}(H') \leq_{\mathrm{AP}} \#\mathrm{Ret}(H')$ (Observation 1.18) together with the fact that $\#\mathrm{BIS} \leq_{\mathrm{AP}} \#\mathrm{Hom}(H')$ since H' is a non-trivial connected graph [62, Theorem 1].

We will now prove #BIS-easiness in item *ii)*. If H' is a trivial graph we have already pointed out that $\#\mathrm{Ret}(H') \in \mathrm{FP}$ and hence $\#\mathrm{Ret}(H')$ is trivially #BIS-easy. If $H' \in \mathcal{H}_{\mathrm{BIS}}$ then $\#\mathrm{Ret}(H') \leq_{\mathrm{AP}} \#\mathrm{BIS}$ by Theorem 1.12. If H' is an irreflexive caterpillar then $\#\mathrm{Ret}(H') \leq_{\mathrm{AP}} \#\mathrm{BIS}$ by Theorem 3.7. Hence, #BIS-easiness in item *ii)* follows from Remark 4.48.

If H is a graph for which item *ii)* does not hold, then H has a connected component H' that is not trivial, not a member of $\mathcal{H}_{\mathrm{BIS}}$ and not an irreflexive caterpillar. Then $\#\mathrm{Ret}(H')$ is approximation-equivalent to #SAT by Lemma 4.47 and $\#\mathrm{Ret}(H') \leq_{\mathrm{AP}} \#\mathrm{Ret}(H)$ by Remark 4.48. This proves item *iii)*. □

Part III

Modular Counting

Chapter 5

Counting Homomorphisms to K_4-minor-free Graphs, modulo 2

The popular view that scientists proceed inexorably from well-established fact to well-established fact, never being influenced by any unproved conjecture, is quite mistaken. Provided it is made clear which are proved facts and which are conjectures, no harm can result. Conjectures are of great importance since they suggest useful lines of research.

– Alan Turing, *Computing Machinery and Intelligence (1950)*

This chapter is based on the following preprint:

[56] Jacob Focke, Leslie Ann Goldberg, Marc Roth, and Stanislav Živný. Counting homomorphisms to K_4-minor-free graphs, modulo 2. *arXiv preprint arXiv:2006.16632*, 2020.

- A preliminary version of this work will appear in the Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2021.

Contents

Organisation of this Chapter

We start with the formal definitions that we need in Section 5.1.3; in particular we set up the framework of hardness gadgets. Section 5.2, our "toolbox", presents the most important class of hardness gadgets that we use.

Sections 5.3-5.6 constitute the proof of the main result of this chapter (Theorem 1.27) and should be considered the technical core of this chapter. In Section 5.3 we deal with biconnected K_4-minor-free graphs that are additionally chordal bipartite graphs (that is, they have the property that every induced cycle is a square). The reason for our separate treatment of these graphs is that our main gadget from Section 5.2 cannot be applied to graphs without an induced cycle of length $\neq 4$. We identify two families of such graphs, *impasses* and *diamonds*, that prevent us from constructing a local hardness gadget; examples of an impasse and a diamond are depicted in Figure 5.1.

After that, we dedicate Section 5.4 to the analysis of K_4-minor-free graphs that contain certain sequences of biconnected components, each of which is either an edge, an impasse, or a diamond. In Section 5.5 we consider biconnected K_4-minor-free graphs that are not necessarily chordal bipartite. We identify another family of graphs that does not allow for a local, i.e., an "internal", hardness gadget; we call such graphs *obstructions*; obstructions always contain an induced cycle of length other than 4, and an example of an obstruction is depicted in Figure 5.1.

In combination, Sections 5.4 and 5.5 reveal the structure of involution-free K_4-minor-free graphs that do not allow for a local hardness gadget. In Section 5.6 we use this structure, which allows us to constructively prove the existence of *global* hardness gadgets for all remaining K_4-minor-free graphs without non-trivial involutions. Our main theorem, Theorem 1.27, follows.

In Sections 5.7 and 5.8 we explore the applicability of our machinery to further classes of graphs and problems: Section 5.7 presents a full classification for counting homomorphisms to graphs of degree at most 3, modulo 2. Section 5.8 considers the problem of counting list homomorphisms, modulo 2, a variation of the homomorphism problem that generalises retractions as follows: Let H be a fixed graph. The problem $\oplus$LHom(H) expects as input a graph G and a function τ that maps every vertex of G to a list of vertices of H. The goal is then to compute the parity of the number of homomorphisms from G to H which additionally map every vertex v of G to a vertex contained in $\tau(v)$. We provide a full classification of $\oplus$LHom(H) for all graphs H, even if self-loops are allowed.

5.1 Introduction

5.1.1 Technical Overview

Given Theorem 1.24, we focus on the case where H is involution-free. In general, our proof proceeds in two steps. Given an involution-free K_4-minor-free graph H, we first try to find a biconnected component of H, let us call it B, that allows us to derive $\oplus$P-hardness of $\oplus$Hom(H) by exploiting the *local* structure of B to construct a reduction from counting independent sets, modulo 2. The latter problem, denoted by $\oplus$IS, is known to be $\oplus$P-complete [142] and cannot be solved in subexponential time, unless the rETH fails [31].

A careful analysis of biconnected and K_4-minor-free graphs, which crucially relies on the absence of non-trivial involutions, shows that the first step is always possible, unless all biconnected components of H have a very restricted form; examples are depicted in Figure 5.1. Note that all of these biconnected components have non-trivial involutions; consider for example the involution given by swapping the vertices x and y in Figure 5.1. Since the overall graph H is promised to be free of such involutions, we infer that at least one of x and y has a neighbour in a further biconnected component of H, which will allow us to successively construct a global "walk-like" structure in H that eventually yields a reduction from $\oplus$IS.

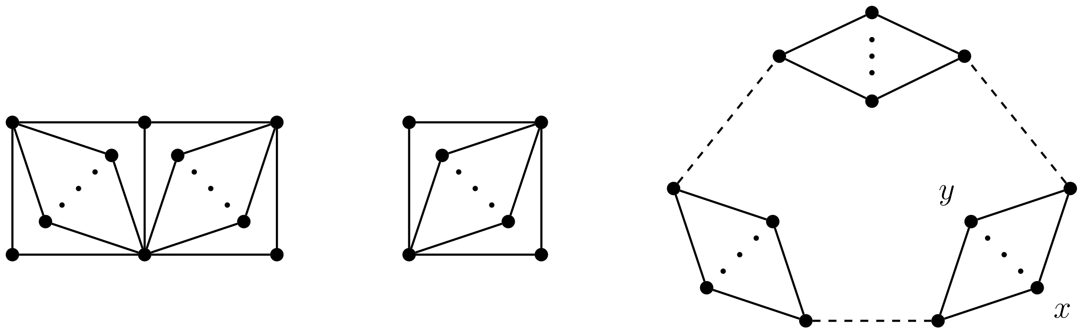
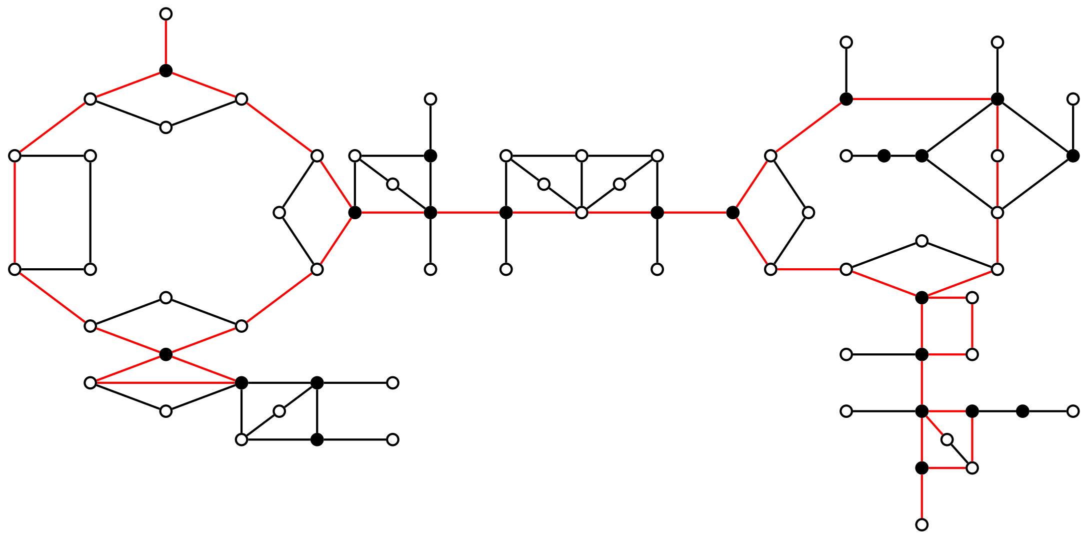


Figure 5.1: Examples of three types of biconnected and K_4-minor-free graphs that, if viewed as biconnected components, do not yield a reduction from ⊕IS. We will re-encounter those graphs as "impasses" *(left)*, "diamonds" *(centre)*, and "obstructions" *(right)*.

We consider the construction of those global substructures as our main technical contribution. While the formal specifications are beyond the scope of the introduction, we give an illustrated example which we hope gives some flavour of the graph theory that we will encounter in this chapter:



The above illustration depicts a K_4-minor-free graph H' without non-trivial involutions, together with a subgraph, highlighted in red, that allows for a reduction from ⊕IS. Solid vertices depict articulation points, i.e., vertices that lie in the intersection of at least 2 biconnected components. Note that each biconnected component of H' that is not an edge is of one of the three types given in Figure 5.1. Also, each biconnected component of H' has an involution, which prevents us, a priori, from achieving hardness of $\oplus\textsc{Hom}(H')$ if we only consider its biconnected components locally. Instead, we will see that the highlighted subgraph is what makes $\oplus\textsc{Hom}(H')$ hard.

In the next section we provide an overview of the general framework that allows us to reduce ⊕IS to $\oplus\textsc{Hom}(H)$. The structures used in such reductions are captured by the so-called hardness gadgets introduced by Göbel, Goldberg and Richerby [68, 69].

Prior applications of hardness gadgets could only be used to construct a reduction from ⊕IS to ⊕Hom(H) if H has certain local substructures, based around a path or a cycle. In contrast, our analysis will establish global walks such as the one highlighted in H'. As far as we can tell, none of the prior machinery [48, 68, 69, 107] is capable of proving the ⊕P-hardness of ⊕Hom(H'), however, this will follow as a result of our abstract consideration of global substructures of K_4-minor-free graphs.

5.1.2 Warm-up: useful Ideas from Previous Papers — Retractions and Hardness Gadgets

Instead of directly reducing ⊕IS to ⊕Hom(H), it is useful to consider the intermediate problem ⊕Ret(H), the problem of counting *retractions* to H, modulo 2. Given a graph H, a *partially H-labelled graph* $J = (G, \tau)$ consists of an *underlying graph* G and a corresponding *pinning function* τ, which is a partial function from $V(G)$ to $V(H)$. A homomorphism from J to H is a homomorphism h from G to H such that, for all vertices v in the domain of τ, $h(v) = \tau(v)$.

A homomorphism from a partially H-labelled graph J to H is also called a *retraction* to H because we can think of the pinning function τ as a way of identifying an induced subgraph H of G which must "retract" to H under the action of the homomorphism — see [49] for details. We use ⊕Ret(H) to denote the computational problem of computing the number of homomorphisms from J to H, modulo 2, given as input a partially H-labelled graph J.

It is known [69] that ⊕Ret(H) reduces to ⊕Hom(H) whenever H is involution-free. Since τ allows us to pin vertices of G to vertices of H arbitrarily, it is much easier to construct a reduction from ⊕IS to ⊕Ret(H) than to construct a direct reduction from ⊕IS to ⊕Hom(H).

Consider the following example. Suppose that H is the 4-vertex path (o, s, i, x) and that our goal is to reduce ⊕IS to ⊕Ret(H). Let G be an input to ⊕IS. That is, G is a graph whose independent sets we wish to count, modulo 2. For ease of presentation, suppose that G is bipartite,[1] that is, the vertices of G can be partitioned into two independent sets U and V. Let $\widehat{G}$ be the graph obtained from G by adding two additional vertices u and v, and by connecting u to all vertices in U, and v to all vertices in V, respectively. Let τ be the pinning function defined by $\tau(u) = s$ and $\tau(v) = i$. We provide an illustration of the construction in Figure 5.2.

Observe that any homomorphism φ from $(\widehat{G}, \tau)$ to H must map every vertex in U to either o or i, and every vertex in V to either s or x. Since H has no edge from o to x, the definition of homomorphism ensures that $\varphi^{-1}(o) \cup \varphi^{-1}(x)$ is an independent set of G. It is easy to verify that the function $\varphi \mapsto \varphi^{-1}(o) \cup \varphi^{-1}(x)$ is a bijection between the homomorphisms from $(\widehat{G}, \tau)$ to H and the independent sets of G, which gives a reduction from (bipartite) ⊕IS to ⊕Ret(H).

The observant reader might notice that the 4-vertex path has a non-trivial involution, and thus, we cannot further reduce ⊕Ret(H) to ⊕Hom(H) in this case.[2]

[1] The case of general graphs will be discussed later in the chapter.

[2] In fact, the problem ⊕Hom(H) is trivial when H is the 4-vertex path since the number of

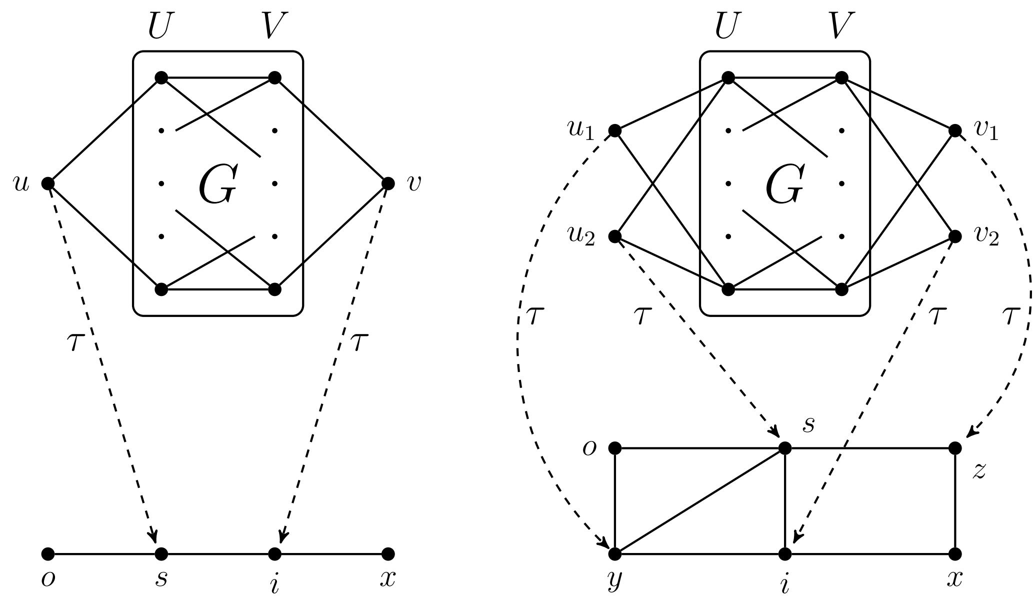


Figure 5.2: Illustration of the reduction from (bipartite) $\oplus$IS to $\oplus$RET(H) where H is the 4-vertex path *(left)*, and H is the graph H_2 from page 28 *(right)*.

However, the construction works for *any* graph H with an induced path (o, s, i, x) such that s and i each only have two neighbours.

The notion of a *hardness gadget*, which we formally introduce in Section 5.1.3, is essentially a generalisation of the previous construction. For example, we could substitute each of o, s, i and x with an odd number of copies, since we are only interested in the parity of the number of independent sets. Furthermore, we could identify o and x, since we only need the edge $\{o, x\}$ to be absent in H. A more sophisticated generalisation is obtained by observing that we can, to some extend, substitute the edges $\{o, s\}$, $\{s, i\}$ and $\{i, x\}$ with more complicated graphs, e.g. with length-2 paths, if we substitute the edges in $\widehat{G}$ accordingly. Finally, observe that the construction $(\widehat{G}, \tau)$ uses the partial function τ in a very simple manner: By adding a common neighbour u for all vertices in U and setting $\tau(u) = s$, the construction enforces the constraint that any homomorphism from $(\widehat{G}, \tau)$ to H must map every vertex in U to a neighbour of s. More sophisticated constructions will allow us to enforce much stronger constraints on homomorphisms. We will need this flexibility to construct reductions from $\oplus$IS to $\oplus$RET(H) for more general graphs H.

We conclude by making a generalisation explicit for one further example — the graph H_2 from page 28. We provide a more convenient drawing of H_2, including a labelling of its vertices and an illustration of the reduction in Figure 5.2. Again, we will assume for ease of presentation that the input G to $\oplus$IS is bipartite. To construct $\widehat{G}$, we add two additional vertices u_1 and u_2 and make them adjacent to every vertex in U. Similarly, we add two additional vertices v_1 and v_2 and make them adjacent to every vertex in V. Let τ be the pinning function defined by $\tau(u_1) = y$,

homomorphisms will always be even.

$\tau(u_2) = s$, $\tau(v_1) = z$, and $\tau(v_2) = i$.

Consider any homomorphism φ from $(\widehat{G}, \tau)$ to H_2. Since φ is edge-preserving, it must map every vertex in U to a common neighbour of s and y in H_2. Consequently, $\varphi(U) \subseteq \{o, i\}$. Similarly, we obtain $\varphi(V) \subseteq \{s, x\}$. Again, it is easy to see that the mapping $\varphi \mapsto \varphi^{-1}(o) \cup \varphi^{-1}(x)$ is a bijection between the homomorphisms from $(\widehat{G}, \tau)$ to H and the independent sets of G, which gives a reduction from (bipartite) $\oplus$IS to $\oplus$Ret(H).

Note that the second example, while being less straightforward than the first, is still a very simple reduction. The proof of Theorem 1.27 requires us to consider much more intricate "hardness gadgets"; the necessary tools will be carefully introduced in Sections 5.2 and 5.3.

5.1.3 Preliminaries

Given a positive integer q let $[q] = \{1, \dots, q\}$. Given a finite set S, we write $|S|$ for its cardinality.

Graph theory In contrast to previous chapters, in order to simplify notation, a *graph* in this chapter is assumed to be simple, i.e., it is irreflexive and without multiple edges (unless stated otherwise). The size of a graph G is defined as $|G| = |V(G)| + |E(G)|$. Given a graph H and a subset S of its vertices, we write $H[S]$ for the subgraph of H *induced* by S.

Given a non-negative integer k, a *walk* of length k in a graph H is a sequence of (not necessarily distinct) vertices $(v_0, \dots, v_k)$ such that, for all $i \in [k]$, $\{v_{i-1}, v_i\} \in E(H)$. The walk is *closed* if $v_0 = v_k$. Note that for $k = 0$, the single vertex (v_0) is a closed walk of length 0. A *path* of length k is a walk of length k for which $v_0, \dots, v_k$ are distinct. For $k \geq 3$, a *cycle* of length k is a closed walk of length k such that $v_1, \dots, v_k$ are distinct. A *square* is a cycle of length 4.

For $i, j \in \{0, \dots, k\}$ with $i \leq j$, we say that $(v_i, v_{i+1} \dots, v_j)$ is a *subwalk* of $(v_0, \dots, v_k)$. For vertices $u, v \in V(H)$, $\mathsf{dist}_H(u, v)$ is the length of a shortest path between u and v.

Definition 5.1 (chordal bipartite graph, see e.g. [83])**.** A graph in which every induced cycle is a square is called a *chordal bipartite graph*.

Given a graph H and a vertex $v \in V(H)$, we write $\Gamma_H(v)$ for the *neighbourhood* of v in H and we write $\deg_H(v)$ for the *degree* of v. That is, $\Gamma_H(v) = \{u \in V(H) \mid \{u, v\} \in E(H)\}$ and $\deg_H(v) = |\Gamma_H(v)|$. Given a subset S of $V(H)$, we set $\Gamma_H(S) = \bigcap_{v \in U} \Gamma_H(v)$ and note that $\Gamma_H(v) = \Gamma_H(\{v\})$.

Definition 5.2 (walk-neighbour-set)**.** Given a closed walk $W = (w_0, \dots, w_{q-1}, w_0)$ in a graph H we use $N_{W,H}(w_i)$ to denote $\Gamma_H(w_{i-1}) \cap \Gamma_H(w_{i+1})$, where the indices are taken modulo q. We refer to the sets $N_{W,H}(w_0), \dots, N_{W,H}(w_{q-1})$ as the *walk-neighbour-sets* of W in H.

Definition 5.3 (articulation point, biconnected, block-cut tree)**.** An *articulation point* of a graph is a vertex whose removal increases the number of connected components. A graph is *biconnected* if it has at least 2 vertices and has no articulation point. A *biconnected component* is a maximal biconnected subgraph.

Let H be a connected graph. The *block-cut tree* of H is the tree $\mathsf{BC}(H)$ that has a vertex for each biconnected component of H (such vertices are called *blocks*) and a vertex for each articulation point of H (such vertices are also called *cut vertices*) such that T has an edge between each biconnected component B and each articulation point a in B.

Partially labelled graphs Let H be a graph. Recall from Section 5.1.2 that a *partially H-labelled graph* $J = (G, \tau)$ consists of an *underlying graph* G and a corresponding *pinning function* τ, which is a partial function from $V(G)$ to $V(H)$. A vertex v in the domain of the pinning function is said to be *pinned*, *pinned to* $\tau(v)$, or a *$\tau(v)$-pin*. We write partial functions as sets of pairs, for example, writing $\tau = \{a \mapsto s, b \mapsto t\}$ for the partial function τ with domain $\{a, b\}$ such that a is an s-pin and b is a t-pin. The size of a partially H-labelled graph $J = (G, \tau)$ is defined as $|J| = |G|$.

Homomorphisms and Counting (mod 2) Let G and H be graphs. Then $\hom(G \to H)$ denotes the *set of homomorphisms* from G to H and $\hom(J \to H)$ denotes the set of homomorphisms from J to H.

It will sometimes be convenient to consider a graph G together with some number of distinguished vertices $x_1, \ldots, x_r$ of G. We denote such a graph by $(G, x_1, \ldots, x_r)$. The distinguished vertices need not be distinct. A homomorphism from a graph $(G, x_1, \ldots, x_r)$ to $(H, y_1, \ldots, y_r)$ is a homomorphism h from G to H with the property that, for each $i \in [r]$, $h(x_i) = y_i$. Correspondingly, we write $\hom((G, x_1, \ldots, x_r) \to (H, y_1, \ldots, y_r))$ for the set of such homomorphisms.

Given a partially labelled graph $J = (G, \tau)$ and distinguished vertices $x_1, \ldots, x_r$ of G that are not in the domain of τ, a homomorphism from $(J, x_1, \ldots, x_r)$ to $(H, y_1, \ldots, y_r)$ is a homomorphism from $J' = (G, \tau \cup \{x_1 \mapsto y_1, \ldots, x_r \mapsto y_r\})$ to H. The set of such homomorphisms is denoted by $\hom((J, x_1, \ldots, x_r) \to (H, y_1, \ldots, y_r))$.

Useful tools The following theorem of Göbel, Goldberg and Richerby will be of crucial importance in this chapter, as it will allow us to derive hardness of $\oplus\textsc{Hom}(H)$ from hardness of $\oplus\textsc{Ret}(H)$.

Theorem 5.4 ([69, Theorem 3.1])**.** *Let H be an involution-free graph. Then there is an algorithm with oracle access to $\oplus\textsc{Hom}(H)$ that takes as input a partially H-labelled graph J and computes $|\hom(J \to H)| \mod 2$ in time $\mathrm{poly}(|J|)$. The size of the input to every oracle query is $O(|J|)$.*

The statement of Theorem 5.4 in [69, Theorem 3.1] does not mention the fact that the size of the input to every oracle query is $O(|J|)$. Nevertheless, it is easy to see, by examining the proof in [69] that this linearity requirement is met (without making any changes to the proof). The reason that we introduce this linearity constraint is so that our hardness results can also rule out subexponential-time algorithms for $\oplus\textsc{Hom}(H)$ in the $\oplus$P-hard cases, using the rETH.

The following theorem of Faben and Jerrum will also be useful, as it will allow us to focus on connected graphs. The statement of [48, Theorem 6.1] does not mention the linearity requirement on the size of oracle queries, but this requirement does not present any difficulties. Faben and Jerrum's proof is given in a slightly different setting (pinning to orbits of vertices of H rather than to vertices) so, for completeness, we give a short proof.

Lemma 5.5 ([48, Theorem 6.1])**.** *Let H be an involution-free graph and let H' be a connected component of H. Then there exists an algorithm with oracle access to $\oplus\textsc{Hom}(H)$ that takes as input a graph G and computes $|\hom(G \to H')| \mod 2$ in time $\mathrm{poly}(|G|)$. The size of every oracle query is $O(|G|)$.*

Proof. Let G be a graph. If G is the empty graph then the algorithm returns 1, which is the number of homomorphisms from G to H'. Otherwise, there exists a vertex $u \in V(G)$. For each $v \in V(H')$ we define the partially H'-labelled $J_v = (G, \{u \mapsto v\})$. Note that $|\hom(G \to H')| = \sum_{v \in V(H')} |\hom(J_v \to H)|$.

By Theorem 5.4, there is an algorithm A with oracle access to $\oplus\textsc{Hom}(H)$ that takes as input a partially H-labelled graph J and computes $|\hom(J \to H)| \mod 2$ in time $\mathrm{poly}(|J|)$ such that the size of every oracle query is bounded by $O(|J|)$. Our algorithm uses algorithm A as a subroutine to compute the parity of $|\hom(J_v \to H)|$ for each $v \in V(H')$. This requires $|V(H')|$ executions of the subroutine A. Thus, the algorithm runs in time

$$O\Big(\sum_{v \in V(H')} \mathrm{poly}(|J_v|)\Big) = \mathrm{poly}(|G|).$$

Moreover, for each $v \in V(H')$, the size of each $\oplus\textsc{Hom}(H)$ oracle query is bounded by $O(|J_v|) = O(|G|)$. □

Hardness Gadgets The following is a slightly generalised version of the *hardness gadget* introduced in [69, Definition 4.1]. The only difference between their definition and ours is that they require the sets I and S to have size 1.

Definition 5.6. [69, Definition 4.1] A *hardness gadget* $(I, S, (J_1, y), (J_2, z), (J_3, y, z))$ for a graph H consists of odd-cardinality sets $I, S \subseteq V(H)$ together with three connected, partially H-labelled graphs with distinguished vertices (J_1, y), (J_2, z) and (J_3, y, z) that satisfy certain properties as explained below. Let

$$\begin{aligned}\Omega_y &= \{a \in V(H) \mid |\hom((J_1, y) \to (H, a))| \text{ is odd}\},\\ \Omega_z &= \{b \in V(H) \mid |\hom((J_2, z) \to (H, b))| \text{ is odd}\}, \text{ and}\\ \Sigma_{a,b} &= \hom((J_3, y, z) \to (H, a, b)).\end{aligned}$$

The properties that we require are the following.

1. $|\Omega_y|$ is even and $I \subset \Omega_y$.

2. $|\Omega_z|$ is even and $S \subset \Omega_z$.

3. For each $i \in I$, $o \in \Omega_y \setminus I$, $s \in S$ and each $x \in \Omega_z \setminus S$,

   - $|\Sigma_{o,x}|$ is even.
   - $|\Sigma_{i,s}|$, $|\Sigma_{o,s}|$ and $|\Sigma_{i,x}|$ are odd.

The following theorem of Göbel, Goldberg and Richerby establishes intractability of $\oplus\text{Ret}(H)$ whenever H has a hardness gadget.

Theorem 5.7 ([69, Theorem 4.2]). *Let H be an involution-free graph that has a hardness gadget. Then $\oplus\text{Ret}(H)$ is $\oplus$P-hard. Also, assuming the randomised Exponential Time Hypothesis, $\oplus\text{Ret}(H)$ cannot be solved in time $\exp(o(|J|))$.*

Proof. Although the hardness gadgets from [69] are more constrained than the ones that we use, the proof of [69, Theorem 4.2] establishes the $\oplus$P-hardness in Theorem 5.7 with only very minor changes, which we now describe.

As noted in the introduction, Valiant [142] showed that the problem $\oplus$IS is $\oplus$P-complete. The proof of [69, Theorem 4.2] gives a polynomial-time Turing reduction from $\oplus$IS to $\oplus\text{Ret}(H)$. The reduction uses G and the hypothesised hardness gadget for H to construct a partially H-labelled graph J such that the number of independent sets of G, which we denote $|\mathcal{I}(G)|$, is equal, modulo 2, to $|\hom(J \to H)|$. The reduction concludes by making a single oracle call to $\oplus\text{Ret}(H)$ with input J.

In our case, the construction of J is exactly as it is in [69]. The proof that $|\mathcal{I}(G)| = |\hom(J \to H)| \mod 2$ needs only a very minor modification to account for the fact that the sets I and S in the hardness gadget may have more than one element. At some point in the proof of [69], it is argued that a certain quantity $n(a, a')$ is even if a and a' are both in I, and odd otherwise. This is still true even when I and S contain more than one element — it follows from item 3 in the definition of hardness gadget (and from the fact that I and S have odd cardinality).

The final sentence in the statement of Theorem 5.7, asserting that $\oplus\text{Ret}(H)$ cannot be solved in time $\exp(o(|J|))$ unless the rETH fails, was not contained in the original theorem of [69], however it follows immediately from the fact that $|J| = O(|G|)$ (which is easily checked) and from the fact that $\oplus$IS cannot be solved in time $\exp(o(|G|))$, unless the rETH fails, which was proved by Dell, Husfeldt, Marx, Taslaman and Wahlen [31]. In more detail, Dell et al. established that counting independent sets cannot be done in time $\exp(o(|E(G)|))$, unless the rETH fails [31, Theorem 1.2]. They point out explicitly that their reduction also works in the case of counting modulo 2. Furthermore, their reduction always yields a graph without isolated vertices — for such graphs we have $|E(G)| = \Theta(|G|)$. □

5.2 Toolbox

5.2.1 Path Gadget

We will use the following path gadget, which is called a "caterpillar gadget" in [69].

Definition 5.8. Given a path $P = (v_0, \dots, v_q)$ in H with $q \geq 1$, define the *path gadget* $J_P = (G, \tau)$ as follows. $V(G) = \{u_1, \dots, u_{q-1}, w_1, \dots, w_{q-1}, y, z\}$ and G is the path $(y, u_1, \dots, u_{q-1}, z)$ together with edges $\{u_j, w_j\}$ for $j \in [q-1]$. $\tau = \{w_1 \mapsto v_1, \dots, w_{q-1} \mapsto v_{q-1}\}$.

We will use the following lemma of Göbel, Goldberg and Richerby. The original lemma was stated for square-free graphs, but the proof only uses the fact that no edge of P is part of a square in H.

Lemma 5.9 ([69, Lemma 4.5])**.** *For an integer $q \geq 1$, let $P = (v_0, \dots, v_q)$ be a path in a graph H. Suppose that no edge of P is part of a square in H and that $\deg_H(v_j)$ is odd for all $j \in [q-1]$. Let $\Omega_y \subseteq \Gamma_H(v_0)$ and $\Omega_z \subseteq \Gamma_H(v_q)$, with $I = \{v_1\} \subset \Omega_y$ and $S = \{v_{q-1}\} \subset \Omega_z$. For $i = v_1$, $s = v_{q-1}$ and for each $o \in \Omega_y \setminus I$ and $x \in \Omega_z \setminus S$ we have the following:*

- $|\hom((J_p, y, z) \to (H, o, x))| = 0$,
- $|\hom((J_p, y, z) \to (H, o, s))| = 1$,
- $|\hom((J_p, y, z) \to (H, i, x))| = 1$, *and*
- $|\hom((J_p, y, z) \to (H, i, s))|$ *is odd.*

5.2.2 Cycle Gadget

We will use the following cycle gadget, which is a generalisation of the cycle gadget in [107].

Definition 5.10 (Cycle gadget)**.** For an integer $q \geq 3$, let $\mathcal{C} = (\mathcal{C}_0, \dots, \mathcal{C}_{q-1})$ where, for $i = 0, \dots, q-1$, s_i is a positive integer and $\mathcal{C}_i = \{c_i^1, \dots, c_i^{s_i}\}$ is a set of s_i vertices. We define the *cycle gadget* $J_{\mathcal{C}} = (G, \tau)$ as follows (see Figure 5.3). For $i = 0, \dots, q-1$, let $U_i = \{u_i^1, \dots, u_i^{s_i}\}$ be a set of s_i vertices. Then $V(G) = \{v_0, \dots, v_{q-1}\} \cup U_0 \cup \dots \cup U_{q-1}$ (where all named vertices are assumed to be distinct) and $E(G) = \{\{v_i, v_{i+1 \bmod q}\} \mid i \in \{0, \dots, q-1\}\} \cup \{\{v_i, u_i^j\} \mid i \in \{0, \dots, q-1\}, j \in \{1, \dots, s_i\}\}$. $\tau = \{u_i^j \mapsto c_i^j \mid \forall i \in \{0, \dots q-1\}, j \in \{1, \dots s_i\}\}$.

In fact, we will also need a further generalisation of the cycle gadget from Definition 5.10.

Definition 5.11 (Generalised cycle gadget)**.** Let H be a graph. For an integer $q \geq 3$, let $\mathcal{C} = (\mathcal{C}_0, \dots, \mathcal{C}_{q-1})$ where, for $i = 0, \dots, q-1$, s_i is a positive integer and $\mathcal{C}_i = \{c_i^1, \dots, c_i^{s_i}\}$ is a set of s_i vertices of H. Let $J_{\mathcal{C}}$ be the cycle gadget from Definition 5.10. Let $(J_0, z_0), \dots, (J_{q-1}, z_{q-1})$ be partially H-labelled graphs with distinguished vertices. Then the *generalised cycle gadget* $J(J_{\mathcal{C}}, J_0, \dots, J_{q-1})$ is the gadget obtained from $J_{\mathcal{C}}, J_0, \dots, J_{q-1}$ by identifying, for each $i \in \{0, \dots, q-1\}$ the vertex v_i from $J_{\mathcal{C}}$ with the vertex z_i from J_i. Intuitively, it is the cycle gadget where at each vertex v_i in addition we attach a gadget J_i.

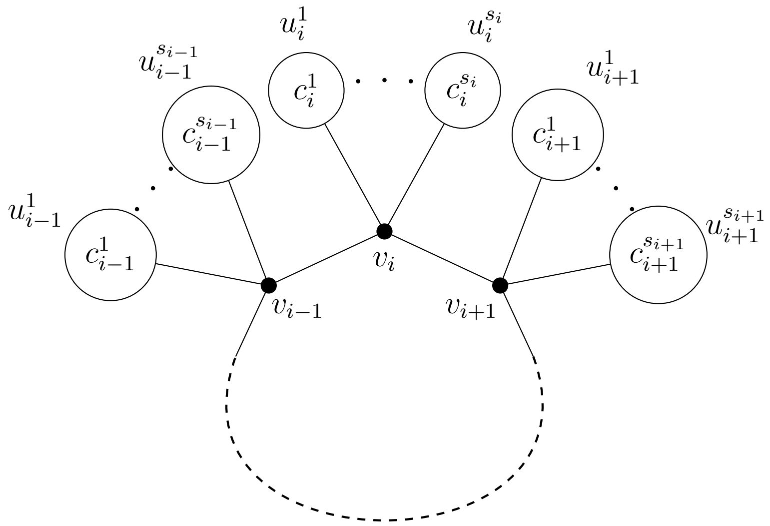


Figure 5.3: The cycle gadget $J_{\mathcal{C}}$. Vertices of the form u_i^j are c_i^j-pins.

Lemma 5.12. *For an integer $q \geq 3$, let H be a graph which contains sets of vertices $\mathcal{C}_0, \ldots, \mathcal{C}_{q-1}$ (not necessarily disjoint or even distinct). Let $(J_0, z_0), \ldots, (J_{q-1}, z_{q-1})$ be partially H-labelled graphs with distinguished vertices, and, for each $i \in \{0, \ldots, q-1\}$, let $\Omega_i = \{a \in V(H) \mid |\hom((J_i, z_i) \to (H, a))|$ is odd$\}$. Suppose that for all $i \in \{0, \ldots, q-1\}$ we have the following.*

- *(L5.12.1) $|\mathcal{C}_{i-1 \bmod q} \cap \Omega_i|$ and $|\mathcal{C}_{i+1 \bmod q} \cap \Omega_i|$ are odd.*
- *(L5.12.2) If $w \in \mathcal{C}_{i-1 \bmod q}$ then $\Gamma_H(w) \cap \Gamma_H(\mathcal{C}_{i+1 \bmod q}) = \mathcal{C}_i$.*
- *(L5.12.3) If $w \in \mathcal{C}_{i+1 \bmod q}$ then $\Gamma_H(\mathcal{C}_{i-1 \bmod q}) \cap \Gamma_H(w) = \mathcal{C}_i$.*
- *(L5.12.4) There is no walk of the form $D = (d_0, \ldots, d_{q-1}, d_0)$ such that, for all $i \in \{0, \ldots, q-1\}$, $d_i \in \Gamma_H(\mathcal{C}_i) \setminus (\mathcal{C}_{i-1 \bmod q} \cup \mathcal{C}_{i+1 \bmod q})$.*

Let $J_{\mathcal{C}}$ be the cycle gadget (Definition 5.10) and let $J^* = J(J_{\mathcal{C}}, J_0, \ldots, J_{q-1})$ be the generalised cycle gadget (Definition 5.11) Then, for all $k \in \{0, \ldots, q-1\}$,

$$\{a \in V(H) \mid |\hom((J^*, v_k) \to (H, a))| \text{ is odd}\} = (\mathcal{C}_{k-1 \bmod q} \cup \mathcal{C}_{k+1 \bmod q}) \cap \Omega_k.$$

Proof. To simplify notation, all indices in this proof are considered to be modulo q. For $a \in V(H)$, let $k \in \{0, \ldots, q-1\}$ and $h \in \hom((J^*, v_k) \to (H, a))$. By construction of J^* and the fact that h has to preserve edges, for all $i \in \{0, \ldots, q-1\}$, we obtain

- $h(v_i) \in \Gamma_H(\mathcal{C}_i)$,
- $h(v_i) \notin \mathcal{C}_i$ (since we do not allow self-loops in H),
- $h(v_i)$ is adjacent to $h(v_{i+1})$ in H,
- $h(v_i) \neq h(v_{i+1})$.

Consequently, it holds that $h(v_{i+1}) \in \Gamma_H(h(v_i)) \cap \Gamma_H(\mathcal{C}_{i+1})$. Suppose, for some $i \in \{0, \ldots, q-1\}$, that $h(v_i) \in \mathcal{C}_{i-1}$. Then, by (L5.12.2), we have $h(v_{i+1}) \in \mathcal{C}_i$. Therefore,

$$\text{If } h(v_i) \in \mathcal{C}_{i-1} \text{ then } h(v_{i+1}) \in \mathcal{C}_i. \tag{5.1}$$

Analogously, using (L5.12.3),

$$\text{If } h(v_i) \in \mathcal{C}_{i+1} \text{ then } h(v_{i-1}) \in \mathcal{C}_i. \tag{5.2}$$

Thus, if there exists some $\ell \in \{0, \ldots, q-1\}$ such that $h(v_\ell) \in \mathcal{C}_{\ell-1}$ then we can use (5.1) iteratively to obtain $h(v_i) \in \mathcal{C}_{i-1}$ for all $i \in \{0, \ldots, q-1\}$. In particular, $h(v_k) \in \mathcal{C}_{k-1}$. Analogously, if there exists some $\ell \in \{0, \ldots, q-1\}$ such that $h(v_\ell) \in \mathcal{C}_{\ell+1}$ then we can use (5.2) iteratively to obtain $h(v_i) \in \mathcal{C}_{i+1}$ for all $i \in \{0, \ldots, q-1\}$. This means that $h(v_k) \in \mathcal{C}_{k+1}$.

Suppose that $h(v_k) \notin \mathcal{C}_{k-1} \cup \mathcal{C}_{k+1}$. We have established that, using (5.1) and (5.2) iteratively, we obtain, for all $i \in \{0, \ldots, q-1\}$, $h(v_i) \notin \mathcal{C}_{i-1} \cup \mathcal{C}_{i+1}$ and consequently $h(v_i) \in \Gamma_H(\mathcal{C}_i) \setminus (\mathcal{C}_{i-1} \cup \mathcal{C}_{i+1})$. However, $(h(v_0), \ldots, h(v_{q-1}), h(v_0))$ is a walk in H, which gives a contradiction to (L5.12.4).

We have shown that $h(v_k) \in \mathcal{C}_{k-1} \cup \mathcal{C}_{k+1}$. Moreover, for each $a \in \mathcal{C}_{k-1}$, $|\mathrm{hom}((J^*, v_k) \to (H, a))| = |\mathrm{hom}((J_k, z_k) \to (H, a))| \cdot \prod_{i \in \{0,\ldots,q-1\} \setminus \{k\}} |\mathcal{C}_{i-1} \cap \Omega_i|$, which is odd if and only if $a \in \mathcal{C}_{k-1} \cap \Omega_k$ by (L5.12.1). The statement for $a \in \mathcal{C}_{k+1}$ is analogous. □

Lemma 5.13. *For an integer $q \geq 3$, let H be a graph which contains sets of vertices $\mathcal{C}_0, \ldots, \mathcal{C}_{q-1}$ (not necessarily disjoint or even distinct). Let $(J_0, z_0), \ldots, (J_{q-1}, z_{q-1})$ be partially H-labelled graphs with distinguished vertices, and, for each $i \in \{0, \ldots, q-1\}$, let $\Omega_i = \{a \in V(H) \mid |\mathrm{hom}((J_i, z_i) \to (H, a))| \text{ is odd}\}$. Suppose that for all $i \in \{0, \ldots, q-1\}$ we have the following properties from the statement of Lemma 5.12.*

- *(L5.12.1) $|\mathcal{C}_{i-1 \bmod q} \cap \Omega_i|$ and $|\mathcal{C}_{i+1 \bmod q} \cap \Omega_i|$ are odd.*
- *(L5.12.2) If $w \in \mathcal{C}_{i-1 \bmod q}$ then $\Gamma_H(w) \cap \Gamma_H(\mathcal{C}_{i+1 \bmod q}) = \mathcal{C}_i$.*
- *(L5.12.3) If $w \in \mathcal{C}_{i+1 \bmod q}$ then $\Gamma_H(\mathcal{C}_{i-1 \bmod q}) \cap \Gamma_H(w) = \mathcal{C}_i$.*
- *(L5.12.4) There is no walk of the form $D = (d_0, \ldots, d_{q-1}, d_0)$ such that, for all $i \in \{0, \ldots, q-1\}$, $d_i \in \Gamma_H(\mathcal{C}_i) \setminus (\mathcal{C}_{i-1 \bmod q} \cup \mathcal{C}_{i+1 \bmod q})$.*

Furthermore, there exists $k \in \{0, \ldots, q-1\}$ such that

- *(L5.13.1) there are no edges between $\mathcal{C}_k$ and $\mathcal{C}_{k+3 \bmod q}$,*
- *(L5.13.2) $|(\mathcal{C}_k \cup \mathcal{C}_{k+2 \bmod q}) \cap \Omega_{k+1}|$ and $|(\mathcal{C}_{k+1 \bmod q} \cup \mathcal{C}_{k+3 \bmod q}) \cap \Omega_{k+2}|$ are even.*

Then H has a hardness gadget.

Proof. To simplify notation all indices in this proof are considered to be modulo q. We construct a hardness gadget $(I, S, (J_1', y), (J_2', z), (J_3', y, z))$ for H, as defined in Definition 5.6.

Let $\mathcal{C} = (\mathcal{C}_0, \ldots, \mathcal{C}_{q-1})$. Let J_1' and J_2' each be an instance of the generalised cycle gadget $J(J_\mathcal{C}, J_0, \ldots, J_{q-1})$, let $y = v_{k+1}$, and let $z = v_{k+2}$. Then we have $\Omega_y = (\mathcal{C}_k \cup \mathcal{C}_{k+2}) \cap \Omega_{k+1}$ and $\Omega_z = (\mathcal{C}_{k+1} \cup \mathcal{C}_{k+3}) \cap \Omega_{k+2}$ by Lemma 5.12. It follows that $|\Omega_y|$ and $|\Omega_z|$ are even by (L5.13.2). Let $I = \mathcal{C}_{k+2} \cap \Omega_{k+1}$ and $S = \mathcal{C}_{k+1} \cap \Omega_{k+2}$. We note that I and S have odd size by (L5.12.1) and that $I \subset \Omega_y$ and $S \subset \Omega_z$.

Let J_3 be an edge from y to z. For each $o \in \Omega_y \setminus I \subseteq \mathcal{C}_k$, $s \in S \subseteq \mathcal{C}_{k+1}$, $i \in I \subseteq \mathcal{C}_{k+2}$ and $x \in \Omega_z \setminus S \subseteq \mathcal{C}_{k+3}$,

- $|\Sigma_{ox}| = 0$ since no edge exists between $\mathcal{C}_k$ and $\mathcal{C}_{k+3}$ according to (L5.13.1).
- $|\Sigma_{is}| = |\Sigma_{ix}| = |\Sigma_{os}| = 1$ since, by (L5.12.2), for all $\ell \in \{0, \dots, q-1\}$ we have $\mathcal{C}_\ell \subseteq \Gamma_H(\mathcal{C}_{\ell+1})$.

□

We point out a corollary which is more easily accessible and does not use the full generality of the gadget $J(J_\mathcal{C}, J_0, \dots, J_{q-1})$ but rather only uses the cycle gadget $J_\mathcal{C}$.

Corollary 5.14. *For an integer $q = 3$ or $q \geq 5$, let H be a graph which contains a cycle $C = c_0, \dots, c_{q-1}, c_0$ such that*

- *for all $i \in \{0, \dots, q-1\}$, we have $|N_{C,H}(c_i)| = 1$, and*
- *there is no walk of the form $D = d_0, \dots, d_{q-1}, d_0$ with $d_i \in \Gamma_H(c_i) \setminus (c_{i-1} \cup c_{i+1})$ ($\forall i \in \{0, \dots, q-1\}$).*

Then H has a hardness gadget.

Proof. All indices in this proof are considered to be modulo q. For $i \in \{0, \dots, q-1\}$ we choose $\mathcal{C}_i = N_{C,H}(c_i)$, which by the fact that $|N_{C,H}(c_i)| = 1$ implies $\mathcal{C}_i = \{c_i\}$. We choose $k = 0$. For each $i \in \{0, \dots, q-1\}$, let (J_i, z_i) be the partially H-labelled graph that only contains the single vertex z_i and has an empty pinning function. It follows that $\Omega_i = V(H)$ and that $J(J_\mathcal{C}, J_0, \dots, J_{q-1})$ is essentially $J_\mathcal{C}$. We check that the requirements of Lemma 5.13 are met. (L5.12.1) holds since $\mathcal{C}_{i-1} \cap \Omega_i = \mathcal{C}_{i-1} = \{c_{i-1}\}$ and $\mathcal{C}_{i+1} \cap \Omega_i = \mathcal{C}_{i+1} = \{c_{i+1}\}$. (L5.12.2) and (L5.12.3) hold since $|N_{C,H}(c_i)| = 1$ and therefore c_i is the only common neighbour of c_{i-1} and c_{i+1}. There is no walk of the form $D = d_0, \dots, d_{q-1}, d_0$ with $d_i \in \Gamma_H(c_i) \setminus (c_{i-1} \cup c_{i+1})$, as required by (L5.12.4). Since $q \geq 3$ and C is a cycle, the vertices c_0, c_1, c_2 are distinct. If $q = 3$, as C is a cycle, we have $c_0 = c_3$, and (L5.13.1) holds since we do not allow self-loops in H. If otherwise $q \geq 5$ then (L5.13.1) holds since $\Gamma_H(c_1) \cap \Gamma_H(c_3) = N_{C,H}(c_2) = \{c_2\}$ and therefore c_0 (which is a neighbour of c_1) cannot be a neighbour of c_3. Since $q \geq 3$ (L5.13.2) holds as $(\mathcal{C}_0 \cup \mathcal{C}_2) \cap \Omega_1 = \{c_0, c_2\}$ and $(\mathcal{C}_1 \cup \mathcal{C}_3) \cap \Omega_2 = \{c_1, c_3\}$ are sets of 2 distinct vertices. □

Remark 5.15. Suppose that a square-free graph H contains a cycle C. Clearly, the requirements of Corollary 5.14 are met and, by Theorem 5.7, we obtain $\oplus$P-hardness for $\oplus$Ret(H). If, in addition, H is involution-free $\oplus$P-hardness carries over to $\oplus$Hom(H) by Theorem 5.4 (from [69, Theorem 3.1]). This argument, together with the classification of $\oplus$Hom(H) for trees by Faben and Jerrum [48] (or alternatively the shorter [69, Lemmas 5.1 and 5.3]) implies the dichotomy for square-free graphs presented in [69].

5.3 Chordal Bipartite Components

Our main strategy for proving $\oplus$P-hardness of $\oplus$HOM(H) for K_4-minor-free graphs will rely on finding induced cycles whose lengths are not equal to 4. However, this requires us to treat the case of (K_4-minor-free) graphs that include *only* squares as induced cycles separately; recall that such graphs are called chordal bipartite graphs.

In the current section we will construct a hardness gadget for every involution-free, K_4-minor-free, biconnected chordal bipartite graph H, unless H has a very restricted form. In this restricted case we call H an *impasse* (which will be formally defined in Definition 5.30). The main tool that we use to construct hardness gadgets relies on two squares that share one edge. More formally, we will consider the following graph:

Definition 5.16 (The graph F, $\Gamma_{H\setminus F}(i,j)$)**.** The graph F is defined to be the graph depicted in Figure 5.4.

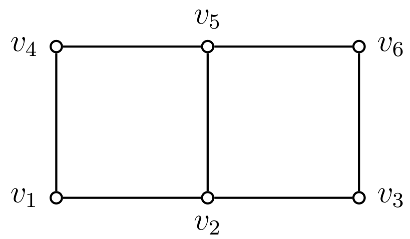


Figure 5.4: The graph F.

Given a graph H that contains F as a subgraph, and $i \neq j \in [6]$, we define

$$\Gamma_{H\setminus F}(i,j) = (\Gamma_H(v_i) \cap \Gamma_H(v_j)) \setminus V(F)\,.$$

Definition 5.17 (Type V)**.** Let H be a K_4-minor-free graph that contains F as a subgraph. We say that F has type V in H if one of the following is true

- $\Gamma_{H\setminus F}(1,5)$ and $\Gamma_{H\setminus F}(3,5)$ are non-empty and $\Gamma_{H\setminus F}(2,4)$ and $\Gamma_{H\setminus F}(2,6)$ are empty.
- $\Gamma_{H\setminus F}(2,4)$ and $\Gamma_{H\setminus F}(2,6)$ are non-empty and $\Gamma_{H\setminus F}(1,5)$ and $\Gamma_{H\setminus F}(3,5)$ are empty.

An illustration of the latter case is given in Figure 5.5.

The following observation will be useful in the remainder of this section:

Lemma 5.18. *Let H be a K_4-minor-free graph containing F as a subgraph. At least one of $\Gamma_{H\setminus F}(1,5)$ and $\Gamma_{H\setminus F}(2,4)$ is empty, and at least one of $\Gamma_{H\setminus F}(2,6)$ and $\Gamma_{H\setminus F}(3,5)$ is empty.*

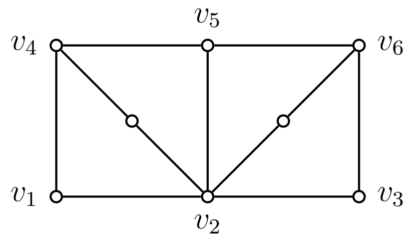


Figure 5.5: A K_4-minor-free graph containing F of type V.

Proof. If $\Gamma_{H\setminus F}(1,5)$ and $\Gamma_{H\setminus F}(2,4)$ are both non-empty, then the vertices v_1, v_2, v_4 and v_5 yield a K_4-minor. If $\Gamma_{H\setminus F}(2,6)$ and $\Gamma_{H\setminus F}(3,5)$ are both non-empty, then the vertices v_2, v_3, v_5 and v_6 yield a K_4-minor. □

Lemma 5.19. *Let H be a K_4-minor-free graph containing F as a subgraph. If F does not have type V in H then either $\Gamma_{H\setminus F}(1,5) = \Gamma_{H\setminus F}(2,6) = \emptyset$ or $\Gamma_{H\setminus F}(2,4) = \Gamma_{H\setminus F}(3,5) = \emptyset$.*

Proof. Note that either $\Gamma_{H\setminus F}(2,6)$ or $\Gamma_{H\setminus F}(3,5)$ are empty by Lemma 5.18. Assume w.l.o.g. that the former is empty; the other case is symmetric. We distinguish two cases:

(I) $\Gamma_{H\setminus F}(3,5) \neq \emptyset$. Now assume for contradiction that $\Gamma_{H\setminus F}(1,5) \neq \emptyset$. Then, again by Lemma 5.18, we obtain $\Gamma_{H\setminus F}(2,4) = \emptyset$, which implies that F has type V in H, yielding the desired contradiction. In combination with the previous assumption, we thus have $\Gamma_{H\setminus F}(1,5) = \Gamma_{H\setminus F}(2,6) = \emptyset$.

(II) $\Gamma_{H\setminus F}(3,5) = \emptyset$. By Lemma 5.18 we have that either $\Gamma_{H\setminus F}(1,5)$ or $\Gamma_{H\setminus F}(2,4)$ is empty. This concludes the proof as the current case provides additionally $\Gamma_{H\setminus F}(3,5) = \emptyset$ and $\Gamma_{H\setminus F}(2,6) = \emptyset$.

□

Lemma 5.20. *Let H be a K_4-minor-free graph containing F as a subgraph. Then H has a hardness gadget, unless F has type V in H.*

Proof. Using Lemma 5.19 and the fact that H is K_4-minor free, we can w.l.o.g. assume that

(a) The edges $\{v_1, v_6\}$ and $\{v_3, v_4\}$ are *not* present in H as, otherwise, we obtain a K_4-minor.

(b) $\Gamma_H(v_1) \cap \Gamma_H(v_5) = \{v_2, v_4\}$.

(c) $\Gamma_H(v_2) \cap \Gamma_H(v_6) = \{v_3, v_5\}$.

This allows us to construct a hardness gadget:

- $S = \{v_5\}$ and $I = \{v_2\}$.

- J_1 is the graph where y is adjacent to a v_1-pin and a v_5-pin Note that $\Omega_y = \{v_2, v_4\}$ by (b).
- J_2 is the graph where z is adjacent to a v_2-pin and a v_6-pin. Note that $\Omega_z = \{v_3, v_5\}$ by (c).
- J_3 is just the edge $\{y, z\}$.

We have $|\Sigma_{v_4,v_5}| = |\Sigma_{v_5,v_2}| = |\Sigma_{v_2,v_3}| = 1$. Furthermore, $|\Sigma_{v_4,v_3}| = 0$ by (a). □

Definition 5.21 (The graph $S_{k,\ell}$)**.** For positive integers k and ℓ, $S_{k,\ell}$ is the graph depicted in Figure 5.6.

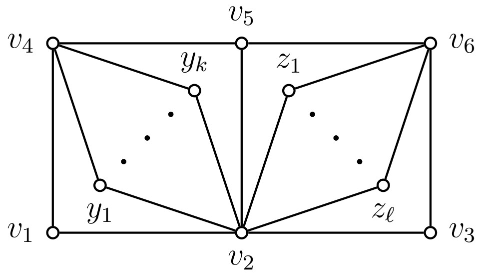


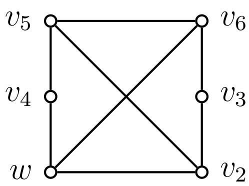
Figure 5.6: The graph $S_{k,\ell}$.

Lemma 5.22. *Let H be a K_4-minor-free graph containing F as a subgraph. If F has type* V *in H and $|\Gamma_{H\setminus F}(1,5)|$ and $|\Gamma_{H\setminus F}(2,4)|$ are even, then H has a hardness gadget.*

Proof. As F has type V in H we can assume w.l.o.g. that $\Gamma_{H\setminus F}(2,4) \neq \emptyset$ and $\Gamma_{H\setminus F}(2,6) \neq \emptyset$, and that $\Gamma_{H\setminus F}(1,5) = \Gamma_{H\setminus F}(3,5) = \emptyset$; the other case is symmetric. In other words, there exist positive integers k and ℓ such that H contains the subgraph $S_{k,\ell}$ (Definition 5.21) with $\Gamma_{H\setminus F}(2,4) = \{y_1, \ldots, y_k\}$ and $\Gamma_{H\setminus F}(2,6) = \{z_1, \ldots, z_\ell\}$. By the premise of the lemma, k must be even. We will emphasise some crucial properties of H:

(a) $\Gamma_H(v_3) \cap \Gamma_H(v_5) = \{v_2, v_6\}$, since $\Gamma_{H\setminus F}(3,5) = \emptyset$.

(b) v_6 is not adjacent to any vertex in $\{y_1, \ldots, y_k, v_1\}$: Assuming otherwise, let $w \in \{y_1, \ldots, y_k, v_1\}$ be adjacent to v_6. We obtain the following K_4-minor of H:



We proceed by constructing a hardness gadget:

- $S = \{v_2\}$ and $I = \{v_5\}$.

- J_1 is the graph where y is adjacent to a v_2-pin and a v_4-pin. Note that

$$\Omega_y = \{v_1, v_5\} \cup \Gamma_{H \setminus F}(2,4) = \{v_1, v_5, y_1, \dots, y_k\}\,.$$

In particular, $|\Omega_y|$ is even as k is.

- J_2 is the graph where z is adjacent to a v_3-pin and a v_5-pin. Note that $\Omega_z = \{v_2, v_6\}$ by (a).

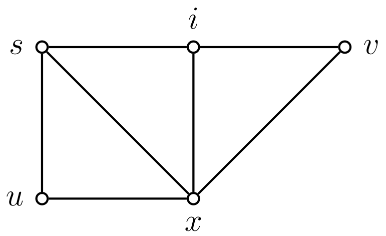
- J_3 is just the edge $\{y, z\}$.

We have $|\Sigma_{v_2,v_5}| = |\Sigma_{v_5,v_6}| = 1$ and, for every $o \in \Omega_y \setminus \{v_5\}$, $|\Sigma_{o,v_2}| = 1$. Furthermore, by (b), $|\Sigma_{o,v_6}| = 0$. □

5.3.1 Strong Hardness Gadgets

Definition 5.23 (strong hardness gadget)**.** A graph J is called a *strong hardness gadget* if every K_4-minor-free graph that contains J as a subgraph has a hardness gadget.

Lemma 5.24. *The following graph J is a strong hardness gadget:*



Proof. Let H be a K_4-minor-free supergraph of J. We construct a hardness gadget of H:

- $S = \{s\}$ and $I = \{i\}$.

- J_1 is the graph where y is adjacent to a u-pin and an i-pin. Note that $\Omega_y = \{x, s\}$ as H is K_4-minor free.

- J_2 is the graph where z is adjacent to a v-pin and an s-pin. Note that $\Omega_z = \{x, i\}$ as H is K_4-minor free.

- J_3 is just the edge $\{y, z\}$.

We have $|\Sigma_{s,i}| = |\Sigma_{s,x}| = |\Sigma_{x,i}| = 1$ and $|\Sigma_{x,x}| = 0$ — recall that we do not allow self-loops. □

For the proof of the following lemma recall the definition of walk-neighbour-sets from Definition 5.2.

Lemma 5.25. *Let H be a K_4-minor-free graph containing two adjacent vertices a and b such that $|\Gamma_H(a) \cap \Gamma_H(b)|$ is odd and at least 3. Then H has a hardness gadget.*

Proof. Let c be a common neighbour of a and b and consider the triangle $C = (a, b, c, a)$: If a and c have a common neighbour apart from b, or if b and c have a common neighbour apart from a then Lemma 5.24 applies, as a and b have a common neighbour apart from c by assumption. Otherwise, we have that $|N_{C,H}(a)| = 1$, $|N_{C,H}(b)| = 1$, and $|N_{C,H}(c)| = j \geq 3$, where j is odd. For any $w \in N_{C,H}(c)$ we can assume that

$$\Gamma_H(w) \cap \Gamma_H(a) = \{b\} \text{ and } \Gamma_H(w) \cap \Gamma_H(b) = \{a\}, \tag{5.3}$$

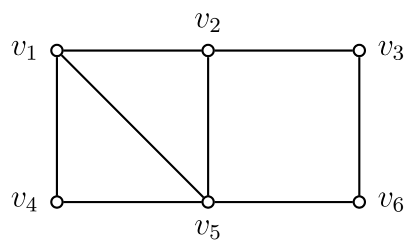
as otherwise we obtain a hardness gadget from Lemma 5.24 (choose w instead of c). Next we can apply Lemma 5.13 to obtain a hardness gadget as follows.

Let $q = 3$ and $\mathcal{C}_0 = N_{C,H}(a) = \{a\}$, $\mathcal{C}_1 = N_{C,H}(b) = \{b\}$, $\mathcal{C}_2 = N_{C,H}(c)$. For each $i \in \{0, 1, 2\}$, let (J_i, z_i) be the partially H-labelled graph that only contains the single vertex z_i and has an empty pinning function. It follows that $\Omega_i = V(H)$. We choose $k = 0$ and check that the requirements of Lemma 5.13 are met.

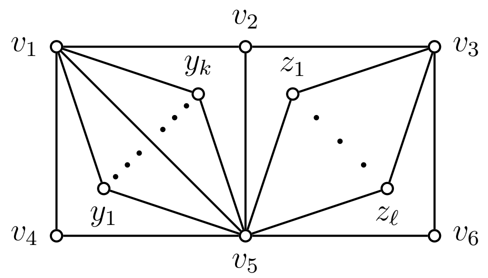
- (L5.12.1) holds since, for each $i \in \{0, 1, 2\}$, $\Omega_i = V(H)$ and $\mathcal{C}_i$ has odd cardinality (either 1 or j).
- (L5.12.2) and (L5.12.3) hold by (5.3) and the fact that $\Gamma_H(a) \cap \Gamma_H(b) = N_{C,H}(c) = \mathcal{C}_2$.
- Suppose for contradiction that there exists a walk $D = (d_a, d_b, d_c, d_a)$ with $d_a \in \Gamma_H(a) \setminus \{b, c\}$, $d_b \in \Gamma_H(b) \setminus \{a, c\}$ and $d_c \in \Gamma_H(c) \setminus \{a, b\}$. Consequently, as we do not allow self-loops in H, $d_a \neq a$, $d_b \neq b$ and $d_c \neq c$. Then the vertices d_a, a, b, c induce a K_4-minor (where the path from d_a to b goes via d_b, and the path from d_a to c goes via d_c). Hence (L5.12.4) holds.
- Since $\mathcal{C}_0 = \mathcal{C}_{3 \bmod q} = \{a\}$, (L5.13.1) holds by the fact that we do not allow self-loops in H.
- (L5.13.2) holds since $(\mathcal{C}_0 \cup \mathcal{C}_2) \cap \Omega_1 = \mathcal{C}_0 \cup \mathcal{C}_2$, which has cardinality $j + 1$ (as we do not allow self-loops in H and therefore $\mathcal{C}_0 = \{a\}$ and $\mathcal{C}_2 = N_{C,H}(c) = \Gamma_H(a) \cap \Gamma_H(b)$ are disjoint), and $j+1$ is even. Analogously, $(\mathcal{C}_1 \cup \mathcal{C}_2) \cap \Omega_1 = \mathcal{C}_1 \cup \mathcal{C}_2$ has even cardinality.

□

Lemma 5.26. *The following graph J is a strong hardness gadget:*

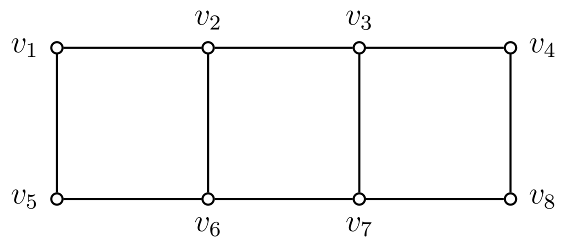


Proof. Let H be a K_4-minor-free supergraph of J. In particular, the graph F is a subgraph of J and thus of H. Note that, due to the edge $\{v_1, v_5\}$, the vertices v_2 and v_4 have no common neighbours apart from v_1 and v_5 in H, as we would obtain a K_4-minor otherwise. In other words, $\Gamma_{H \setminus F}(2, 4) = 0$. By Lemma 5.20 we are done, unless F has type V in H. In particular, as $\Gamma_{H\setminus F}(2,4) = \emptyset$, only the following case remains:



In particular, $\Gamma_{H\setminus F}(1,5) = \{y_1, \ldots, y_k\}$ and $\Gamma_{H\setminus F}(3,5) = \{z_1, \ldots, z_\ell\}$ and $k, \ell > 0$. Now, if k is even, then Lemma 5.22 yields a hardness gadget of H. Finally, if k is odd, then Lemma 5.25 yields a hardness gadget of H — note that Lemma 5.25 is applicable as v_1 and v_5 have precisely $k + 2$ common neighbours, which is an odd number greater or equal than 3 since k is odd and positive. □

Lemma 5.27. *The following graph J is a strong hardness gadget:*

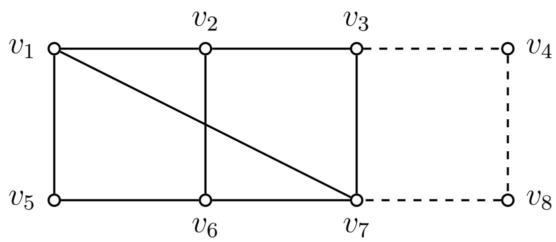


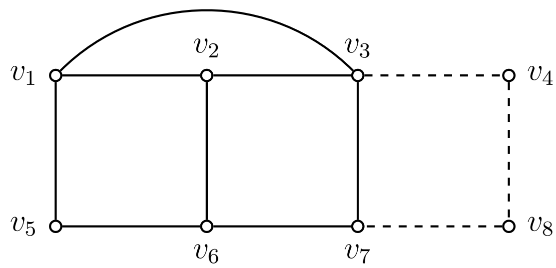
Proof. Let H be a K_4-minor-free supergraph of J.

Claim A *If J is not an induced subgraph of H then H has a K_4-minor or a hardness gadget.*

Proof: If J is not an induced subgraph of H then there is an edge $e = \{v_i, v_j\} \in E(H) \setminus E(J)$ for some $i \neq j \in [8]$. If e is a diagonal of one of the three squares, such as $\{v_2, v_7\}$, then H has a hardness gadget by Lemma 5.26.

If e is not a diagonal of a square, then we obtain a K_4-minor; each case is similar to one of the following two:





■

Thus assume for the remainder of the proof that J is an induced subgraph of H. Note that J has two subgraphs isomorphic to F. We are done unless both have type V in H by Lemma 5.20. If both have type V, but Lemma 5.22 is applicable, we are done as well. There is thus only one case (up to symmetry) remaining:

(a) $\Gamma_H(v_1) \cap \Gamma_H(v_6) = \{v_2, v_5\}$,

(b) $\Gamma_H(v_3) \cap \Gamma_H(v_6) = \{v_2, v_7\}$,

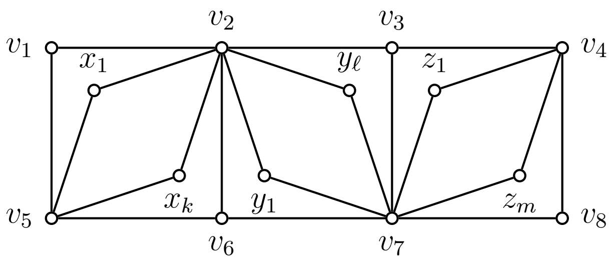
(c) $\Gamma_H(v_3) \cap \Gamma_H(v_8) = \{v_4, v_7\}$,

(d) $|\Gamma_H(v_2) \cap \Gamma_H(v_5)|$ is odd,

(e) $|\Gamma_H(v_2) \cap \Gamma_H(v_7)|$ is odd, and

(f) $|\Gamma_H(v_4) \cap \Gamma_H(v_7)|$ is odd.

We provide an illustration for convenience:

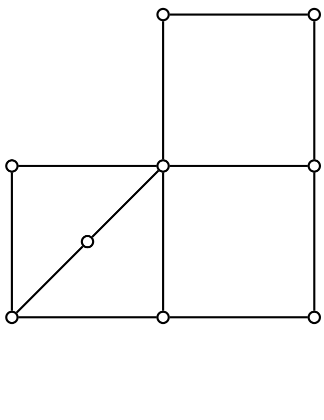


Note that k, ℓ and m are odd. We construct a hardness gadget:

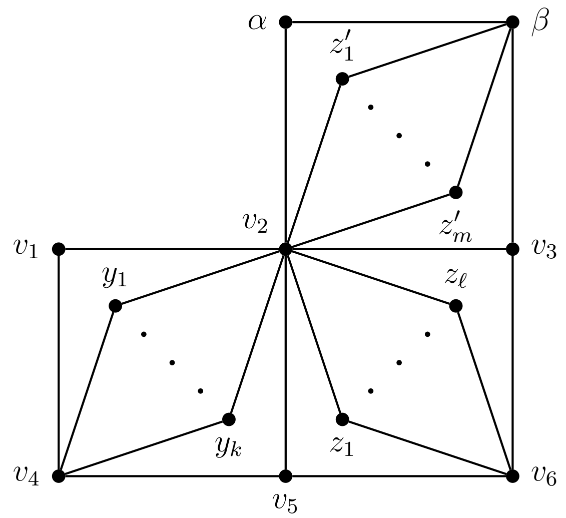
- $S = \{v_2\}$ and $I = \{v_7\}$.
- J_1 is the graph where y is adjacent to a v_1-pin and a v_6-pin. Note that $\Omega_y = \{v_2, v_5\}$ by (a).
- J_2 is the graph where z is adjacent to a v_3-pin and a v_8-pin. Note that $\Omega_z = \{v_7, v_4\}$ by (c).
- J_3 is a path of length 2 from y to z.

By (d), (e) and (f) we have that $|\Sigma_{v_5,v_2}|$, $|\Sigma_{v_2,v_7}|$ and $|\Sigma_{v_7,v_4}|$ are odd. Furthermore, we observe that $|\Sigma_{v_5,v_4}| = 0$ as any path of length 2 from v_5 to v_4 would create a K_4-minor. $\square$

Lemma 5.28. *The following graph J is a strong hardness gadgets:*



Proof. Let H be a K_4-minor-free supergraph of J. Note that J has two subgraphs isomorphic to F. By Lemma 5.20, we obtain a hardness gadget of H, unless both of the subgraphs isomorphic to F have type V. If this is the case, however, we obtain the following subgraph $\hat{J}$ of H:



In $\hat{J}$, $k, \ell, m > 0$ and all common neighbours in H between the pairs (v_2, v_4), (v_2, v_6) and (v_2, β) are depicted. By definition of type V, we also obtain that each of the pairs (v_1, v_5), (v_5, v_3) and (v_3, α) has only the two common neighbours in H depicted. Note further, that H has a hardness gadget if at least one of k, ℓ or m is even by Lemma 5.22. Thus assume for the remainder of the proof that all three are odd. We will rely on the following claim, that we can assume $\hat{J}$ to be an *induced* subgraph of H:

Claim A: *If $\hat{J}$ is not an induced subgraph of H then H has a K_4-minor or a hardness gadget.*

Proof: Let $e \in E(H) \setminus E(\hat{J})$ be an edge of H that connects two vertices of $\hat{J}$. We first assume that e connects two vertices in

$$\{v_1, v_2, v_3, v_4, v_5, v_6, y_1, \dots, y_k, z_1, \dots, z_\ell\}\,.$$

We show by case distinction that e either yields a hardness gadget, or a K_4-minor:

(I) $x \in e$ for $x \in \{v_1, y_1, \dots, y_k\}$. Let x' be the other endpoint of e and note that $x' \notin \{v_4, v_2, x\}$ as we do not allow self-loops and multiple edges.

(i) If $x' \in \{v_1, y_1, \dots, y_k, v_5\}$ then we obtain a K_4-minor induced by x, x', v_2, v_4 — note that, as $k > 0$, there exists a 2-path from v_2 to v_4 whose internal vertex is neither x nor x'.

(ii) If $x' \in \{v_3, z_1, \dots, z_\ell\}$ then we obtain a K_4-minor induced by x, v_2, x', v_5 — note that there is a 2-path from v_5 to x via v_4, and a 2-path from v_5 to x' via v_6.

(iii) If $x' = v_6$, then we obtain a K_4-minor induced by x, v_2, v_6, v_5 — note that there is a 2-path from v_5 to x via v_4, and a 2-path from v_6 to v_2 via v_3.

(II) $x \in e$ for $x \in \{v_3, z_1, \ldots, z_\ell\}$. Symmetric to the previous case (I).

(III) $v_4 \in e$. Let x' be the other endpoint of e and note that $x' \notin \{v_4, v_1, y_1, \ldots, y_k, v_5\}$ as we do not allow self-loops and multiple edges.

(i) If $x' \in \{v_3, z_1, \ldots, z_\ell\}$ then the case is symmetric to case (I)(iii).

(ii) If $x' = v_6$ then we obtain a K_4-minor induced by v_4, v_2, v_6, v_5 — note that there is a 2-path from v_4 to v_2 via v_1, and a 2-path from v_2 to v_6 via v_3.

(iii) If $x' = v_2$, then H has a hardness gadget by Lemma 5.26.

(IV) $v_6 \in e$. Symmetric to the previous case (III).

(V) $v_2 \in e$. Let x' be the other endpoint of e. It follows that $x' \notin \{v_2, v_1, y_1, \ldots, y_k, v_5, z_1, \ldots, z_\ell, v_3\}$ as we do not allow self-loops and multiple edges. The only remaining candidates for x' are thus v_4 and v_6. However, both of the latter candidates yield a hardness gadget by Lemma 5.26.

(VI) $v_5 \in e$. Let x' be the other endpoint of e and note that $x' \notin \{v_5, v_4, v_2, v_6\}$ as we do not allow self-loops and multiple edges. Similarly as in the previous case (V), all other candidates for x' yield a hardness gadget by Lemma 5.26.

This concludes the case distinction. Observe now, that a symmetric case analysis shows H has a hardness gadget or a K_4-minor if e connects two vertices in

$$\{v_5, v_2, \alpha, v_6, v_3, \beta, z_1, \ldots, z_\ell, z'_1, \ldots, z'_m\}.$$

The remaining possibility for e is to have one endpoint in $\{v_4, v_1, y_1, \ldots, y_k\}$ and the other endpoint in $\{\alpha, \beta, z'_1, \ldots, z'_m\}$. However, in this case, we find a path from v_5 to v_3 whose vertices are disjoint from $\{v_2, z_1, \ldots, z_\ell, v_6\}$. Consequently, we obtain a K_4-minor induced by v_2, v_3, v_5, v_6. ■

We thus assume that $\hat{J}$ is an induced subgraph of H in what follows. Next, we perform a case distinction on the parity of the degree of v_2; in both cases, we construct a hardness gadget.

(I) $\deg_H(v_2)$ is even. We construct a hardness gadget:

- $I = \{v_4\}$ and $S = \{v_6\}$.
- J_1 is the graph where y is adjacent to a v_1-pin and a v_5-pin so $\Omega_y = \{v_2, v_4\}$.
- J_2 is the graph where z is adjacent to a v_5-pin and a v_3-pin so $\Omega_z = \{v_2, v_6\}$.
- J_3 is a 2-path between y and z.

As the degree of v_2 is even, $|\Sigma_{v_2,v_2}|$ is even. As k and ℓ are odd, $|\Sigma_{v_2,v_6}|$ and $|\Sigma_{v_2,v_4}|$ are odd. Finally, we claim that $|\Sigma_{v_6,v_4}|$ is odd: Otherwise there must be an additional 2-path from v_6 to v_4. As $\hat{J}$ is an induced subgraph of H, the internal vertex of this path, let us call it x, cannot be contained in $V(\hat{J})$; otherwise, H would contain an edge between x and a vertex v of $\hat{J}$ while x and v are not adjacent in $\hat{J}$.

This, however, yields a K_4-minor induced by the vertices v_4, v_2, v_6 and v_5 — note that v_4 and v_2 are connected by the 2-path via v_1, v_2 and v_6 are connected by the 2-path via v_3, and v_4 and v_6 are connected by the 2-path via x.

(II) $\deg_H(v_2)$ is odd. We construct a hardness gadget:

- $I = S = \{v_2\}$.
- J_1 is the graph where y is adjacent to a v_1-pin and a v_5-pin so $\Omega_y = \{v_2, v_4\}$.
- J_2 is the graph where z is adjacent to an α-pin and a v_3-pin so $\Omega_z = \{v_2, \beta\}$.
- J_3 is a 2-path between y and z.

As the degree of v_2 is odd, $|\Sigma_{v_2,v_2}|$ is odd. As k and ℓ are odd, we have that $|\Sigma_{v_4,v_2}|$ and $|\Sigma_{v_2,\beta}|$ are odd. Finally, we claim that $|\Sigma_{v_4,\beta}|$ is even: Assuming otherwise, there must be at least one 2-path in H from v_4 to β; we show that there is none.

As $\hat{J}$ is an induced subgraph of H, the internal vertex of this path, let us call it x, cannot be contained in $V(\hat{J})$; otherwise, H would contain an edge between x and a vertex v of $\hat{J}$ while x and v are not adjacent in $\hat{J}$.

This, however, yields a K_4-minor induced by the vertices v_2, β, v_5 and v_4 — note that v_4 and v_2 are connected by the 2-path via v_1, v_2 and β are connected by the 2-path via α, v_4 and β are connected by the 2-path via x, and v_5 and β are connected by the 3-path via v_6 and v_3.

□

5.3.2 Chordal Bipartite Component Lemma

Definition 5.29 ((1,2)-supergraph)**.** Let J be a connected graph. We say that a supergraph H of J is a *(1,2)-supergraph* of J if every edge of H connecting vertices of J is also an edge of J and every length-2 path of H connecting vertices of J is also a path of J.

For what follows, recall that a chordal bipartite graph is a graph in which every induced cycle is a square. The following notion captures the K_4-minor-free (biconnected) graphs that are obtained by gluing squares together without inducing ⊕P-hardness.

Definition 5.30 (impasse, pair of connectors)**.** A K_4-minor-free biconnected graph B is called an *impasse* if there are odd positive integers k and ℓ such that B is a (1,2)-supergraph of the graph $S_{k,\ell}$. Also, with the vertex labels from Definition 5.21, all of the vertices in $\{v_1, y_1, \ldots, y_k, v_3, z_1, \ldots, z_\ell\}$ are required to have degree 2 in B. The pair (v_1, v_3) is called a *pair of connectors* of the impasse B. (Note that a pair of connectors of B is not unique as, for instance, (v_1, z_1) is also a pair of connectors.)

The graph in Figure 5.5 is an example of an impasse.

Definition 5.31 (diamond)**.** A biconnected graph B is a *diamond* if, for an integer $k \geq 2$, $V(B) = \{s, t, x_1, \ldots, x_k\}$ and $E(B) = \cup_{i\in[k]}\{\{s, x_i\}, \{x_i, t\}\}$.

Note that a square is a diamond with $k = 2$. The following lemma classifies biconnected chordal bipartite graphs:

Lemma 5.32 (Chordal Bipartite Component Lemma)**.** *Let H be a K_4-minor-free graph and let B be a biconnected component of H. If B is chordal bipartite and not just a single edge, then at least one of the following is true:*

(a) B is a diamond.

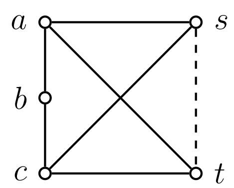
(b) H has a hardness gadget.

(c) B is an impasse.

Proof. As B is biconnected, chordal bipartite and not a single edge, there exists an induced square $C = (a, b, c, d, a)$ in B. Let us write $\Gamma_{H\setminus C}(a, c)$ for the set $\Gamma_H(a) \cap \Gamma_H(c) \setminus \{b, d\}$ and $\Gamma_{H\setminus C}(b, d)$ for the set $\Gamma_H(b) \cap \Gamma_H(d) \setminus \{a, c\}$. Since B is a biconnected component of H, and $a, b, c, d \in B$, we actually have that $\Gamma_{H\setminus C}(a, c) = \Gamma_B(a) \cap \Gamma_B(c) \setminus \{b, d\}$ and $\Gamma_{H\setminus C}(b, d) = \Gamma_B(b) \cap \Gamma_B(d) \setminus \{a, c\}$. As H is K_4-minor free, we observe that at least one of $\Gamma_{H\setminus C}(a, c)$ and $\Gamma_{H\setminus C}(b, d)$ is empty. Assume w.l.o.g., that $\Gamma_{H\setminus C}(b, d)$ is empty. Let B' be the graph consisting of C together with the edges from a and c to $\Gamma_{H\setminus C}(a, c)$. If $B = B'$ then B is a diamond. Otherwise, as B is biconnected, there is a shortest path P in B connecting two vertices of $C \cup \Gamma_{H\setminus C}(a, c)$ whose internal vertices are not in B'. This path P has an internal vertex since B has no triangle.

Claim A: *P has length 3, one endpoint of P is contained in $\Gamma_B(a) \cap \Gamma_B(c)$ and the other endpoint is contained in $\{a, c\}$.*

Proof: Assume first, for contradiction, that both endpoints of P, let us call them s and t, are in $\Gamma_B(a) \cap \Gamma_B(c)$. The only possible length for P under this assumption is 2, as, otherwise, we obtain an induced cycle (a, s, P, t, a) of length $\neq 4$. As P must have length 2, the endpoints of P cannot be b and d, as $\Gamma_{H\setminus C}(b, d)$ is empty. Thus we can assume w.l.o.g. that $s \neq b$ and $t \neq b$, which yields the following K_4-minor; P is depicted dashed:



This yields the desired contradiction.

Next, if P starts in a and ends in c, then we obtain an induced cycle that is not a square, unless P as length 2. However, in the latter case, the internal vertex of P is contained in $\Gamma_{H\setminus C}(a,c)$, contradicting the fact that P is not fully contained in $C \cup \Gamma_{H\setminus C}(a,c)$. This shows that one endpoint of P is in $\Gamma_B(a) \cap \Gamma_B(c)$ and the other endpoint is in $\{a,c\}$.

Recall that the length of P is greater than 1 (since it has internal vertices). If P has length 2, then we obtain a triangle, contradicting the fact that B is chordal bipartite. Finally, if P has length at least 4, we obtain an induced cycle of length at least 5, also contradicting chordal-bipartiteness. Consequently, P must have length 3. ■

Claim A yields that B contains a subgraph isomorphic to the graph F — recall from Definition 5.16 that F is just the graph containing two squares that share one edge. Now assume that (b) is not true, i.e., that H does not have a hardness gadget. Using the fact that H is K_4-minor free, and invoking Lemma 5.20 and Lemma 5.22, we obtain that there are odd positive integers k and ℓ such that B contains a subgraph isomorphic to the graph $S_{k,\ell}$ from Definition 5.21.
We use the vertex labels from Definition 5.21, i.e., $\Gamma_B(v_4) \cap \Gamma_B(v_2) = \{v_1, y_1, \ldots, y_k, v_5\}$ and $\Gamma_B(v_6) \cap \Gamma_B(v_2) = \{v_5, z_1, \ldots, z_\ell, v_3\}$.

Claim B *$S_{k,\ell}$ is an induced subgraph of B.*

Proof: Assume that $S_{k,\ell}$ is not an induced subgraph. Then B (equivalently, H) contains an edge $e \notin E(S_{k,\ell})$ between two vertices of $S_{k,\ell}$. We need to distinguish a variety of (simple) cases:

- $v_4 \in e$: The other endpoint of e cannot be one of $v_4, v_5, y_1, \ldots, y_k, v_1$ as we do not allow self-loops and multi-edges. Further, it cannot be v_6 or v_2, as this would create a triangle, contradicting the fact that B is chordal-bipartite. Finally, if the other endpoint of e is $x \in \{v_3, z_1, \ldots, z_\ell\}$, then we obtain a K_4-minor induced by the vertices v_4, v_5, v_2 and x — note that there is a 2-path from x to v_5 via v_6, and a 2-path from v_2 to v_4 via v_1.

- $v_6 \in e$: Symmetric to the previous case.

- $x \in e$ for some $x \in \{v_1, y_1, \ldots, y_k\}$: The other endpoint of e cannot be one of

$$v_1, y_1, \ldots, y_k, v_4, v_2, v_5, v_3, z_1, \ldots, z_\ell\,,$$

  as each of those cases would yield a self-loop, a multi-edge, or a triangle (in B). The remaining candidate for the other endpoint is v_6, which, however, yields a K_4-minor induced by the vertices v_5, v_6, v_2 and x — note that there is a 2-path from x to v_5 via v_4, and a 2-path from v_2 to v_6 via v_3.

- $x \in e$ for some $x \in \{v_3, z_1, \ldots, z_\ell\}$: Symmetric to the previous case.

- $v_5 \in e$: Any (additional) edge from v_5 to a vertex of $S_{k,\ell}$ would create either a multi-edge, a self-loop, or a triangle.

- $v_2 \in e$: The other endpoint of e cannot be v_2 as we this would create a self-loop. Consequently, one of the previous cases must be true for the other endpoint of e.

■

Recall that we want to show that (a) B is a diamond, (b) H has a hardness gadget, or (c) B is an impasse. For what follows, we distinguish two cases:

(I) All vertices $v_1, y_1, \ldots, y_k, z_1, \ldots, z_\ell, v_3$ have degree 2 in B. In this case we will show that B is a (1,2)-supergraph of $S_{k,\ell}$. This implies (see Definition 5.30) that B is an impasse, so we are finished. To see that B is a (1,2)-supergraph of $S_{k,\ell}$, recall (from Claim B) that $S_{k,\ell}$ is an induced subgraph of B. All neighbours of $v_1, y_1, \ldots, y_k, z_1, \ldots, z_\ell, v_3$ in B are included in $S_{k,\ell}$. Thus, it suffices to show that B has no 2-path connecting vertices in $\{v_4, v_5, v_6, v_2\}$ whose internal vertex x, is outside of $S_{k,\ell}$. We noted above that $\Gamma_B(v_4) \cap \Gamma_B(v_2) \subseteq V(S_{k,\ell})$ and $\Gamma_B(v_6) \cap \Gamma_B(v_2) \subseteq V(S_{k,\ell})$. There is no 2-path in B from v_2 to v_5 because that would yield a triangle in B. Similarly, 2-paths from v_5 to v_4 or v_6 would yield triangles in B, so the only possibility is a 2-path from v_4 to v_6 but this would yield the K_4-minor $\{v_4, v_5, v_6, v_2\}$ in B, contradicting the fact that H (hence B) has no K_4-minor.

(II) Otherwise, assume w.l.o.g. that v_1 has degree at least 3 in B. As B is biconnected, there exists a shortest path P in the remainder of B connecting v_1 with another vertex w of $S_{k,\ell}$. We claim that the only candidates for w are v_4 and v_2, which we will prove by case distinction:

  - $w \in \{y_1, \ldots, y_k, v_5\}$. Then we obtain a K_4-minor: (v_4, w, v_2, v_1, v_4) is a square, P connects v_1 and w via vertices not contained in $S_{k,\ell}$, and v_4 and v_2 are connected by a 2-path via a vertex $x \in \{y_1, \ldots, y_k, v_5\} \setminus w$ — note that x exists as $k \geq 1$.
  - $w = v_6$. Then we obtain a K_4-minor induced by the vertices v_5, v_6, v_1 and v_2 — note that v_1 is connected to v_5 by the 2-path via v_4, and that v_2 is connected to v_6 by the 2-path via v_3.
  - $w \in \{z_1, \ldots, z_\ell, v_3\}$. Then we obtain a K_4-minor induced by the vertices v_5, v_1, v_2 and w — note that v_1 is connected to v_5 by the 2-path via v_4, and that v_5 is connected to w by the 2-path via v_6.

  Consequently, w must either be v_4 or v_2 as all other possibilities create a K_4-minor. As B is chordal bipartite, P must have length three. However, if P connects v_1 and v_2, we obtain a strong hardness gadget by Lemma 5.28, and if P connects v_1 and v_4, we obtain a strong hardness gadget by Lemma 5.27. In both cases, H therefore has a hardness gadget.

□

The following lemma shows that impasses already yield hardness if the vertex v_2 has even degree:

Lemma 5.33. *Let H be a graph containing an impasse B as biconnected component, that is, there are odd integers k and ℓ such that B is a (1,2)-supergraph of the graph $S_{k,\ell}$ such that, using the vertex labels from Figure 5.6, all vertices $v_1, y_1, \dots, y_k, v_3, z_1, \dots, z_\ell$ have degree 2 in B. If $\deg_H(v_2)$ is even, then H has a hardness gadget.*

Proof. We construct a hardness gadget:

- $I = \{v_4\}$ and $S = \{v_6\}$.
- J_1 is the graph where y is adjacent to a v_1-pin and a v_5-pin so $\Omega_y = \{v_2, v_4\}$ as H has the impasse B as a biconnected component.
- J_2 is the graph where z is adjacent to a v_5-pin and a v_3-pin so $\Omega_z = \{v_2, v_6\}$ as H has the impasse B as a biconnected component.
- J_3 is a 2-path between y and z.

As the degree of v_2 is even, $|\Sigma_{v_2,v_2}|$ is even. As k and ℓ are odd, we have that $|\Sigma_{v_2,v_4}|$ and $|\Sigma_{v_2,v_6}|$ are odd. Finally, we also have $|\Sigma_{v_4,v_6}| = 1$ as an additional 2-path from v_4 to v_6 would contradict the fact that the biconnected component B of H is an impasse. □

5.4 Sequences of Chordal Bipartite Components

Definition 5.34 (good start, good stop)**.** Let H be a graph and let B be a subgraph of H. Let y be a vertex in B and let $L_B \subseteq \Gamma_H(y) \cap V(B)$.

- We say that (L_B, y) is a *good start* in B if there is a gadget (J, z) such that $\{v \in V(H) \mid |\hom((J, z) \to (H, v))| \text{ is odd}\} = L_B \cup R_B$, where $|L_B|$ is odd and $R_B = \Gamma_H(y) \setminus V(B)$.
- We say that (L_B, y) is a *good stop* in B if it is a good start in B and $|R_B|$ is odd.

For non-negative integers k and ℓ, we define some (classes of) graphs with a pair of distinguished vertices a and b each, see Figure 5.7 (The graph $S_{k,\ell}$ was already defined in Definition 5.30, however, for the scope of this section it will be more convenient to work with the vertex labels as given in Figure 5.7.).

5.4.1 Good Starts

Lemma 5.35. *Let B be a biconnected component of a graph H, where B is an edge between vertices a and b. Then $(\{a\}, b)$ is a good start in B.*

Proof. Clearly, $\{a\}$ has odd cardinality, and is contained in $\Gamma_H(b) \cap V(B)$. Let (J_B, z_B) be the gadget where z_B is adjacent to a b-pin and let $R_B = \Gamma_H(b) \setminus \{a\}$. Then $\{v \in V(H) \mid |\hom((J_B, z_B) \to (H, v))| \text{ is odd}\} = \Gamma_H(b) = \{a\} \cup R_B$, as desired. □

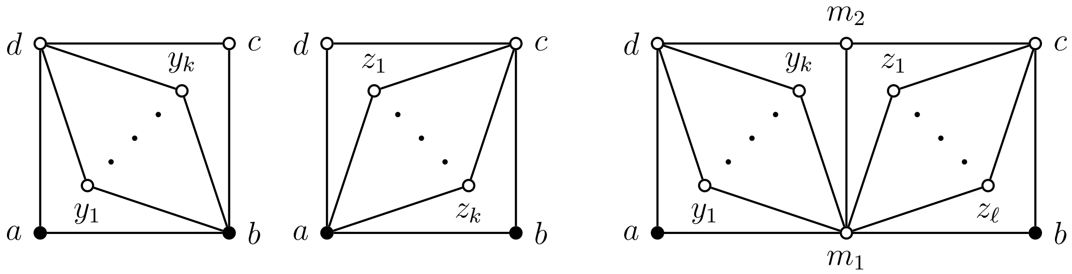


Figure 5.7: The graphs BD_k (for "backward diamond"), FD_k (for "forward diamond") and $S_{k,\ell}$ (from left to right).

Lemma 5.36. *Let B be a biconnected component of a graph H such that, for an* even *non-negative integer k, B is a graph of the form FD_k and the vertices a and b are as given in Figure 5.7. Let A be a subgraph of H such that $V(A) \cap V(B) = \{a\}$ Suppose that $L_A \subseteq \Gamma_H(a) \cap V(A)$. If (L_A, a) is a good start in A but not a good stop in A then $(\{a\}, b)$ is a good start in B.*

Proof. By the definition of a good start, $|L_A|$ is odd and there is a gadget (J_A, z_A) such that $\{v \in V(H) \mid |\hom((J_A, z_A) \to (H, v))| \text{ is odd}\} = L_A \cup R_A$ where $R_A = \Gamma_H(a) \setminus V(A)$. Since (L_A, a) is not a good stop in A, $|R_A|$ is even.

Let $L_B = \{a\}$. We now prove the lemma by showing that (L_B, b) is a good start in B. Clearly, $|L_B|$ is odd.

Let (J_B, z_B) be the gadget where z_B is adjacent to the vertex z_A of the gadget J_A and it is also adjacent to a b-pin. In order to prove that (L_B, b) is a good start we check that $\{v \in V(H) \mid |\hom((J_B, z_B) \to (H, v))| \text{ is odd}\} = L_B \cup R_B$, where $L_B = \{a\}$ and $R_B = \Gamma_H(b) \setminus V(B)$.

Since z_B is adjacent to a b-pin we need only consider each $v \in \Gamma_H(b)$ and homomorphisms with $z_B \mapsto v$. Then z_A is also adjacent to z_B and can be mapped to every vertex in the set $\Gamma_H(v) \cap (L_A \cup R_A)$. We determine the cardinality of this set depending on v:

- If $v = a$ then $\Gamma_H(v) \cap (L_A \cup R_A) = L_A \cup R_A$ and $|L_A \cup R_A|$ is odd, as required.
- If $v = c$ (for c as given in Figure 5.7) then $v = c$ does not have any neighbours in L_A since every path from L_A to B goes through a because B is a biconnected component of H. Hence, $\Gamma_H(v) \cap (L_A \cup R_A) = \Gamma_H(v) \cap R_A$ and $\Gamma_H(v) \cap R_A = \Gamma_H(c) \cap (\Gamma_H(a) \setminus V(A))$ by definition of R_A. Finally, since B is a biconnected component, the vertices of B have no common neighbours outside of B. Thus $\Gamma_H(c) \cap (\Gamma_H(a) \setminus V(A)) = \{d, b, z_1, \ldots, z_k\}$, which has even cardinality, as required (since k is even).
- If $v \in \Gamma_H(b) \setminus V(B)$ then since B is a biconnected component we have $\Gamma_H(v) \cap \Gamma_H(a) = \{b\}$. Consequently, $\Gamma_H(v) \cap (L_A \cup R_A) = \{b\}$, which is odd, as required.

□

Lemma 5.37. *Let B be a biconnected component of a graph H such that, for a non-negative integer k, B is a graph of the form BD_k and the vertices a and b are as* *given in Figure 5.7. Let A be a subgraph of H such that $V(A) \cap V(B) = \{a\}$. Suppose that $L_A \subseteq \Gamma_H(a) \cap V(A)$. If (L_A, a) is a good start in A but not a good stop in A then $(\{a\}, b)$ is a good start in B.*

Proof. The proof is analogous to that of Lemma 5.36. We define $L_B = \{a\}$ and use the same gadget and again have to consider each $v \in \Gamma_H(b)$ and homomorphisms with $z_B \mapsto v$ and consequently determine the cardinality of the set $\Gamma_H(v) \cap (L_A \cup R_A)$ depending on v:

- If $v = a$ then $\Gamma_H(v) \cap (L_A \cup R_A) = L_A \cup R_A$ and $|L_A \cup R_A|$ is odd, as required.
- If $v \in \{c, y_1, \dots y_k\}$ (as given in Figure 5.7) then v does not have any neighbours in L_A since every path from L_A to B goes through a. Hence, $\Gamma_H(v) \cap (L_A \cup R_A) = \Gamma_H(v) \cap R_A$ and $\Gamma_H(v) \cap R_A = \Gamma_H(v) \cap (\Gamma_H(a) \setminus V(A))$ by definition of R_A. Finally, since B is a biconnected component, the vertices of B have no common neighbours outside of B, $\Gamma_H(v) \cap (\Gamma_H(a) \setminus V(A)) = \{b, d\}$, which has even cardinality, as required.
- If $v \in \Gamma_H(b) \setminus V(B)$ then since B is a biconnected component we have $\Gamma_H(v) \cap \Gamma_H(a) = \{b\}$. Consequently, $\Gamma_H(v) \cap (L_A \cup R_A) = \{b\}$, which is odd, as required.

□

Lemma 5.38. *Let B be a biconnected component of a graph H, where B is an impasse (Definition 5.30). Let (a, b) be a pair of connectors of B and let m_1 be the unique common neighbour of a and b in H (see Figure 5.7). Suppose further that $\deg_H(m_1)$ is odd. Let A be a subgraph of H such that $V(A) \cap V(B) = \{a\}$. Suppose that $L_A \subseteq \Gamma_H(a) \cap V(A)$. If (L_A, a) is a good start in A but not a good stop in A then $(\{m_1\}, b)$ is a good start in B.*

Proof. By the definition of a good start, $|L_A|$ is odd and there is a gadget (J_A, z_A) such that $\{v \in V(H) \mid |\hom((J_A, z_A) \to (H, v))| \text{ is odd}\} = L_A \cup R_A$, where $R_A = \Gamma_H(a) \setminus V(A)$. Since (L_A, a) is not a good stop, $|R_A|$ is even.

Let $L_B = \{m_1\}$. We now prove the lemma by showing that (L_B, b) is a good start in B. Clearly, $|L_B|$ is odd.

Let (J_B, z_B) be the gadget that consists of the gadget J_A joined with a path of length 2 from the vertex z_A to the vertex z_B, and a b-pin that is adjacent to z_B. In order to prove that (L_B, b) is a good start we check that $\{v \in V(H) \mid |\hom((J_B, z_B) \to (H, v))| \text{ is odd}\} = L_B \cup R_B$, where $L_B = \{m_1\}$ and $R_B = \Gamma_H(b) \setminus V(B)$.

Since z_B is adjacent to a b-pin we need only consider $v \in \Gamma_H(b)$ and homomorphisms with $z_B \mapsto v$. Then there is a path of length 2 from z_A to z_B and therefore, for $v \in \Gamma_H(b)$,

$$|\hom((J_B, z_B) \to (H, v))| = |\{u \in L_A \cup R_A \mid \ |\Gamma_H(u) \cap \Gamma_H(v)| \text{ is odd.}\}|.$$

We determine $|\{u \in L_A \cup R_A \mid \ |\Gamma_H(u) \cap \Gamma_H(v)| \text{ is odd.}\}|$ depending on v and using the vertex labels from Figure 5.7. Note that m_1 and c are the only neighbours of b in B since the degree of b is 2 in B (by the definition of an impasse).

- Consider $v = m_1$.
  - If $u \in \Gamma_H(a) \setminus \{d, m_1\}$ then $u \notin V(B)$ since $\deg_B(a) = 2$. As B is a biconnected component, it follows that a is the only common neighbour of $v = m_1$ and u.
  - The vertices $v = m_1$ and $u = d$ have an odd number of common neighbours in $S_{k,\ell}$ since k is odd. They have no further common neighbours in B, since B is an impasse, and no further common neighbours in H since B is a biconnected component of H.
  - Finally, $v = m_1$ and $u = m_1$ have an odd number of common neighbours since $\deg_H(m_1)$ is odd by assumption of the lemma.

  Therefore, $\{u \in L_A \cup R_A \mid \ |\Gamma_H(u) \cap \Gamma_H(v)| \text{ is odd}\} = L_A \cup R_A$ and $|L_A \cup R_A|$ is odd, as required.
- Consider $v = c$.
  - If $u \in \Gamma_H(a) \setminus \{d, m_1\}$, then $u \notin V(B)$ and, as B is a biconnected component, $v = c$ and u have no common neighbours.
  - The vertices $v = c$ and $u = d$ have one common neighbour in $S_{k,\ell}$ (the vertex m_2) and no further common neighbours in H (by the same argument as we used for $v = m_1$), so $v = c$ and $u = d$ have an odd number of common neighbours in H.
  - Finally, $v = c$ and $u = m_1$ have an odd number of common neighbours (since ℓ is odd).

  Therefore, $\{u \in L_A \cup R_A \mid \ |\Gamma_H(u) \cap \Gamma_H(v)| \text{ is odd}\} = \{d, m_1\}$ which has even cardinality, as required.
- Consider $v \in \Gamma_H(b) \setminus V(B)$.
  - If $u \in \Gamma_H(a) \setminus \{d, m_1\}$ then $u \notin V(B)$ (since $\deg_B(a) = 2$) and, as B is a biconnected component, v and u have no common neighbours.
  - If $u = d$ then $\{u, b\}$ is not an edge of B (by the definition of impasse) so it is not an edge of H (since B is a biconnected component). Hence b is not a common neighbour of u and v. Also, v and u have no other common neighbours since v is not in the biconnected component containing b and d.
  - If $u = m_1$ then the only neighbour of u and v is b since v is not in the biconnected component containing m_1 and b.

  Since $L_A \cup R_A \subseteq \Gamma_H(a)$ and $m_1 \in R_A$ it follows that $\{u \in L_A \cup R_A \mid |\Gamma_H(u) \cap \Gamma_H(v)| \text{ is odd}\} = \{m_1\}$ which has odd cardinality, as required.

□

5.4.2 Good Stops

Lemma 5.39. *Let B be a biconnected component of a graph H. Suppose that, for an* even *non-negative integer k, B is a graph of the form FD_k with vertices as given in Figure 5.7. If $(\{a\}, b)$ is a good stop in B then H has a hardness gadget.*

Proof. By the definition of a good stop, $R_B = \Gamma_H(b) \setminus \{a, c\}$ has odd cardinality and there is a gadget (J_B, z_B) such that $\{v \in V(H) \mid |\hom((J_B, z_B) \to (H, v))| \text{ is odd}\} = \{a\} \cup R_B$.

We give a hardness gadget $(I, S, (J_1, y), (J_2, z), (J_3, y, z))$ for H as follows:

- $I = \{a\}$ and $S = \{b\}$.
- J_1 is the gadget J_B with $y = z_B$ so $\Omega_y = \{a\} \cup R_B$, which has even cardinality, as required.
- J_2 is the graph where z is adjacent to an a-pin and a c-pin so $\Omega_z = \{b, d, z_1, \ldots, z_k\}$, which has even cardinality, as required (since k is even).
- J_3 is an edge between y and z.

Note that a is adjacent to every vertex in Ω_z, and b is adjacent to every vertex in Ω_y, as required. Since $\Omega_y \setminus I = R_B = \Gamma_H(b) \setminus \{a, c\}$ and $\Omega_z \setminus S = \{d, z_1, \ldots, z_k\}$ and B is a biconnected component, there is no edge from $\Omega_y \setminus I$ to $\Omega_z \setminus S$, as required. □

Lemma 5.40. *Let B be a biconnected component of a graph H. Suppose that, for a non-negative integer k, B is a graph of the form BD_k with vertices as given in Figure 5.7. If $(\{a\}, b)$ is a good stop in B then H has a hardness gadget.*

Proof. By the definition of a good stop, $R_B = \Gamma_H(b) \setminus V(B)$ has odd cardinality and there is a gadget (J_B, z_B) such that $\{v \in V(H) \mid |\hom((J_B, z_B) \to (H, v))| \text{ is odd}\} = \{a\} \cup R_B$. We give a hardness gadget $(I, S, (J_1, y), (J_2, z), (J_3, y, z))$ for H as follows:

- $I = \{a\}$ and $S = \{b\}$.
- J_1 is the gadget J_B with $y = z_B$ so $\Omega_y = \{a\} \cup R_B$, which has even cardinality, as required.
- J_2 is the graph where z is adjacent to an a-pin and a c-pin so $\Omega_z = \{b, d\}$, which has even cardinality, as required.
- J_3 is an edge between y and z.

Note that a is adjacent to every vertex in Ω_z, and b is adjacent to every vertex in Ω_y, as required. Since $R_B = \Gamma_H(b) \setminus V(B)$ and B is a biconnected component, note that there are no edges between $\Omega_y \setminus I = R_B$ and $\Omega_z \setminus S = \{d\}$, as required. □

Lemma 5.41. *Let B be a biconnected component of a graph H. Suppose that B is an impasse (Definition 5.30) and that (a, b) is a pair of connectors of B. Let m_1 be the unique common neighbour of a and b in H (see Figure 5.7). Suppose further that $\deg_H(m_1)$ is odd. If $(\{m_1\}, b)$ is a good stop in B then H has a hardness gadget.*

Proof. By the definition of a good stop, $R_B = \Gamma_H(b) \setminus V(B)$ has odd cardinality and there is a gadget (J_B, z_B) such that $\{v \in V(H) \mid |\hom((J_B, z_B) \to (H, v))| \text{ is odd}\} = \{m_1\} \cup R_B$, Using the vertex labels from Figure 5.7, we give a hardness gadget $(I, S, (J_1, y), (J_2, z), (J_3, y, z))$ for H as follows:

- $I = \{m_1\}$ and $S = \{m_1\}$.
- J_1 is the gadget J_B with $y = z_B$ so $\Omega_y = \{m_1\} \cup R_B$, which has even cardinality, as required.

- J_2 is the graph where z is adjacent to an a-pin and an m_2-pin so $\Omega_z = \{m_1, d\}$, which has even cardinality, as required.
- J_3 is a 2-path between y and z.

There are an odd number of 2-walks from m_1 to itself since $\deg_H(m_1)$ is odd by assumption. There are an odd number of 2-walks from m_1 to d since k is odd and no pair of vertices of $S_{k,\ell}$ has common neighbours outside of $S_{k,\ell}$. Since $R_B = \Gamma_H(b)\setminus V(B)$ and B is biconnected there is exactly one 2-walk from m_1 to each vertex in R_B. Thus, for $s \in S = \{m_1\}$, $i \in I = \{m_1\}$, $o \in \Omega_y \setminus I = R_B$, $x \in \Omega_z \setminus S = \{d\}$, we have shown that $|\Sigma_{i,s}|$, $|\Sigma_{o,s}|$ and $|\Sigma_{i,x}|$ are odd, as required. Finally, since B is a biconnected component there are no 2-walks from d to a vertex in R_B and therefore $|\Sigma_{o,x}|$ is even, as required. □

5.4.3 Hardness Results

In this section we establish hardness results which are used to prove Lemma 5.47 in Section 5.4.4.

Lemma 5.42. *Let H be a graph and let B be a biconnected component of H. Suppose that, for an* odd *non-negative integer k, B is a graph of the form FD_k with vertex labels as given in Figure 5.7. If $\deg_H(a)$ is even then H has a hardness gadget.*

Proof. We give a hardness gadget $(I, S, (J_1, y), (J_2, z), (J_3, y, z))$ for H as follows:

- $I = \{a\}$ and $S = \{b, d, z_1, \dots z_k\}$.
- J_1 is the graph where y is adjacent to a b-pin and a d-pin so $\Omega_y = \{a, c\}$, which has even cardinality, as required.
- J_2 is the graph where z is adjacent to an a-pin so $\Omega_z = \Gamma_H(a)$, which has even cardinality, as required.
- J_3 is an edge between y and z.

Note that a is adjacent to every vertex in Ω_z, and each vertex of S is adjacent to every vertex in Ω_y, as required. Since B is a biconnected component there are no edges between $\Omega_y \setminus I = \{c\}$ and $\Omega_z \setminus S = \Gamma_H(a) \setminus V(B)$, as required. □

Lemma 5.43. *Let H be a graph and let A and B be biconnected components of H. Suppose that, for* odd *integers $k \geq 1$ and $\ell \geq 1$, there is an isomorphism f from the graph BD_k to A and an isomorphism g from the graph FD_ℓ to B. Suppose that there is a vertex $w = f(b) = g(a)$ such that $\deg_H(w)$ is odd. Then H has a hardness gadget.*

Proof. We give a hardness gadget $(I, S, (J_1, y), (J_2, z), (J_3, y, z))$ for H as follows:

- $I = \{w\}$ and $S = \{w\}$.
- J_1 is the graph where y is adjacent to an $f(a)$-pin and an $f(c)$-pin so $\Omega_y = \{f(d), f(b)\} = \{f(d), w\}$, which has even cardinality, as required.
- J_2 is the graph where z is adjacent to a $g(b)$-pin and an $g(d)$-pin so $\Omega_z = \{g(c), g(a)\} = \{g(c), w\}$, which has even cardinality, as required.
- J_3 is a 2-path between y and z.

By the fact that A and B are biconnected components, there are exactly $k+2$ walks of length 2 from $f(d)$ to w, and there are exactly $\ell+2$ walks of length 2 from $g(c)$ to w, where k and ℓ are odd. Since $\deg_H(w)$ is odd, there is a an odd number of length-2 walks from w to itself. Finally, there are no length-2 walks from $f(d)$ to $g(c)$, as required. □

Lemma 5.44. *Let H be a graph and let B be a biconnected component of H that is of the form BD_k for some integer $k \geq 0$. Using the vertex names from Figure 5.7, there is a gadget (J, z) such that $\{v \in V(H) \mid |\hom((J,z) \to (H,v))|$ is odd$\} = \Gamma_H(b) \setminus V(B)$.*

Proof. The graph J has three pinned vertices — an a-pin, a b-pin, and a c-pin. The b-pin is adjacent to the vertex z and the other two pins are attached to z by paths of length 2.

We will now consider each $v \in V(H)$ to determine whether $|\hom((J,z) \to (H,v))|$ is odd. Since z is adjacent to a b-pin in J, this can only be true for $v \in \Gamma_H(b)$.

First, consider a vertex $v \in \Gamma_H(b) \cap V(B)$.

- If $v \in \{a, y_1, \ldots, y_k\}$ then v has exactly two length-2 walks to c, so $|\hom((J,z) \to (H,v))|$ is even.
- If $v = c$ then v has exactly two length-2 walks to a so $|\hom((J,z) \to (H,v))|$ is even.

Finally, consider a vertex $v \in \Gamma_H(b) \setminus V(B)$. There is exactly one 2-walk to a, and exactly one 2-walk to c, so $|\hom((J,z) \to (H,v))|$ is odd. □

The following lemma is essentially the same as Lemma 5.44.

Lemma 5.45. *Let H be a graph and let B be a biconnected component of H that is of the form FD_k for some integer $k \geq 0$. Using the vertex names from Figure 5.7, there is a gadget (J, z) such that $\{v \in V(H) \mid |\hom((J,z) \to (H,v))|$ is odd$\} = \Gamma_H(a) \setminus V(B)$.*

Proof. The graph J has three pinned vertices — an a-pin, a b-pin, and a d-pin. The a-pin is adjacent to the vertex z and the other two pins are attached to z by paths of length 2.

We will now consider each $v \in V(H)$ to determine whether $|\hom((J,z) \to (H,v))|$ is odd. Since z is adjacent to an a-pin in J, this can only be true for $v \in \Gamma_H(a)$.

First, consider a vertex $v \in \Gamma_H(a) \cap V(B)$.

- If $v \in \{d, z_1, \ldots, z_k\}$ then v has exactly two length-2 walks to b, so $|\hom((J,z) \to (H,v))|$ is even.
- If $v = b$ then v has exactly two length-2 walks to d so $|\hom((J,z) \to (H,v))|$ is even.

Finally, consider a vertex $v \in \Gamma_H(b) \setminus V(B)$. There is exactly one 2-walk to b, and exactly one 2-walk to d, so $|\hom((J,z) \to (H,v))|$ is odd. □

We obtain the following lemma, which is a generalisation of [69, Lemma 4.5].

Lemma 5.46. *For an integer $q \geq 1$, let $P = v_0, \ldots, v_q$ be a path in a graph H. Suppose that no edge of P is part of a square in H and that $\deg_H(v_j)$ is odd for all $j \in [q-1]$. Suppose that*

1 a) $\deg_H(v_0)$ *is even, or*
1 b) $\deg_H(v_0)$ *is odd and there is a biconnected component B_0 that is isomorphic to BD_k for some* odd *integer $k \geq 1$, where the isomorphism maps v_0 to the vertex b from Figure 5.7.*

Suppose further that

2 a) $\deg_H(v_q)$ *is even, or*
2 b) $\deg_H(v_q)$ *is odd and there is a biconnected component B_{q+1} that is isomorphic to FD_k for some* odd *integer $k \geq 1$, where the isomorphism maps v_q to the vertex a from Figure 5.7.*

Then H has a hardness gadget.

Proof. We give a hardness gadget $(I, S, (J_1, y), (J_2, z), (J_3, y, z))$ for H as follows:

- $I = \{v_1\}$ and $S = \{v_{q-1}\}$.
- If 1 a) holds, then J_1 is the graph where y is adjacent to a v_0-pin so $\Omega_y = \Gamma_H(v_0)$, which has even cardinality as required. If 1 b) holds then (J_1, y) is the gadget from Lemma 5.44 and $\Omega_y = \Gamma_H(v_0) \setminus V(B_0)$, which has even cardinality as required. The vertex v_1 is in Ω_y because the edge $\{v_0, v_1\}$ is not part of a square in H.
- If 2 a) holds, then J_2 is the graph where z is adjacent to a v_q-pin so $\Omega_z = \Gamma_H(v_q)$, which has even cardinality as required. If 2 b) holds then (J_2, z) is the gadget from Lemma 5.45 and $\Omega_z = \Gamma_H(v_q) \setminus V(B_{q+1})$, which has even cardinality as required. The vertex v_{q-1} is in Ω_z because the edge $\{v_{q-1}, v_q\}$ is not part of a square in H.
- J_3 is the path gadget J_P.

This is a hardness gadget by Lemma 5.9. □

5.4.4 Chordal Bipartite Sequence Lemma

Lemma 5.47 (Chordal Bipartite Sequence Lemma)**.** *For an integer $q \geq 1$, let $B_1, \ldots, B_q$ be biconnected components of a graph H and let $b_0, \ldots, b_q$ be vertices such that, for all $i \in [q]$, b_{i-1} and b_i are distinct vertices of B_i, and B_i satisfies one of the following:*

- *B_i is an edge from b_{i-1} to b_i,*
- *B_i is a diamond in which $\{b_{i-1}, b_i\}$ is an edge, or*
- *B_i is an impasse, where (b_{i-1}, b_i) is a pair of connectors of B_i. In this case, let d_i be the unique common neighbour of b_{i-1} and b_i in H.*

If $|\Gamma_H(b_0) \setminus V(B_1)|$ is odd, then at least one of the following holds:

- *B_q is an edge or a diamond and $(\{b_{q-1}\}, b_q)$ is a good start in B_q but not a good stop in B_q,*
- *B_q is an impasse and $(\{d_q\}, b_q)$ is a good start in B_q but not a good stop in B_q, or*
- *H has a hardness gadget.*

Proof. We start by collecting some facts that we will need.
Fact 1. *If $i \in [q]$ and B_i is a diamond, then at least one of the following holds:*

- *for some non-negative integer k there is an isomorphism from FD_k to B_i, mapping the vertex a from Figure 5.7 to b_{i-1} and the vertex b to b_i (we refer to this situation below by saying "B_i is of the form FD_k"), or*
- *for some non-negative integer k there is an isomorphism from BD_k to B_i, mapping the vertex a from Figure 5.7 to b_{i-1} and vertex b to b_i. (We refer to this situation as "B_i is of the form BD_k").*

Fact 2. *If B_1 is an edge or a biconnected component of the form FD_k for an odd integer k then $\Gamma_H(b_0)$ is even.* (This is because b_0 has and odd number of neighbours in B_1 and an odd number outside of B_1, by assumption.)

Let $L_0 = \Gamma_H(b_0) \setminus V(B_1)$ and let B_0 be the subgraph of H induced by the vertices in $L_0 \cup \{b_0\}$. For every $i \in [q]$ such that B_i is an edge or a diamond, let $L_i = \{b_{i-1}\}$. For every $i \in [q]$ such that B_i is an impasse, let $L_i = \{d_i\}$. For every $i \in \{0, \ldots, q\}$, let $R_i = \Gamma_H(b_i) \setminus V(B_i)$.

We start by considering $i \in \{0, 1\}$ and showing that (L_i, b_i) is a good start in B_i or that H has a hardness gadget. We first deal with the easy case $i = 0$. $|L_0|$ is odd by assumption so, to show that (L_0, b_0) is a good start in B_0, it suffices to use the gadget (J, z) in which z is adjacent to a b_0-pin. Note that $R_0 = \Gamma_H(b_0) \cap V(B_1)$. We next deal with $i = 1$ by considering four cases depending on B_1:

- If B_1 is an edge from b_0 to b_1 then, by Lemma 5.35, (L_1, b_1) is a good start in B_1.
- If B_1 is a diamond of the form FD_k for an odd integer $k \geq 0$ then H has a hardness gadget by Fact 2 and Lemma 5.42.
- If B_1 is a diamond of the form FD_k for an even integer $k \geq 0$ then R_0 has even cardinality. Therefore (L_0, b_0) is not a good stop in B_0 and we can apply Lemma 5.36 to show that (L_1, b_1) is a good start in B_1.
- If B_1 is a diamond of the form BD_k then R_0 has even cardinality. Therefore (L_0, b_0) is not a good stop in B_0 and we can apply Lemma 5.37 to show that (L_1, b_1) is a good start in B_1.
- If B_1 is an impasse where (b_0, b_1) is a pair of connectors. Then b_0 has 2 neighbours in B_1 and hence R_0 has even cardinality. Therefore (L_0, b_0) is not a good stop in B_0. Recall that d_1 be the unique common neighbour of b_0 and b_1 in H. If $\deg_H(d_1)$ is even then H has a hardness gadget by Lemma 5.33. Otherwise Lemma 5.38 shows that (L_1, b_1) is a good start in B_1.

For the rest of the proof, let j be the smallest index in $[q]$ that satisfies one of the following properties:

(P1) $|R_j|$ is odd and there is no odd integer k such that B_j is of the form FD_k.

(P2) There is an odd integer k such that B_j is of the form FD_k.

(P3) $j = q$ and $|R_j|$ is even and there is no odd integer k such that B_j is of the form FD_k.

We will use the following claims.

Claim A *Suppose that j does not satisfy (P2). Then H has a hardness gadget or, for all $\ell \in [j]$, the following are satisfied.*

- *(L_ℓ, b_ℓ) is a good start in B_ℓ, and*
- *If $\ell > 1$ then $(L_{\ell-1}, b_{\ell-1})$ is not a good stop in $B_{\ell-1}$.*

Proof: The proof of Claim A is by induction on ℓ. We have already established the base case $\ell = 1$. Now fix $\ell \in \{2, \dots, j\}$ and suppose (from the inductive hypothesis) that $(L_{\ell-1}, b_{\ell-1})$ is a good start in $B_{\ell-1}$. By the minimality of j, $B_{\ell-1}$ is not of the form FD_k for an odd integer k (otherwise $\ell - 1$ would satisfy (P2)). Again, by minimality of j, $|R_{\ell-1}|$ is even (otherwise $\ell-1$ would satisfy (P1)). By the definition of good stop, $(L_{\ell-1}, b_{\ell-1})$ is not a good stop in $B_{\ell-1}$. Since j does not satisfy (P2), B_ℓ is not of the form FD_k for an odd integer k. Thus, we can apply one of Lemmas 5.35, 5.36, 5.37 or 5.38 depending on the form of B_ℓ to show that (L_ℓ, b_ℓ) is a good start in B_ℓ. This completes the proof of Claim A. ■

Claim B *Suppose that B_j satisfies one of the following.*

(B1) B_j is the edge $\{b_{j-1}, b_j\}$ and $\deg_H(b_j)$ is even, or

(B2) there is an odd integer k such that B_j is of the form FD_k.

Suppose that there is an integer ℓ in the range $1 \leq \ell \leq j$ such that, for $i \in \{\ell, \dots, j-1\}$, B_i is the edge $\{b_{i-1}, b_i\}$ and for $i \in \{\ell, \dots, j\}$, $\Gamma_H(b_{i-1})$ and $\Gamma_H(b_{i-1}) \cap V(B_{i-1})$ have odd cardinality. Then H has a hardness gadget or there is an integer p in the range $1 \leq p \leq \ell - 1$ and an odd integer k' such that B_p is of the form $BD_{k'}$. Also, for $i \in \{p+1, \dots, j-1\}$, B_i is the edge $\{b_{i-1}, b_i\}$ where $\Gamma_H(b_{i-1})$ has odd cardinality.

Proof: The proof of Claim B is by induction on ℓ. The base case $\ell = 1$ is vacuous — taking $i = 1$, the precondition of the claim ensures that $|\Gamma_H(b_0)|$ is odd, contrary to Fact 2. So consider some $\ell > 1$ for which we wish to prove the claim. Since taking $i = \ell$ guarantees that $|\Gamma_H(b_{\ell-1}) \cap V(B_{\ell-1})|$ is odd, $B_{\ell-1}$ is either an edge or it is of the form $BD_{k'}$ for an odd integer k'. We consider each case.

- $B_{\ell-1}$ is an edge: If $\deg_H(b_{\ell-2})$ is even H has a hardness gadget by Lemma 5.46 (take $v_0, \dots, v_q = b_{\ell-2}, \dots, b_j$ in Case (B1) and $v_0, \dots, v_q = b_{\ell-2}, \dots, b_{j-1}$ in Case (B2). Thus, assume that $\deg_H(b_{\ell-2})$ is odd. By Fact 2, $\ell - 2 \geq 1$ and consequently (by Claim A), $(L_{\ell-2}, b_{\ell-2})$ is a good start in $B_{\ell-2}$ that is not a good stop in $B_{\ell-2}$. This implies that $|R_{\ell-2}|$ is even, which together with the

fact that $\deg_H(b_{\ell-2})$ is odd implies that $b_{\ell-2}$ has an odd number of neighbours in $B_{\ell-2}$. So the preconditions of the claim are met with $i = \ell - 1$ and we can finish by induction.
- $B_{\ell-1}$ is of the form $BD_{k'}$ for an odd integer k'. The claim follows by taking $p = \ell - 1$.

This concludes the proof of Claim B. ■

We now make a case distinction, depending on which property j satisfies.

Case (P1). We will show that H has a hardness gadget.

By Claim A, either H has a hardness gadget (in which case we are finished) or (L_j, b_j) is a good start in B_j. Since $|R_j|$ is odd, (L_j, b_j) is a good stop in B_j. We now distinguish several cases, depending on the form of B_j.

- If B_j is of the form FD_k for even k, or of the form BD_k, then H has a hardness gadget by Lemmas 5.39 or 5.40, respectively.
- If B_j is an impasse, then depending on the degree of d_j, H has a hardness gadget either by Lemma 5.33 (if the degree of d_j is even) or by Lemma 5.41 (if the degree of d_j is odd).
- Finally, suppose that B_j is an edge. We will use Claim B with $\ell = j$ to show that H has a hardness gadget. The first step is to show that (unless H has a hardness gadget) the preconditions of the claim are met — that is $\deg_H(b_j)$ is even, $\deg_H(b_{j-1})$ is odd, and $|\Gamma_H(b_{j-1}) \cap V(B_{j-1})|$ is odd.

  By (P1), $|R_j|$ is odd. Since b_j has only one neighbour in B_j, $\deg_H(b_j)$ is even. If $\deg_H(b_{j-1})$ is even, H has a hardness gadget by Lemma 5.46 (taking $q = 1$, $v_0 = b_{j-1}$ and $v_1 = b_j$). From now on, we assume that $\deg_H(b_{j-1})$ is odd. By Fact 2, $j - 1 \geq 1$. By the minimality of j, $|R_{j-1}|$ is even, which implies that b_{j-1} has an odd number of neighbours in B_{j-1}.

  Applying Claim B with $\ell = j$, either H has a hardness gadget. or there is an integer p in the range $1 \leq p \leq j - 1$ and an odd integer k' such that B_p is of the form $BD_{k'}$. Also, for $i \in \{p + 1, \dots, j - 1\}$, B_i is the edge $\{b_{i-1}, b_i\}$ where $\Gamma_H(b_{i-1})$ has odd cardinality.

  Now we apply Lemma 5.46 with the path $v_0, \dots, v_q$ equal to $b_p, \dots, b_j$. The degrees of $v_0, \dots, v_{q-1}$ are odd and the degree of v_q is even. v_0 is in the biconnected component B_p. This shows that H has a hardness gadget.

Case (P2). We will use Claim B with $\ell = j$ to show that H has a hardness gadget.

The first step is to show that (unless H has a hardness gadget) the preconditions of the claim are met — that is $\deg_H(b_{j-1})$ is odd, and $|\Gamma_H(b_{j-1}) \cap V(B_{j-1})|$ is odd.

If $\deg_H(b_{j-1})$ is even then H has a hardness gadget by Lemma 5.42. From now on, we assume that $\deg_H(b_{j-1})$ is odd. By Fact 2, $j - 1 \geq 1$. By the minimality of j, $|R_{j-1}|$ is even, which implies that b_{j-1} has an odd number of neighbours in B_{j-1}.

Applying Claim B with $\ell = j$, either H has a hardness gadget. or there is an integer p in the range $1 \leq p \leq j-1$ and an odd integer k' such that B_p is of the form $BD_{k'}$. Also, for $i \in \{p+1, \ldots, j-1\}$, B_i is the edge $\{b_{i-1}, b_i\}$ where $\Gamma_H(b_{i-1})$ has odd cardinality.

If $p = j-1$ then H has a hardness gadget by Lemma 5.43. Otherwise, we apply Lemma 5.46 with the path $v_0, \ldots, v_q$ equal to $b_p, \ldots, b_{j-1}$. The degrees of $v_0, \ldots, v_q$ are odd. v_0 is in the biconnected component B_p and b_q is in the biconnected component B_j. This shows that H has a hardness gadget.

Case (P3) By Claim A, H has a hardness gadget or (L_q, b_q) is a good start in B_q. In the latter case, since $|R_q|$ is even, (L_q, b_q) is not a good stop in B_q.

□

5.5 K_4-minor-free Components

Definition 5.48 (separation, separator)**.** Let G be a graph and let A and B be subsets of $V(G)$. The pair (A, B) is a *separation* of G if $V(G) = A \cup B$ and G has no edges between $A \setminus B$ and $B \setminus A$. The set $A \cap B$ is called the *separator* of this separation.

5.5.1 Induced Cycles

Recall Definition 5.2, which defines for a closed walk $W = (w_0, \ldots, w_{q-1}, w_0)$ in a graph H the walk-neighbour-set $N_{W,H}(w_i) = \Gamma_H(w_{i-1}) \cap \Gamma_H(w_{i+1})$, where the indices are taken modulo q. In this section we will use this notion mainly for cycles.

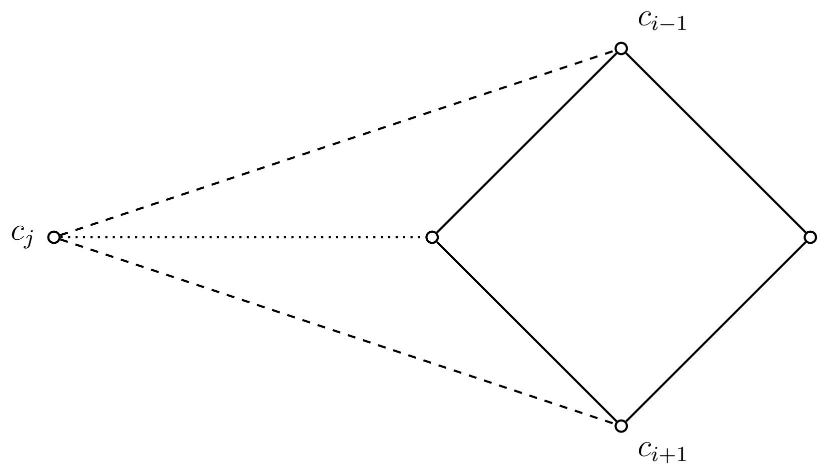
Lemma 5.49. *Let H be a biconnected K_4-minor-free graph containing an induced cycle $C = (c_0, \ldots, c_{q-1}, c_0)$ for some $q \neq 4$. Then the walk-neighbour-sets $N_{C,H}(c_0), \ldots, N_{C,H}(c_{q-1})$ are pairwise disjoint.*

Proof. If $q = 3$ then the fact that we do not allow self-loops in H together with the fact that H does not contain K_4 as a subgraph ensures that the $N_{C,H}(c_i)$ are pairwise disjoint.

Suppose $q > 4$. Assume for contradiction that there exists a vertex $w \in N_{C,H}(c_i) \cap N_{C,H}(c_j)$ for some $i \neq j$. If w is part of the cycle C, then we obtain a chord (note that $q > 4$), contradicting the fact that C is induced. If w is not part of the cycle C, then w is adjacent to at least 3 vertices of the cycle, yielding a K_4-minor. □

Lemma 5.50. *Let H be a biconnected K_4-minor-free graph containing an induced cycle $C = (c_0, \ldots, c_{q-1}, c_0)$. If $q > 4$ and $|N_{C,H}(c_i)| > 1$ for some $i \in \{0, \ldots, q-1\}$ then there exists a separation (A, B) of H such that $C \setminus \{c_i\} \subseteq A$, $N_{C,H}(c_i) \subseteq B$ and $A \cap B = \{c_{i-1}, c_{i+1}\}$. Furthermore, H is a (1,2)-supergraph of $H[B]$.*

Proof. Let $S_1, \dots, S_k$ be the connected components of the graph obtained from H by deleting c_{i-1}, c_{i+1}, and all edges incident to c_{i-1} and c_{i+1}. Then w.l.o.g. we can assume that $c_j \in V(S_1)$ for all $j \notin \{i-1, i, i+1\}$. Set $A = V(S_1) \cup \{c_{i-1}, c_{i+1}\}$ and $B = V(S_2) \cup \dots \cup V(S_k) \cup \{c_{i-1}, c_{i+1}\}$. By construction, we have $A \cup B = V(H)$ and $A \cap B = \{c_{i-1}, c_{i+1}\}$, and there are no edges between $A \setminus B$ and $B \setminus A$. We claim that $N_{C,H}(c_i) \cap V(S_1) = \emptyset$ as, otherwise, we obtain the following K_4-minor; recall that $|N_{C,H}(c_i)| > 1$ and $q > 4$:



Here, the dashed lines depict the path $c_{i-1}, \dots, c_j, \dots, c_{i+1}$ which is $C \setminus \{c_i\}$. Further, c_j is a vertex of C satisfying that there exists a (shortest) path P from a vertex in $N_{C,H}(c_i)$ to c_j such that the internal vertices of P are disjoint from $C \cup N_{C,H}(c_i)$. Note that c_j exists if $N_{C,H}(c_i) \cap V(S_1)$ is not empty. We depict P in the above picture with a dotted line. In particular, P has length at least one, i.e., $c_j \notin N_{C,H}(c_i)$ by Lemma 5.49. Hence we obtain indeed a K_4-minor. Consequently, no vertex of $N_{C,H}(c_i)$ is contained in A, and thus $N_{C,H}(c_i) \subseteq B$.

It remains to show that H is a (1,2)-supergraph of $H[B]$: It is immediate that an edge e between two vertices in B is present in H if and only if it is present in $H[B]$. By the definition of B, H and $H[B]$ cannot have a different number of 2-paths between two different vertices b_1 and b_2 in B, unless $\{b_1, b_2\} = \{c_{i-1}, c_{i+1}\}$. However, regarding the latter case, all common neighbours of c_{i-1} and c_{i+1} are contained in $N_{C,H}(c_i) \subseteq B$ and thus the claim also holds for those two vertices. □

Corollary 5.51. *Let H be a biconnected K_4-minor-free graph containing an induced cycle $C = (c_0, \dots, c_{q-1}, c_0)$. If $q > 4$ then, for all $i \in \{0, \dots, q-1\}$, we have that at least one of $N_{C,H}(c_i)$ and $N_{C,H}(c_{i+1})$ has cardinality* 1.

Proof. Assume for contradiction that for some i, both, $N_{C,H}(c_i)$ and $N_{C,H}(c_{i+1})$, have cardinality greater than 1. We invoke Lemma 5.50 for C and i, which yields a separation (A, B) of H such that $C \setminus \{c_i\} \subseteq A$, $N_{C,H}(c_i) \subseteq B$, and $A \cap B = \{c_{i-1}, c_{i+1}\}$. However, by assumption, there exists $c' \in N_{C,H}(c_{i+1}) \setminus \{c_{i+1}\}$. Note further, that $c' \neq c_{i-1}$ as $q > 4$ and C is induced. Thus there is a path connecting $c_i \in B$ and $c_{i+2} \in A$ which does not pass through either one of c_{i-1} and c_{i+1} contradicting the assumption that (A, B) is a separation with $A \cap B = \{c_{i-1}, c_{i+1}\}$. □

Corollary 5.52. *Let H be a biconnected K_4-minor-free graph containing an induced cycle $C = (c_0, \ldots, c_{q-1}, c_0)$. If $q \neq 4$ and H does not have a hardness gadget then, for all $i \in \{0, \ldots, q-1\}$, we have that at least one of $N_{C,H}(c_i)$ and $N_{C,H}(c_{i+1})$ has cardinality* 1.

Proof. If $q > 4$ the statement follows from Corollary 5.51. If $q = 3$ then C is a triangle and the statement follows directly from Lemma 5.24. □

5.5.2 Pre-Hardness Gadgets and Obstructions

Definition 5.53 (obstruction)**.** Let B be a K_4-minor-free biconnected graph and let C be an induced cycle of B whose length is not 4. We say that B is an *obstruction with cycle C* if every even-cardinality walk-neighbour-set of C in B only contains vertices whose degree in B is 2. We say that B is an obstruction if, for some C, it is an obstruction with cycle C. We use $\mathrm{Cy}(B)$ to denote $\{C \mid B \text{ is an obstruction with cycle } C\}$.

Definition 5.54 (pre-hardness gadget)**.** Let J be a connected graph. We say that J is a *pre-hardness gadget* if, for every (1,2)-supergraph H of J without K_4-minors, H has a hardness gadget.

Note that if J is a biconnected graph that is a pre-hardness gadget, then every K_4-minor-free graph H which contains J as a biconnected component has a hardness gadget.

It will be convenient to establish the following special case of an obstruction.

Lemma 5.55. *Let J be a K_4-minor-free biconnected graph such that the largest induced cycle of J is a square. If J contains a triangle then J is either a pre-hardness gadget or an obstruction.*

Proof. Let (a, b, c, a) be a triangle of J, let $a_1, \ldots, a_k$ be the common neighbours of b and c with $a = a_1$, let $b_1, \ldots, b_\ell$ be the common neighbours of a and c with $b = b_1$, and let $c_1, \ldots, c_m$ be the common neighbours of a and b with $c = c_1$. If at least two of k, ℓ, and m are at least 2, then J is a strong hardness gadget by Lemma 5.24. In particular, every strong hardness gadget is also a pre-hardness gadget. If $k = \ell = m = 1$ then J is a strong (and thus also a pre-) hardness gadget by Corollary 5.14, as follows. Let H be a supergraph of J, let $q = 3$, and let $C = (a, b, c, a)$. Then $|N_{C,H}(a)| = |N_{C,H}(b)| = |N_{C,H}(c)| = 1$ since $k = \ell = m = 1$. Also, suppose for contradiction that there exists a walk $D = (d_a, d_b, d_c, d_a)$ with $d_a \in \Gamma_H(a) \setminus \{b, c\}$, $d_b \in \Gamma_H(b) \setminus \{a, c\}$ and $d_c \in \Gamma_H(c) \setminus \{a, b\}$. Consequently, as we do not allow self-loops in H, $d_a \neq a$, $d_b \neq b$ and $d_c \neq c$. Then the vertices d_a, a, b, c induce a K_4-minor (contract the edges $\{b, d_b\}$ and $\{c, d_c\}$ to obtain a K_4).

Hence assume w.l.o.g. that $k > 1$ and $\ell = m = 1$. If k is odd then J is a pre-hardness gadget by Lemma 5.25. If k is even and all a_j have degree 2 then J is an obstruction. Otherwise, for some $j \in \{1, \ldots, k\}$, let a_j have degree at least 3. As J is biconnected, there exists a shortest (induced) path P of length at least 2 from a_j to one of the vertices b, c or to some a_i with $i \in [k] \setminus \{j\}$. The internal vertices of P are disjoint from b, c and $\{a_i \mid i \in [k]\}$. If the endpoint of P is one of the other a_i, we obtain a K_4-minor, hence the endpoint must be b or c; suppose w.l.o.g. that it is c. As the largest induced cycle of J is a square, P has either length 2 or 3. In the former case, we obtain a strong (and thus also a pre-) hardness gadget by Lemma 5.24. In the latter case, J is a strong (and thus also a pre-) hardness gadget by Lemma 5.26. □

Lemma 5.56. *Let H be a biconnected K_4-minor-free graph. If H contains an induced cycle of length at least 5 then H is either an obstruction or a pre-hardness gadget.*

Proof. We perform induction on $|V(H)|$: Let $C = (c_0, \dots, c_{q-1}, c_0)$ be an induced cycle of length $q \geq 5$. If H is not an obstruction then, by Definition 5.53 there exists i such that $N_{C,H}(c_i)$ has even cardinality and contains a vertex of degree not equal to 2. Assume w.l.o.g. that $i = 1$. So we can assume that $N_{C,H}(c_1) = \{c_1^1, \dots, c_1^k\}$ where $k > 0$ is even and $\deg_H(c_1^1) \neq 2$.

We invoke Lemma 5.50 and obtain a separation (A, B) of H such that $C \setminus \{c_1\} \subseteq A$, $N_{C,H}(c_1) \subseteq B$ and $A \cap B = \{c_0, c_2\}$. Furthermore, H is a (1,2)-supergraph of $H[B]$. Now consider the neighbours of c_1^1: We have that $c_0 \in \Gamma_H(c_1^1)$ and $c_2 \in \Gamma_H(c_1^1)$ by the definition of $N_{C,H}(c_1)$. As $\deg_H(c_1^1) \neq 2$, there exists another neighbour $w \in \Gamma_H(c_1^1)$. By the properties of the separation (A, B), for any $w \in \Gamma_H(c_1^1) \setminus \{c_0, c_2\}$, $w \in B$.

Claim A: *There is a vertex w in $\Gamma_H(c_1^1) \setminus \{c_0, c_2\}$ and an induced path P in $H[B]$ from w to either c_0 or c_2 such that all internal vertices of P are contained in $B \setminus (N_{C,H}(c_1) \cup \{c_0, c_2\})$. Furthermore, no internal vertex of P is a neighbour of c_1^1.*
Proof: Let $w' \in \Gamma_H(c_1^1) \setminus \{c_0, c_2\}$. As H is biconnected, the vertex c_1^1 is not an articulation point. Consequently, there exists a path P' from w' to c_0 not containing c_1^1 as internal vertex. We can assume P' to be induced by taking possible "shortcuts". W.l.o.g. we have that P' does not visit c_2 as internal vertex as, otherwise, we can just continue with c_2 instead of c_0.

Assume first that P' contains a vertex in $A \setminus B$. As (A, B) is a separation and $w' \in B$, we have that P' is of the form

$$w' \overset{P_1}{\to} x \overset{P_2}{\to} c_0 \,,$$

such that P_1 is contained in $H[B]$ and $x \in A \cap B = \{c_0, c_2\}$. However, as P' is a path that does not contain c_2 as internal vertex, we obtain that $P_2 = \emptyset$ and $x = c_0$, contradicting the assumption.

Next assume that P' contains an internal vertex z in $N_{C,H}(c_1) \setminus \{c_1^1\}$; we obtain the contradiction by identifying a K_4-minor in H as depicted in Figure 5.8. We have now shown that there is an induced path P' in $H[B]$ from w' to c_0 or c_2 such that all internal vertices of P' are contained in $B \setminus (N_{C,H}(c_1) \cup \{c_0, c_2\})$. Now choose w to be the first neighbour of c_1^1 along P' from c_0 or c_2, respectively, and let P be the sub-path of P' going from c_0 or c_2, respectively, to w.

■

We assume in the remainder of the proof that the Claim A holds for c_0; the case of c_2 is completely symmetric (by substituting every subsequent appearance of c_0 by c_2 and vice versa). For convenience, we also provide an illustration of our

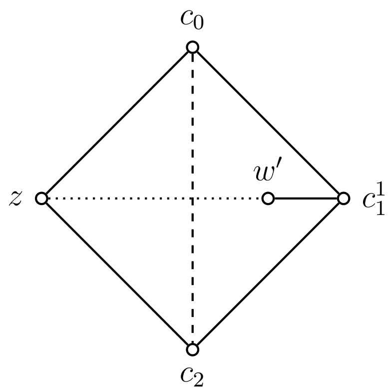


Figure 5.8: The K_4-minor used in the proof of Claim A in Lemma 5.56. The dashed line depicts the remainder of the cycle, and the dotted line depicts P'.

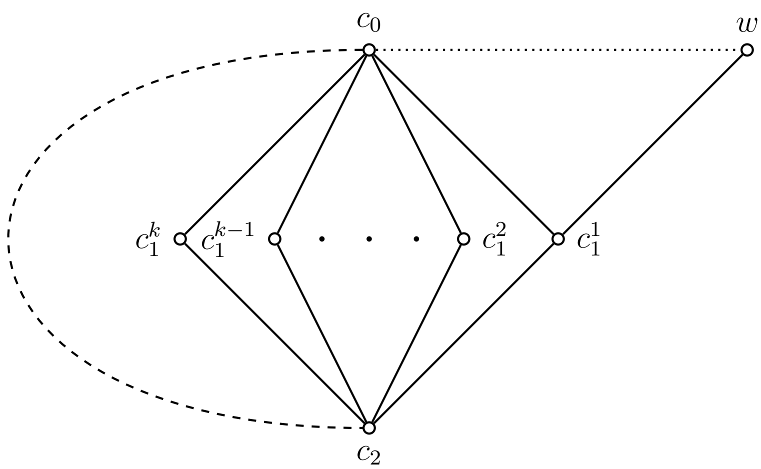


Figure 5.9: Illustration of cycle C consisting of the dashed line and one of the vertices c_1^i, and path P (dotted) in the proof of Lemma 5.56.

current situation in Figure 5.9. For the remainder of the proof, we need the following observation:

Claim B: *$H[B]$ is biconnected.*
Proof: By Menger's Theorem, we have to show that there are two internally vertex-disjoint paths (in $H[B]$) between every pair of different vertices x and y in B. As H is biconnected, there are two such paths Q_1 and Q_2 connecting x and y in H. If $\{x, y\} = \{c_0, c_2\}$, then the claim follows immediately as $N_{C,H}(c_1) \subseteq B$ and $|N_{C,H}(c_i)| \geq 2$.

Hence we can assume that $\{x, y\} \neq \{c_0, c_2\}$. The next step is to show that at least one of Q_1 and Q_2 is fully contained in $H[B]$. If $\{x, y\}$ intersects $\{c_0, c_2\}$ (for example, if $y = c_2$) then this is clear because c_0 can only be on one of Q_1, Q_2. Otherwise, suppose that one of the paths, say Q_2, from x to y, leaves $H[B]$. It leaves by one of the vertices in the separator $\{c_0, c_2\}$ and returns by the other. So Q_1 stays within $H[B]$. If Q_2 is also fully contained in $H[B]$ we are done.

Otherwise we have that w.l.o.g. (otherwise switch c_0 and c_2 and proceed symmetrically):

$$Q_2 = x \overset{Q_2^1}{\to} c_0 \overset{Q_2^2}{\to} c_2 \overset{Q_2^3}{\to} y\,,$$

where Q_2^1 and Q_2^3 are in $H[B]$ and Q_2^2 is non-empty in $H[A \setminus B]$. Next we claim that Q_1 contains at most one vertex in $N_{C,H}(c_1)$ as internal vertex. Assuming otherwise, we have

$$Q_1 = x \overset{Q_1^1}{\to} c_1^1 \overset{Q_1^2}{\to} c_1^2 \overset{Q_1^3}{\to} y\,,$$

where $c_1^1 \neq c_1^2 \in N_{C,H}(c_1)$. As Q_1 is fully contained in $H[B]$, we obtain a K_4-minor, unless Q_1^2 contains c_0 or c_2 as internal vertices: The K_4-minor is induced by c_0, c_1^1, c_1^2, c_2 — note that c_0 is connected to c_2 by $C \setminus \{c_1\}$, and c_1^1 is connected to c_1^2 by Q_1^2.

Thus we can assume that Q_1^2 contains c_0 or c_2 as internal vertices.

- If c_2 is an internal vertex of Q_1^2, then $y \neq c_2$. In this case, however, Q_1 and Q_2 share c_2 as internal vertex, which leads to a contradiction.

- If c_0 is an internal vertex of Q_1^2, then $x \neq c_0$. In this case, however, Q_1 and Q_2 share c_0 as internal vertex, which leads to a contradiction.

Consequently, Q_1 contains at most one vertex in $N_{C,H}(c_1)$. As $N_{C,H}(c_1)$ is of even positive cardinality, there exists hence a vertex $z \in N_{C,H}(c_1)$ which is not part of Q_1. Finally, this enables us to modify Q_2 be substituting Q_2^2 by the path c_0, z, c_2. The resulting path is fully contained in $H[B]$ and, by the previous analysis, internally vertex-disjoint from Q_1. This concludes the proof of Claim B. ■
We proceed with the following claim.

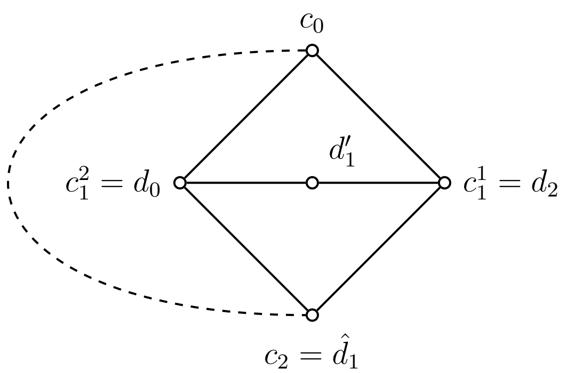
Claim C: *If $H[B]$ is an obstruction, then so is H.*
Proof: If $H[B]$ is an obstruction then it contains an induced cycle D satisfying the requirements of Definition 5.53. For the sake of readability, we state those requirements explicitly: The graph $H[B]$ contains an induced cycle $D = (d_0, \ldots, d_{r-1}, d_0)$ for some $r \neq 4$. Furthermore, we have that for all i, every vertex in $N_{D,H[B]}(d_i)$ has degree 2 in $H[B]$, unless $|N_{D,H[B]}(d_i)|$ is odd.

We claim that H is an obstruction with cycle D: Observe that

$$N_{D,H[B]}(d_i) = \Gamma_{H[B]}(d_{i-1}) \cap \Gamma_{H[B]}(d_{i+1}) = \Gamma_H(d_{i-1}) \cap \Gamma_H(d_{i+1}) = N_{D,H}(d_i)\,,$$

where the second equality is true as H is a (1,2)-supergraph of $H[B]$. Consequently, it remains to show that for all i with $|N_{D,H}(d_i)|$ even, every vertex in $N_{D,H}(d_i)$ has degree 2 in H. For the sake of contradiction, we assume w.l.o.g. that $N_{D,H}(d_1)$ is of even cardinality and contains a vertex $\hat{d}_1$ such that $\hat{d}_1$ has degree 2 in $H[B]$, but degree at least 3 in H. As the separator of (A, B) is $\{c_0, c_2\}$, the only possibility for this to happen is $\hat{d}_1 = c_0$ or $\hat{d}_1 = c_2$. However, $\hat{d}_1 = c_0$ is impossible, as c_0 has at least three neighbours already in $H[B]$: c_0 is adjacent to every vertex in $N_{C,H}(c_1)$, which is of positive even cardinality (i.e., of size at least 2), and c_0 is adjacent to the first vertex in the path P from c_0 to w (see Figure 5.9).

Hence the remaining possibility is $\hat{d}_1 = c_2$. Recall that $\hat{d}_1 = c_2$ has neighbours $c_1^1, \ldots, c_1^k$ in B. So if there are only two of them, then $k = 2$ and $|N_{C,H}(c_1)| = 2$. However, as $\hat{d}_1 \in N_{D,H}(d_1)$ has degree 2 in $H[B]$, and c_1^1 and c_1^2 are adjacent to $\hat{d}_1 = c_2$ in $H[B]$, we obtain that $\{c_1^1, c_1^2\} = \{d_0, d_2\}$ — recall that $\hat{d}_1 \in \Gamma_{H[B]}(d_0) \cap \Gamma_{H[B]}(d_2)$. Finally, $N_{D,H}(d_1)$ has positive, even cardinality. Thus there exists a vertex $d_1' \neq \hat{d}_1$ in $N_{D,H}(d_1)$ which is also adjacent to d_0 and d_2. This yields the following K_4-minor of H; note that $N_{D,H}(d_1) \subseteq B$ and the dashed line is $C \setminus c_1$, which is in A.



This concludes the proof of Claim C. ■

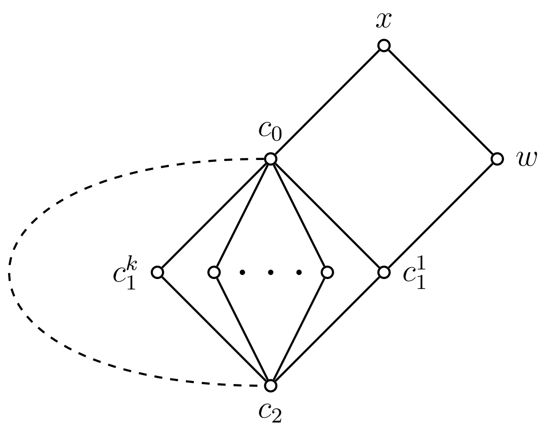
In what follows, we perform a case distinction along the length L of the largest induced cycle in $H[B]$.

(I) $L \geq 5$. This allows us to invoke the induction hypothesis to the graph $H[B]$; note that $|V(H[B])| = |B|$ is indeed strictly smaller than $|V(H)|$ as the cycle C has length at least 5, and thus A is not empty. Furthermore, $H[B]$ is biconnected by Claim B. If $H[B]$ is a pre-hardness gadget, then so is H, as H is a (1,2)-supergraph of $H[B]$. If $H[B]$ is an obstruction, then so is H by Claim C.

(II) $L \leq 4$. Consider again the path P in Figure 5.9. By the assumption of this case, P is either an edge or a 2-path. If P is an edge, then $H[B]$ satisfies all conditions of Lemma 5.55. Consequently, $H[B]$ is a pre-hardness gadget or an obstruction. In the former case, we are done as H is a (1,2)-supergraph of $H[B]$

and thus also a pre-hardness gadget. In the latter case, we obtain that H is an obstruction as well by invoking Claim C.

Finally, assume that P is a 2-path. We claim that H is a pre-hardness gadget. To this end, let H' be a K_4-minor-free (1,2)-supergraph of H. Then H' contains the following subgraph:



In particular, we have that c_1^1 and c_1^k have no common neighbours in H' apart from c_0 and c_2: This is due to the fact that they have no further common neighbours in H, as otherwise H has a K_4-minor similarly as in the proof of Claim A, and as H' is a (1,2)-supergraph, it cannot add common neighbours to vertices. Furthermore, we have that $k > 0$ is even and that $c_1^1, \ldots, c_1^k$ are all common neighbours of c_0 and c_2 in H and thus in H'. We apply Lemma 5.20 to the subgraph of H' induced by the vertices $c_1^k, c_0, x, w, c_1^1, c_2$ and obtain a hardness gadget in H', unless this subgraph, call it F, is of type V. By the previous argument, the only possibility for F being of type V is k being strictly greater than 2. However, as k is even, we obtain a hardness gadget in H' in this case as well: We found an instance of Lemma 5.22.

□

5.5.3 K_4-minor-free Component Lemma

Lemma 5.57 (K_4-minor-free Component Lemma)**.** *Let B be a biconnected K_4-minor-free graph. If B is not an edge then at least one of the following is true:*

(a) B is a diamond.

(b) B is an obstruction.

(c) B is an impasse.

(d) For every K_4-minor-free graph H containing B as a biconnected component, H has a hardness gadget.

Proof. Let L be the size of the largest induced cycle of B. Note that $L \geq 3$ is well-defined as B is biconnected, but not an edge. If $L \geq 5$ we obtain by Lemma 5.56 that B is either a pre-hardness gadget or an obstruction. In the latter case, (b) holds. In the former case, (d) holds, as every K_4-minor-free graph H containing B as a biconnected component is a (1,2)-supergraph of B.

If $L \leq 4$ and B contains a triangle, then B is either a pre-hardness gadget or an obstruction by Lemma 5.55. Similarly as before, (b) or (d) hold.

In the remaining case, B is chordal bipartite and we can invoke Lemma 5.32, yielding that either (a), (c) or (d) hold. □

5.6 K_4-minor-free Graphs

5.6.1 Suitable Connectors

Definition 5.58 (suitable connector)**.** Let H be a graph, let B be a biconnected component of H, and let $A \subseteq V(B)$ be a set of articulation points of H. We say that (B, A) is a *suitable connector* in H if one of the following cases holds:

- B is an edge $\{a, b\}$ and $A = \{a, b\}$, or
- B is a diamond (Definition 5.31) that contains an edge $\{a, b\}$ such that $A = \{a, b\}$, or
- B is an impasse (Definition 5.30) that has a pair of connectors (a, b) such that $A = \{a, b\}$, or
- B is an obstruction (Definition 5.53). In this case (B, A) is a suitable connector in H if there is a cycle $C \in \mathrm{Cy}(B)$ such that

$$A = \{c \in C \mid \text{the cardinality of } N_{C,H}(c) \text{ is even}\}.$$

Note that A could be the empty set. If (B, A) is a suitable connector in H then we fix a particular cycle $C(B, A) \in \mathrm{Cy}(B)$ such that

$$A = \{c \in C(B, A) \mid \text{the cardinality of } N_{C(B,A),H}(c) \text{ is even}\}.$$

(It does not matter if there are multiple possibilities for $C(B, A)$ in $\mathrm{Cy}(B)$ — we just fix one, for example, the lexicographically least one.)

Lemma 5.59. *Let B be a biconnected component in an involution-free graph H. If B is a diamond then there exists a set $A \subseteq V(B)$ of articulation points of H such that (B, A) is a suitable connector in H.*

Proof. If B is a diamond with vertices as given in Definition 5.31, then as H is involution-free there exist articulation points $a \in \{s, t\}$ and $b \in \{x_1, \ldots, x_k\}$. Hence, for $A = \{a, b\}$, (B, A) is a suitable connector in H. □

Lemma 5.60. *Let B be a biconnected component in an involution-free graph H. If B is an impasse then there exists a set $A \subseteq V(B)$ of articulation points of H such that (B, A) is a suitable connector in H.*

Proof. If B is an impasse with vertices as given in Definition 5.30, then as H is involution-free there exist articulation points $a \in \{v_1, y_1, \ldots, y_k\}$ and $b \in \{v_3, z_1, \ldots, z_\ell\}$. Note that (a, b) is a pair of connectors (cf. Definition 5.30) and hence, for $A = \{a, b\}$, (B, A) is a suitable connector in H. □

Lemma 5.61. *Let B be a biconnected component in an involution-free graph H. If B is an obstruction then there exists a set $A \subseteq V(B)$ of articulation points of H such that (B, A) is a suitable connector in H.*

Proof. If B is an obstruction then there exists a cycle C with $C \in \mathrm{Cy}(B)$. Let $c \in C$ such that $|N_{C,H}(c)|$ is even. By definition of an obstruction, every vertex in $|N_{C,H}(c)|$ has degree 2 in B. Since $c \in N_{C,H}(c)$, $|N_{C,H}(c)| \geq 2$. Therefore, as H is involution-free, at least one vertex in $N_{C,H}(c)$ is an articulation point of H. By renaming vertices, we can assume without loss of generality that c is an articulation point. Hence, for $A = \{c \in C \mid \text{the cardinality of } N_{C,H}(c) \text{ is even}\}$, (B, A) is a suitable connector in H, where $C(B, A) = C$. □

5.6.2 Finding a Suitable Subtree

In this section we will use the notion of rooted trees. Given a tree T and a vertex r in T, (T, r) is a *rooted tree* and the *tree-order* $<_r$ induced by r (on T) is the partial order of the vertices of T, where for vertices u and v of T we have $u <_r v$ if and only if the unique path from r (the root) to v passes through u. Such a partial order gives rise to the standard notion of child, parent, ancestor and descendant. In order to clarify which tree-order we are referring to we speak of an r-child, r-parent, r-ancestor and r-descendant when we mean child, parent, ancestor and descendant with respect to $<_r$.

For a connected graph H, recall the definition of the block-cut tree $\mathsf{BC}(H)$ from Definition 5.3.

Definition 5.62 (R-open, R-closed)**.** Let H be a connected graph, let a be a cut vertex in $\mathsf{BC}(H)$, and let R be a block in $\mathsf{BC}(H)$. If a has exactly one descendant with respect to $<_R$ in $\mathsf{BC}(H)$ and this descendant is a block in $\mathsf{BC}(H)$ that is an edge, then a is *R-closed* (in $\mathsf{BC}(H)$). Otherwise, a is *R-open* (in $\mathsf{BC}(H)$).

Definition 5.63 (suitable subtree, closed)**.** Let H be a connected graph. Let T be a subtree of $\mathsf{BC}(H)$. We say that T is *suitable* if it has the following properties:

1. For every block B in T, $(B, \Gamma_T(B))$ is a suitable connector in H (Definition 5.58).
2. Every cut vertex of T has degree at most 2 in T.

A suitable subtree T is *closed* if there exists a block R in T such that every cut vertex that is a leaf in T is R-closed in $\mathsf{BC}(H)$.

Lemma 5.64. *Let H be a connected graph and let T be a suitable subtree of* $\mathsf{BC}(H)$. *Let R and R' be distinct blocks in T and let a be a cut vertex that is a leaf in T. If a is R-closed in* $\mathsf{BC}(H)$ *then it is R'-closed in* $\mathsf{BC}(H)$.

Proof. Let B be a block of $\mathsf{BC}(H)$. We show that B is an R-descendant of a in $\mathsf{BC}(H)$ if and only if it is an R'-descendant. From this it follows immediately that if a is R-closed in $\mathsf{BC}(H)$ then it is R'-closed in $\mathsf{BC}(H)$. Let B be an R-descendant of a. Since a is a leaf of T and R is in T it follows that B is not in T. Since R, R' and a are all in T, there is a path in T from R' to a and consequently this path does not contain B. Hence the unique path from R' to B goes through a, which means that B is an R'-descendant of a in $\mathsf{BC}(H)$. It is analogous to show that if B is an R'-descendant of a it is also an R-descendant. □

The following lemma gives the initialisation for finding a closed suitable subtree (which is then done in Lemma 5.66).

Lemma 5.65. *Let H be an involution-free, connected graph such that every biconnected component of H is an edge, a diamond, an impasse or an obstruction. Then there exists a biconnected component B_0 and a set of articulation points $A_0 \subseteq V(B_0)$ such that (B_0, A_0) is a suitable connector in H and hence $T(B_0) = (\{B_0\} \cup A_0, \{\{B_0, a\} \mid a \in A_0\})$ is a suitable subtree of $\mathsf{BC}(H)$.*

Proof. First note that if all biconnected components of H are edges, then there is at least one edge between articulation points as H is involution-free and therefore H is not a star. Therefore, H contains a biconnected component R that is one of the following: a diamond, an impasse, an obstruction, or an edge for which both endpoints are articulation points of H. In the first three cases we can use Lemmas 5.59, 5.60 or 5.61, respectively, to obtain a suitable connector. If B_0 is an edge $\{a, b\}$ where both end points are articulation points, then $(B_0, \{a, b\})$ is a suitable connector. Then it is immediate that $T(B_0)$ is a suitable subtree of $\mathsf{BC}(H)$. □

Lemma 5.66. *Let H be an involution-free, connected graph such that every biconnected component of H is an edge, a diamond, an impasse or an obstruction. Then there exists a closed suitable subtree of $\mathsf{BC}(H)$.*

Proof. Let $B_0, A_0, T(B_0)$ be as given by Lemma 5.65. Algorithm 2 keeps track of a suitable subtree T of $\mathsf{BC}(H)$, a block R of T, and the set $A(T)$ of leaves of T that are cut vertices (i.e., that are articulation points of H).

We now show that Algorithm 2 (see page 200) is well-defined and finds a closed suitable subtree. In order to show that the algorithm is well-defined note that any R-open cut vertex a^* is an articulation point of H and therefore is adjacent to at least two blocks of $\mathsf{BC}(H)$. At most one of these blocks can be an R-parent. Therefore a^* has an R-child in $\mathsf{BC}(H)$. If there is such an R-child B that is a diamond, an impasse, or an obstruction, then by Lemmas 5.59, 5.60 or 5.61, respectively, there exists a suitable connector of the form (B, A). If otherwise all R-children of a^* are edges then a^* has at least one such R-child $B = \{a^*, b\}$ for which b is an articulation point (as a^* is R-open and H is involution-free). Therefore $(B, \{a^*, b\})$ is a suitable connector. Thus, the algorithm is well-defined as we can always choose a suitable connector (B, A) where B is an R-child of a^*.

We next show that at any point during the algorithm, T is a suitable subtree of $\mathsf{BC}(H)$, R is a block in T, and $A(T)$ is the set of leaves of T that are cut vertices of

$\mathsf{BC}(H)$. First note that in the initialisation this clearly holds by Lemma 5.65. We show that after each update these properties still hold. Note that if we update T, R, and $A(T)$ as part of the else-block then $R = B$ is the only block in T, $\Gamma_T(B) = A$, and (B, A) is a suitable connector. Thus, T is a suitable subtree. Furthermore, the cut vertex leaves of T are precisely the elements of A and we have $A(T) = A$, as required.

Algorithm 2

```
T ← T(B_0)
R ← B_0
A(T) ← A_0
while A(T) contains an R-open cut vertex a*
    // Invariant: All elements of A(T) are B_0-descendants of R.
    if there is a suitable connector (B, A) in H such that B is an R-child of a* and a* ∈ A
        // By the invariant, every element of A \ {a*} is a B_0-descendant of a*.
        V ← V(T) ∪ {B} ∪ A
        E ← E(T) ∪ {{B, a} | a ∈ A}
        T ← (V, E)
        A(T) ← (A(T) ∪ A) \ {a*}
    else
        Choose a suitable connector (B, A) in H such that B is an R-child of a* in BC(H).
        // By the invariant, every element of A is a B_0-descendant of a*.
        V ← {B} ∪ A
        E ← {{B, a} | a ∈ A}
        T ← (V, E)
        R ← B
        A(T) ← A
```

If otherwise we update T and $A(T)$ as part of the if-block then

1. The block R continues to be a vertex of T.
2. We add precisely one block B together with the articulation points A and the edges $\{\{B, a\} \mid a \in A\}$, which ensures that $\Gamma_T(B) = A$ and hence $(B, \Gamma_T(B))$ is a suitable connector.
3. All cut vertices in $A \setminus \{a^*\}$ are leaves in T and since a^* was a leaf before the update, it now has degree 2 in T.

Consequently, T is a suitable subtree after the update. Furthermore, we remove a^* from $A(T)$ as it now has degree 2 in T, and we add the cut vertices $A \setminus \{a^*\}$ to $A(T)$ since they are leaves in T.

We have established that at any point during the algorithm, T is a suitable subtree of $\mathsf{BC}(H)$, R is a block in T, and $A(T)$ is the set of leaves of T that are cut vertices. It remains to show that Algorithm 2 terminates (in which case it is immediate that T is a closed suitable subtree). Note that with each iteration we remove a vertex a^*

from $A(T)$. With each iteration we may also add some vertices to $A(T)$. As noted in Algorithm 2, the vertices that are added in each iteration are always B_0-descendants in $\mathsf{BC}(H)$ of the vertex a^* that is deleted. It follows immediately that Algorithm 2 terminates as we only consider finite graphs. □

5.6.3 Suitable Subtrees without Obstructions

Lemma 5.67. *Let H be a connected graph and let T be a closed suitable subtree of $\mathsf{BC}(H)$. If no block of T is an obstruction then H has a hardness gadget.*

Proof. As T does not contain an obstruction, the degree of every block in T is 2. Together with the fact that every cut vertex has degree at most 2, this implies that, for a non-negative integer q, T is a path of the form $(b_0, B_1, b_1, B_2, \ldots, B_q, b_q)$, where $B_1, \ldots, B_q$ are blocks, i.e. biconnected components of H, and $b_0, \ldots, b_q$ are cut vertices, i.e. articulation points of H. Since T is closed it contains at least one block R and therefore $q \geq 1$. Furthermore, for each $i \in [q]$, $(B_i, \{b_{i-1}, b_i\})$ is a suitable connector. And since B_i is no obstruction, one of the following holds:

- B_i is an edge $\{b_{i-1}, b_i\}$, or
- B_i is a diamond that contains the edge $\{b_{i-1}, b_i\}$, or
- B_i is an impasse such that (b_{i-1}, b_i) is a pair of connectors.

Since T is closed, there is a block R among $B_1, \ldots, B_q$ such that both b_0 and b_q are R-closed. By Lemma 5.64, b_0 is B_1-closed and b_q is B_q-closed. It follows that $|\Gamma_H(b_0) \setminus V(B_1)| = 1$ and $|\Gamma_H(b_q) \setminus V(B_q)| = 1$.

Thus, we can apply Lemma 5.47 to obtain that H has a hardness gadget or otherwise there exists $L_q \subseteq \Gamma_{B_q}(b_q)$ such that (L_q, b_q) is a good start in B_q. Since $\Gamma_H(b_q) \setminus V(B_q)$ has odd cardinality, this means that (L_q, b_q) is a good stop in B_q. Then Lemma 5.47 ensures that H has a hardness gadget in this case as well. □

5.6.4 Suitable Subtrees with Obstructions

The goal of this section is to prove Lemma 5.81, which gives a hardness gadget in a connected K_4-minor-free graph using a closed suitable subtree that contains an obstruction. In order to find this hardness gadget we use Lemma 5.13, which derives a hardness gadget based on the generalised cycle gadget from Definition 5.11. The sets of vertices $\mathcal{C}_0, \ldots, \mathcal{C}_{q-1}$ from Lemma 5.13 will correspond to the walk-neighbour-sets of a specific closed walk W. With Algorithms 3 and 4 we define this walk W — it is the output of Algorithm 4. In Lemmas 5.74, 5.75 and 5.76 we establish that the algorithms are well-defined and give as output a closed walk in H whose length is at least 3, and not equal to 4. In Figure 5.10 we give an example that illustrates how W is derived. In Lemmas 5.79 and 5.80 we then show that the walk-neighbour-sets of W satisfy certain properties required to apply Lemma 5.13. In the proof of Lemma 5.81 we put all the pieces together and establish the remaining necessary properties of W.

Definition 5.68 (obstruction-free path, proper)**.** Let H be a connected graph and let T be a closed suitable subtree of $\mathsf{BC}(H)$. A path in T is *obstruction-free* if it does not contain a block that is an obstruction. An obstruction-free path is *proper* if its endpoints are cut vertices of $\mathsf{BC}(H)$. Note that it is possible that a proper obstruction-free path has length 0. Then it is of the form (v) where v is a cut vertex of $\mathsf{BC}(H)$.

Definition 5.69 ($\mathrm{P}_H(a,b)$)**.** Let H be a graph and let a and b be vertices of H. If $a = b$ then $\mathrm{P}_H(a,b) = (a)$. If $a \neq b$ then $\mathrm{P}_H(a,b)$ is a shortest path from a to b in H.

In Definition 5.69, it is of course possible that H might have multiple shortest paths from a to b. In this case, it doesn't matter which of these is chosen to be $\mathrm{P}_H(a,b)$ — for concreteness, the reader may assume that $\mathrm{P}_H(a,b)$ is the lexicographically least of these. (In fact, when we use the definition, this shortest path will turn out to be unique.)

Lemma 5.70. *Let H be a connected graph and let T be a closed suitable subtree of $\mathsf{BC}(H)$. For a non-negative integer q, let $P = (b_0, B_1, b_1, B_2, \ldots, B_q, b_q)$ be a proper obstruction-free path in T. Then $\mathrm{P}_H(b_0, b_q)$ is the unique shortest path from a to b in H. It passes through $b_0, b_1, \ldots, b_q$ in order. For $i \in [q]$, the subpath of $\mathrm{P}_H(b_0, b_q)$ that connects b_{i-1} and b_i is either an edge or it is of the form (b_{i-1}, v, b_i), where v is the unique common neighbour of b_{i-1} and b_i in H.*

Proof. Since P is a proper obstruction-free path, $b_0, b_1 \ldots, b_q$ are cut vertices and $B_1, \ldots, B_q$ are blocks. Since T is a suitable subtree and P is obstruction free, for each $i \in [q]$, $(B_i, \{b_{i-1}, b_i\})$ is a suitable connector, where B_i is an edge, diamond or impasse. Since $B_1, \ldots, B_q$ are biconnected components, every path from b_0 to b_q traverses $b_0, b_1 \ldots, b_q$ in order. The shortest path from b_0 to b_q is unique, if for each $i \in [k]$, the shortest path from b_{i-1} to b_i is unique. If B_i is an edge or diamond, this is clearly the case since then the shortest path from b_{i-1} to b_i is an edge. If B_i is an impasse then (b_{i-1}, b_i) is a pair of connectors of B_i and by Definition 5.30 there is no edge between b_{i-1} and b_i, but there is a unique common neighbour v of b_{i-1} and b_i in H and consequently the unique shortest path from b_{i-1} to b_i is of the form (b_{i-1}, v, b_i), as required. □

Definition 5.71 (attachment point, exit, destination)**.** Let H be a connected graph and let T be a closed suitable subtree of $\mathsf{BC}(H)$. Let a be a cut vertex that has an obstruction B as a neighbour in T. Then, since every cut vertex of T has degree at most 2, there is a unique maximal-length proper obstruction-free path P^* in T starting at a. Let b be the other endpoint of P^* (possibly $P^* = (a)$ in which case $b = a$). The vertex a is an *attachment point of* (T, B) if b is a leaf in T. Otherwise, a is an *exit of* (T, B). In this case, b is adjacent to a block $B' \neq B$ which is an obstruction. We say that (b, B') is the *destination* of a in T.

At the beginning of this section we outlined our plan to define a particular closed walk W. We chose the names in Definition 5.71 since W will *exit* an obstruction when it encounters an exit, and it will then proceed towards the destination of that exit. The walk W will not exit an obstruction when it encounters an attachment point. However, W will be designed so that every even-cardinality walk-neighbour-set of W

contains an attachment point, and the structure that is *attached* to such a point will allow us to construct a hardness gadget.

Definition 5.72 (concatenation "+")**.** Let $W = (w_0, \dots, w_k)$ and $W' = (w'_0, \dots, w'_\ell)$ be two walks such that $w_k = w'_0$. If $k = 0$ then the *concatenation* $W + W'$ of W with W' is equal to W'. Similarly, if $\ell = 0$, it is equal to W. If both k and ℓ are positive then $W + W' = (w_0, \dots, w_{k-1}, w_k, w'_1 \dots, w'_\ell)$.

Definition 5.73 ($D(C)$, $\mathrm{W}_C(a)$, $\mathrm{W}_C(a, b)$)**.** For an integer $q \geq 3$, let $C = (c_0, \dots, c_{q-1}, c_0)$ be a cycle in a graph H. Then $D(C)$ is the cyclic order induced by the order in which the walk C traverses the vertices $\{c_0, \dots, c_{q-1}\}$. For $a \in C$, $\mathrm{W}_C(a)$ is the walk from a to itself following all of the vertices of C in the order given by $D(C)$. For $a, b \in C$, $\mathrm{W}_C(a, b)$ is the walk from a to b along C in the order given by $D(C)$.

Algorithm 3 $\textsc{ExitWalk}(T, a^*, B, \ell, a_0)$

Input: A closed suitable subtree T of $\mathsf{BC}(H)$ of a connected graph H, a cut vertex a^* in T, an obstruction B that is a block in T such that $\mathsf{dist}_T(a^*, B) = \ell$, and an exit a_0 of (T, B)

```
C ← C(B, Γ_T(B))
{a_0, ..., a_k} ← The exits of (T, B) in the order of D(C), starting from a_0
if k = 0
    W ← W_C(a_0).
else
    {(b_1, B_1), ..., (b_k, B_k)} ← The destinations of a_1, ..., a_k, respectively
    W ← W_C(a_0, a_1)
    for i = 1, ..., k
        r_i ← dist_T(B, b_i)
        W ← W + P_H(a_i, b_i) + ExitWalk(T, a*, B_i, ℓ + r_i + 1, b_i) + P_H(b_i, a_i)+
                W_C(a_i, a_{i+1 mod k+1})
```

Output: W

Algorithm 4 $\textsc{Walk}(T, B')$

Input: A closed suitable subtree T of $\mathsf{BC}(H)$ of a connected graph H, an obstruction B' that is a block in T

```
if there is an exit a* of (T, B')
    (b*, B*) ← The destination of a*
    r* ← dist_T(a*, b*)
    W ← ExitWalk(T, a*, B', 1, a*) + P_H(a*, b*)+
            ExitWalk(T, a*, B*, r* + 1, b*) + P_H(b*, a*)
else
    W ← C(B', Γ_T(B'))
```

Output: W

In Figure 5.10 we provide some illustrations of a graph H, a closed suitable subtree $T \in \mathsf{BC}(H)$, and the walk W returned by Algorithm 4. In order to gain intuition, it is probably useful to simulate $\textsc{Walk}(T, O_1)$. The exit a^* can be chosen to be a_2 with destination $(b^*, B^*) = (b_2, O_2)$. The variable r^* is set to 0. So the first part of W is the output of the call $\textsc{ExitWalk}(T, a_2, O_1, 1, a_2)$.

Let's start by considering that call. $\Gamma_T(O_1) = \{t_1, a_1, a_2\}$ and C is the cycle around O_1 shown in red. The exits are $\{a_2, a_1\}$ so the output W of this call starts by following the red cycle clockwise from a_2 to a_1. In the else-clause we have $k = 1$ and the destination of a_1 is (b_1, O_3). The walk next takes the unique shortest path from a_1 to b_1. Then there is a call to $\textsc{ExitWalk}(T, a_2, O_3, \ell, b_1)$, for some value of ℓ (the value of ℓ doesn't matter — it is just for accounting). The only exit of (T, O_3) is b_1, so this call returns a walk around the red cycle in O_3 from b_1 to itself. Finally, the call to $\textsc{ExitWalk}(T, a_2, O_1, 1, a_2)$ takes the unique shortest path back from b_1 to a_1 and finishes the red cycle in O_1 clock-wise, back to a_2. Thus, the output of $\textsc{ExitWalk}(T, a_2, O_1, 1, a_2)$ is a closed walk from a_2 to itself that covers all of the red edges in the picture, apart from the triangle in O_2.

This is concatenated with $P_H(a_2, b_2) = (a_2)$ (which does nothing). Then it is concatenated with the output of a call to $\textsc{ExitWalk}(T, a_2, O_2, 1, a_2)$. Now $(O_2, \{a_2\})$ is a suitable connector in H with $C(O_2, \{a_2\})$ equal to the red triangle in O_2, so C is assigned to be this triangle. The cut-vertex a_2 is the only exit of C, so this call returns the walk from a_2 to itself around C. Concatenating $P_H(a_2, b_2)$ with this does not change the output. The entire walk is coloured in red.

We now proceed to establish the correctness of Algorithms 3 and 4, and to prove some properties of the walks that they output.

Lemma 5.74. *All calls to* $\textsc{ExitWalk}(\cdot)$ *in Algorithms 3 and 4 have arguments that are feasible inputs to Algorithm 3.*

Proof. First, consider Algorithm 3, where for $i \in [k]$ we make a call $\textsc{ExitWalk}(T, a^*, B_i, \ell{+}r_i{+}1, b_i)$. Observe that b_i is an exit of (T, B_i) by the definition of a destination (Definition 5.71). It remains to check that $\mathsf{dist}_T(a^*, B_i) = \ell + r_i + 1$. This is true since $\ell = \mathsf{dist}_T(a^*, B)$ and $r_i = \mathsf{dist}_T(B, b_i)$ using the fact that the (unique) path from a^* to B_i in the tree T goes from a^* to B then from B to b_i and then from b_i to B_i, where B_i is adjacent to b_i.

Second, consider Algorithm 4, where the if-block makes two calls to $\textsc{ExitWalk}(\cdot)$ — one is $\textsc{ExitWalk}(T, a^*, B', 1, a^*)$ and the other is $\textsc{ExitWalk}(T, a^*, B^*, r^* + 1, b^*)$. Observe that a^* is an exit of (T, B') by the condition of the if-block and b^* is an exit of (T, B^*) by the definition of a destination. It remains to check that $\mathsf{dist}_T(a^*, B') = 1$ and $\mathsf{dist}_T(a^*, B^*) = r^* + 1$. The former is immediate since a^* is adjacent to B' in T. The latter is true since $r^* = \mathsf{dist}_T(a^*, b^*)$ and B^* is adjacent to b^* in T where the (unique) path from a^* to B^* goes via b^*. □

Lemma 5.75. $\textsc{ExitWalk}(T, a^*, B, \ell, a_0)$ *(Algorithm 3) terminates, is well-defined, and returns a closed walk in H of length at least* 3 *from a_0 to itself.*

Proof. First consider the case $k = 0$. Clearly, Algorithm 3 terminates and is well-defined. It returns $\mathrm{W}_C(a_0)$, which is a cycle from a_0 to itself of length at least 3. Now

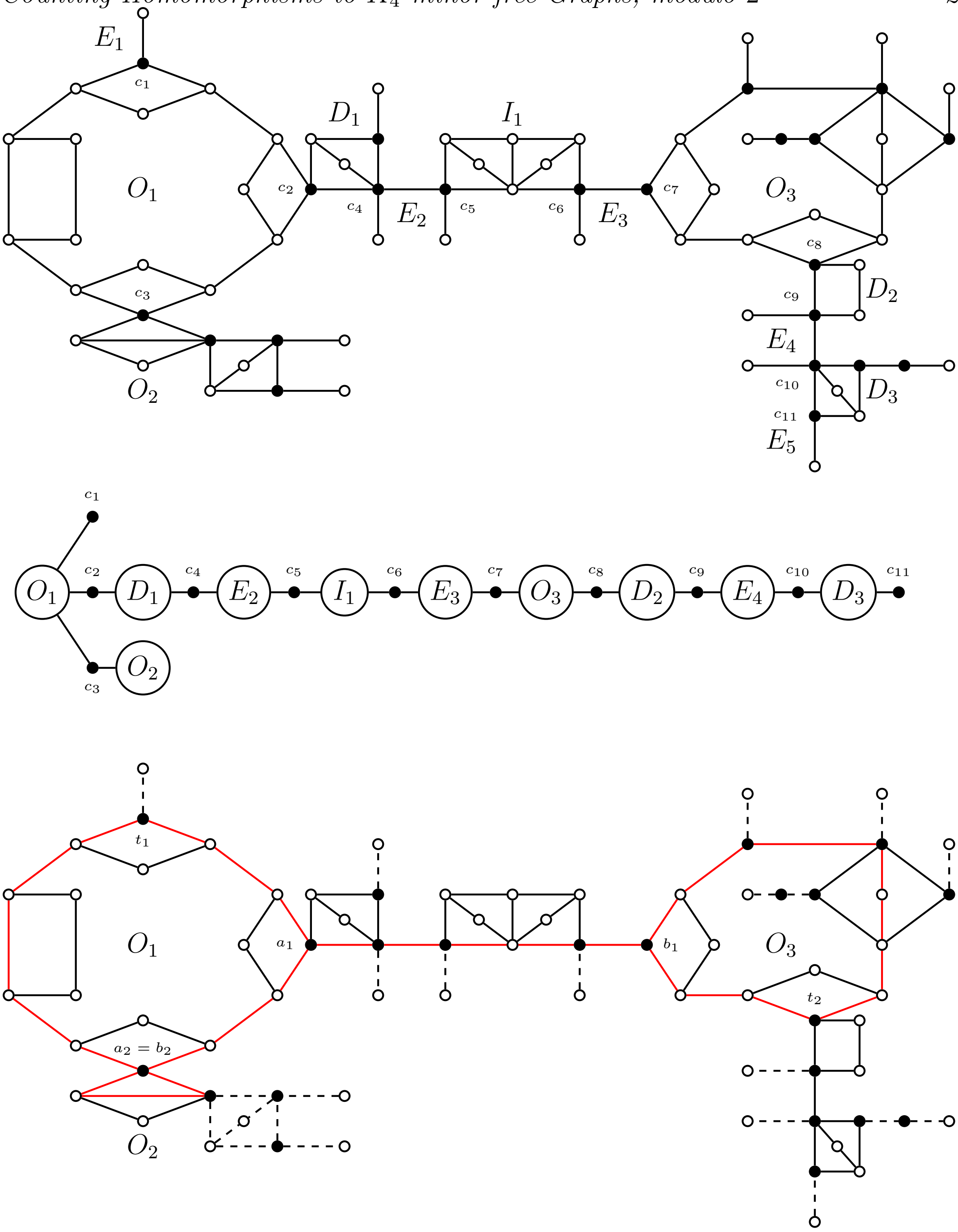


Figure 5.10: *A graph H, a closed suitable subtree T of* $\mathsf{BC}(H)$ *with a block O_1 that is an obstruction, and* $\textsc{Walk}(T, O_1)$.
(Top) An involution-free and K_4-minor-free graph H such that every biconnected component is an edge, a diamond, an impasse or an obstruction. Articulation points are depicted as filled vertices.
(Center) A *closed* and *suitable* subtree T of the block-cut tree of H, rooted at O_1. Note that every cut vertex of T that is a leaf (i.e., c_1 or c_{11}) is O_1-closed in $\mathsf{BC}(H)$.
(Bottom) Solid lines are contained in the subgraph of H induced by $V(T)$, while dashed lines are not. The red closed walk is the output of $\textsc{Walk}(T, O_1)$. Observe that a_1 and a_2 are exits of (T, O_1) with destinations (b_1, O_3) and (b_2, O_2), respectively, and that t_1 and t_2 are attachment points of (T, O_1) and (T, O_3), respectively.

consider the case where $k \geq 1$. Note that with each recursive call of $\textsc{ExitWalk}(\cdot)$ the value of the parameter ℓ increases. Since ℓ corresponds to the distance between a^* and B in the finite graph T, Algorithm 3 terminates.

We now show that Algorithm 3 returns a closed walk of length at least 3 from a_0 to itself. If, for $i \in [k]$, $\textsc{ExitWalk}(T, a^*, B_i, \ell + r_i + 1, b_i)$ returns a closed walk from b_i to itself of length at least 3 then $\mathrm{P}_H(a_i, b_i) + \textsc{ExitWalk}(T, a^*, B_i, \ell + r_i + 1, b_i) + \mathrm{P}_H(b_i, a_i) + \mathrm{W}_C(a_i, a_{i+1 \bmod k+1})$ is a walk from a_i to $a_{i+1 \bmod k+1}$ of length at least 3. Thus, $\textsc{ExitWalk}(T, a^*, B, \ell, a_0)$ returns a closed walk from a_0 to itself of length at least 3. Since Algorithm 3 terminates, it reaches the base of the recursion, i.e., the case $k = 0$, at some point, and we have already verified that the base case returns a closed walk of length at least 3, as required.

Finally, we show that Algorithm 3 is well-defined. By Lemma 5.74 all subroutine calls have feasible inputs. Also observe that all concatenation operations are well-defined since, for each $i \in [k]$, $\textsc{ExitWalk}(T, a^*, B_i, \ell + r_i + 1, b_i)$ returns a closed walk from b_i to itself. □

Lemma 5.76. $\textsc{Walk}(T, B')$ *(Algorithm 4) terminates, is well-defined, and returns a closed walk in H of length q, where $q \geq 3$ and $q \neq 4$.*

Proof. Since Algorithm 3 terminates (Lemma 5.75), it is immediate that Algorithm 4 terminates. If there is no exit of (T, B') then $\textsc{Walk}(T, B')$ returns $C(B', \Gamma_T(B'))$ which is a cycle in $\mathrm{Cy}(B')$ by Definition 5.58 and hence has length at least 3, but not 4, by Definition 5.53. If there is an exit a^* of (T, B') then, by Lemma 5.74 all subroutine calls have feasible inputs. By Lemma 5.75, $\textsc{ExitWalk}(T, a^*, B', 1, a^*)$ returns a closed walk from a^* to itself, of length at least 3, and $\textsc{ExitWalk}(T, a^*, B^*, r^* + 1, b^*)$ returns a closed walk from b^* to itself, also of length at least 3. It follows that the concatenations in the if-block are well-defined and therefore that Algorithm 4 is well-defined. Furthermore, $\textsc{ExitWalk}(T, a^*, B', 1, a^*) + \mathrm{P}_H(a^*, b^*) + \textsc{ExitWalk}(T, a^*, B^*, r^* + 1, b^*) + \mathrm{P}_H(b^*, a^*)$ is a closed walk from a^* to itself of length $q \geq 6$. □

Observation 5.77. $\textsc{Walk}(T, B')$ *(Algorithm 4) outputs a closed walk W. If (T, B') has no exit then $W = C$ for a cycle $C \in Cy(B')$. Otherwise, the following holds. For a positive integer j, there are obstructions $B'_0, \dots, B'_j$ such that W is of the form $W = Q_0 + P_0 + Q_1 + P_1 \cdots + Q_j + P_j$ where Q_i and P_i satisfy the following properties for all $i \in \{0, \dots, j\}$.*

- *Let $C_i = C(B'_i, \Gamma_T(B'_i))$. Then there are vertices a and a' in B'_i such that Q_i is of the form $\mathrm{W}_{C_i}(a)$ or $\mathrm{W}_{C_i}(a, a')$. Either way, all vertices of Q_i are in B'_i. Furthermore, only the endpoints of Q_i are exits of (T, B'_i).*
- *There is an exit a of (T, B'_i) with a destination $(b, B'_{i+1 \bmod (j+1)})$ such that P_i is a path of the form $\mathrm{P}_H(a, b)$. Hence, by Definition 5.71 and Lemma 5.70, the endpoints of P_i are the only vertices of P_i that are part of an obstruction.*
- *The obstruction B'_i is distinct from the obstruction $B'_{i+1 \bmod (j+1)}$.*

Explanation of Observation 5.77. If (T, B') has no exit then the result follows directly from the definition of $C(B, A)$ for a suitable connector (B, A) of an obstruction B (Definition 5.58).

For the remaining case, we will prove the following about Algorithm 3. ExitWalk(T, a^*, B, ℓ, a_0) outputs a closed walk W'. For a positive integer q, there are obstructions $B''_0, \ldots, B''_q$ such that W' is of the form $W' = Q'_0 + \sum_{i=1}^{q}(P'_i + Q'_i)$ where Q'_i and P'_i satisfy the following properties for all $i \in \{0, \ldots, q\}$.

- Let $C_i = C(B''_i, \Gamma_T(B''_i))$. Then there are vertices a and a' in B''_i such that Q'_i is of the form $\mathrm{W}_{C_i}(a)$ or $\mathrm{W}_{C_i}(a, a')$. Either way, all vertices of Q'_i are in B''_i. Furthermore, only the endpoints of Q'_i are exits of (T, B''_i).
- There is an exit a of (T, B''_i) with a destination $(b, B''_{i+1 \bmod (j+1)})$ such that P'_i is a path of the form $\mathrm{P}_H(a, b)$. Hence, by Definition 5.71, the endpoints of P'_i are the only vertices of P'_i that are part of an obstruction.
- The obstruction B''_i is distinct from the obstruction $B''_{i+1 \bmod (q+1)}$.

The proof is by induction on the recursion depth. If ExitWalk(T, a^*, B, ℓ, a_0) makes no recursive calls then the variable "k" is equal to 0 and we also set $q = 0$. In this case, B''_0 is equal to B, and the first property follows easily (the others are vacuous). Otherwise, k is positive and (T, B) has exits $\{a_0, \ldots, a_k\}$ where $a_1, \ldots, a_k$ have destinations $(b_1, B_1), \ldots, (b_k, B_k)$. Note that $B_1, \ldots, B_k$ are disjoint from B and from each other. Once again, B''_0 is B. Q'_0 is $W_C(a_0, a_1)$, as defined in the algorithm. Then B''_1 is B_1, and P'_1 is $P_H(a_1, b_1)$ as defined in the algorithm. The rest follows by induction, and examination of the algorithm, using the fact that the block-cut tree is a tree.

Given this fact for Algorithm 3, we obtain the conclusion for Algorithm 4 by putting together the pieces in the output W. This completes our explanation of Observation 5.77.

Lemma 5.78. *Let H be a connected graph. Let T be a closed suitable subtree of $\mathsf{BC}(H)$. Let B' be an obstruction that is a block of T. Let $W = (w_0, , \ldots, w_{q-1}, w_0)$ be the output of* Walk(T, B') *(Algorithm 4). Then, for each $i \in \{0, \ldots, q-1\}$, w_i and $w_{i+2 \bmod q}$ are distinct.*

Proof. All indices in this proof are considered to be modulo q. For any $i \in \{0, \ldots, q-1\}$, our goal is to show $w_i \neq w_{i+2}$. We make a case distinction based on Observation 5.77.

- If W is a cycle $C \in \mathrm{Cy}(B')$ then $w_i \neq w_{i+2}$ is immediate.
- Otherwise, for a positive integer j, W is of the form $W = Q_0 + P_0 + Q_1 + P_1 \cdots + Q_j + P_j$ with the properties stated in Observation 5.77. We consider the walk (w_i, w_{i+1}, w_{i+2}).
  - If for some $\ell \in [j]$, (w_i, w_{i+1}, w_{i+2}) is a subwalk of Q_ℓ then $w_i \neq w_{i+2}$ since, by Observation 5.77, Q_i is a subwalk of a cycle.
  - If for some $\ell \in [j]$, (w_i, w_{i+1}, w_{i+2}) is a subwalk of P_ℓ then, since P_ℓ is a path, we have $w_i \neq w_{i+2}$.
  - Otherwise, by Observation 5.77, there is no biconnected component that contains both w_i and w_{i+2} and consequently $w_i \neq w_{i+2}$.

□

The following lemma establishes (a stronger version of) the properties (L5.12.2) and (L5.12.3) for the walk returned by Algorithm 4, as required by Lemma 5.13.

Lemma 5.79. *Let H be a connected K_4-minor-free graph. Let T be a closed suitable subtree of $\mathsf{BC}(H)$. Let B' be an obstruction that is a block of T. Let $W = (w_0, \ldots, w_{q-1}, w_0)$ be the output of* Walk(T, B') *(Algorithm 4). By Lemma 5.76, W is a closed walk and $q \geq 3$. For each $i \in \{0, \ldots, q-1\}$, let $W_i = N_{W,H}(w_i)$ (Definition 5.2). If H has no hardness gadget then the following statement holds:*

$$\text{If } u \in W_{i-1 \bmod q} \text{ and } v \in W_{i+1 \bmod q} \text{ then } \Gamma_H(u) \cap \Gamma_H(v) = W_i.$$

Proof. All indices in this proof are considered to be modulo q. Let $i \in \{0, \ldots, q-1\}$, $u \in W_{i-1}$ and $v \in W_{i+1}$. Our goal is to show that $\Gamma_H(u) \cap \Gamma_H(v) = W_i$. We split the proof into two cases (Claims A and B).

Claim A: *If there is no biconnected component of H that contains both w_{i-1} and w_{i+1} then $\Gamma_H(u) \cap \Gamma_H(v) = W_i$.*

Proof: If there is no biconnected component that contains both w_{i-1} and w_{i+1} then, by the definition of W_i, $W_i = \{w_i\}$. Since w_{i-1} and w_{i+1} are not in the same biconnected component every path from w_{i-1} to w_{i+1} goes through w_i. There is a path from w_{i-1} to u via w_{i-2} and there is a path from v to w_{i+1} via w_{i+2}. Since w_{i-2} and w_{i+2} are distinct from w_i by Lemma 5.78 these paths do not go through w_i. Hence every path from u to v also goes through w_i. Thus, there is no biconnected component that contains both u and v. Hence, $\Gamma_H(u) \cap \Gamma_H(v) = \{w_i\} = W_i$, as required. This concludes the proof of Claim A. ■

Claim B: *If there is a biconnected component B such that w_{i-1} and w_{i+1} are in B then $\Gamma_H(u) \cap \Gamma_H(v) = W_i$.*

Proof: By Lemma 5.78, $w_{i-1} \neq w_{i+1}$. This together with the fact that w_i is adjacent to both w_{i-1} and w_{i+1} implies that w_i is also in B. If $u = w_{i-1}$ then it is trivial that u is in B. If $u \neq w_{i-1}$ then $|W_{i-1}| > 1$. By the fact that $W_{i-1} = \Gamma_H(w_{i-2}) \cap \Gamma_H(w_i)$ and the fact that both w_{i-1} and w_i are in B, it follows that $W_{i-1} \subseteq V(B)$ and that w_{i-2} is in B. Thus, we have established that u is in B. Analogously, v is in B. We state this formally so we can refer to it.

Fact 1: *If $|W_{i-1}| > 1$ then every vertex in $W_{i-1} \cup \{w_{i-2}\}$ is in B. Similarly, if $|W_{i+1}| > 1$ then every vertex in $W_{i+1} \cup \{w_{i+2}\}$ is in B. Consequently, both u and v are in B.*

If $u = w_{i-1}$ and $v = w_{i+1}$ then $\Gamma_H(u) \cap \Gamma_H(v) = \Gamma_H(w_{i-1}) \cap \Gamma_H(w_{i+1}) = W_i$, as required.

Therefore, we assume for the rest of the proof that $u \neq w_{i-1}$ (the case $v \neq w_{i+1}$ is symmetric). By Fact 1, the walk $(w_{i-2}, w_{i-1}, w_i, w_{i+1})$ is in B. We show that B is an obstruction. Suppose, for contradiction, that B is an edge, diamond or impasse. Then by Observation 5.77, there are cut-vertices a and b such that the walk $(w_{i-2}, w_{i-1}, w_i, w_{i+1})$ is a subpath of $\mathrm{P}_H(a, b)$. This contradicts Lemma 5.70, which states that no four consecutive vertices of this path are part of the same biconnected component.

Thus, we have established that $(w_{i-2}, w_{i-1}, w_i, w_{i+1})$ is a walk in the obstruction B. By Observation 5.77 and the definition of $\mathrm{W}_C(\cdot)$ (Definition 5.73), it is a subwalk of some cycle $C \in \mathrm{Cy}(B)$ following the order $D(C)$. It follows that $W_{i-1} = N_{C,H}(w_{i-1})$ and $W_i = N_{C,H}(w_i)$. By Corollary 5.52, from the fact that H has no hardness gadget and $|W_{i-1}| > 1$ it follows that $W_i = \{w_i\}$. Let ℓ be the length of C. Since $C \in \mathrm{Cy}(B)$ we have $\ell = 3$ or $\ell > 4$. We make a case distinction depending on ℓ.

- Suppose $\ell = 3$ so $w_{i-2} = w_{i+1}$. Suppose, for contradiction that $|W_{i+1}| > 1$. Then by Fact 1, w_{i+2} is also in B and $(w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2})$ is a subwalk of $\mathrm{W}_C(\cdot)$. This gives a contradiction to the fact that all vertices of $\mathrm{W}_C(\cdot)$, apart from possibly its endpoints, are distinct (see Definition 5.73). Therefore, $W_{i+1} = \{w_{i+1}\}$ and consequently $v = w_{i+1}$. Since $u \neq w_{i-1}$, (v, u, w_i, v) and (v, w_{i-1}, w_i, v) are two distinct triangles that share the edge $\{w_i, v\}$. By Lemma 5.24, since H has no hardness gadget, u and v have no common neighbour other than w_i.
- Suppose $\ell > 4$. Apply Lemma 5.50 to the cycle C and the index $i - 1$. This shows that there is a separation (A_1, A_2) of H such that $C \setminus \{w_{i-1}\} \subseteq A_1$, $W_{i-1} = N_{C,H}(w_{i-1}) \subseteq A_2$, and $A_1 \cap A_2 = \{w_{i-2}, w_i\}$. Since $u \in A_2$, u is not adjacent to any vertex in $C \setminus \{w_{i-2}, w_{i-1}, w_i\}$. By the definition of $\mathrm{Cy}(B)$ (Definition 5.53) C is an induced cycle of B, so the cycle C' that is obtained from C by replacing w_{i-1} with u is also an induced cycle of B. Also, C' has length $\ell > 4$. Since H has no hardness gadget, we can apply Corollary 5.52 to obtain (using the fact that $N_{C',H}(u) = N_{C,H}(w_{i-1}) = W_{i-1}$ has cardinality greater than 1) that $|N_{C',H}(w_i)| = 1$. By definition, $N_{C',H}(w_i) = \Gamma_H(u) \cap \Gamma_H(w_{i+1})$, so $\Gamma_H(u) \cap \Gamma_H(w_{i+1}) = \{w_i\}$.
  - If $|W_{i+1}| = 1$ then $v = w_{i+1}$, so we are finished.
  - Suppose that $|W_{i+1}| > 1$. By Fact 1, w_{i+1} and w_{i+2} are in B, and consequently the walk $(w_{i-2}, u, w_i, w_{i+1}, w_{i+2})$ is a subwalk of C'. It follows that $W_{i+1} = N_{C',H}(w_{i+1})$. Apply Lemma 5.50 to the cycle C' and the index $i + 1$. This shows that there is a separation (A_3, A_4) of H such that $C' \setminus \{w_{i+1}\} \subseteq A_3$, $W_{i+1} \subseteq A_4$, and $A_3 \cap A_4 = \{w_i, w_{i+2}\}$.
    Since $v \in A_4$, v is not adjacent to any vertex in $C' \setminus \{w_i, w_{i+1}, w_{i+2}\}$. So the cycle C'' that is obtained from C' by replacing w_{i+1} with v is also an induced cycle of B with length $\ell > 4$. Since H has no hardness gadget, we can apply Corollary 5.52 to obtain (using the fact that $N_{C'',H}(v) = W_{i+1}$ has cardinality greater than 1) that $|N_{C'',H}(w_i)| = 1$. By definition, $N_{C'',H}(w_i) = \Gamma_H(u) \cap \Gamma_H(v)$, so we are finished.

This concludes the proof of Claim B. ■

The lemma follows immediately from Claim A and Claim B. □

The following lemma establishes property (L5.12.4) for the walk returned by Algorithm 4, as required by Lemma 5.13.

Lemma 5.80. *Let H be a connected graph. Let T be a closed suitable subtree of $\mathsf{BC}(H)$. Let B' be an obstruction that is a block of T. Let $W = (w_0, , \dots, w_{q-1}, w_0)$ be the output of* Walk(T, B') *(Algorithm 4). By Lemma 5.76, W is a closed walk and $q \geq 3$. For each $i \in \{0, \dots, q-1\}$, let $W_i = N_{W,H}(w_i)$. Then there exists no closed walk $D = (d_0, \dots, d_{q-1}, d_0)$ with $d_i \in \Gamma_H(W_i) \setminus (W_{i-1} \cup W_{i+1})$ for all i (indices taken modulo q).*

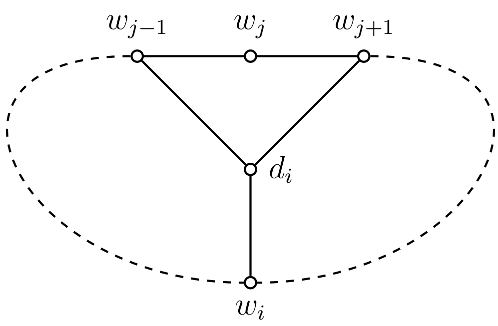
Proof. Assume for the sake of contradiction that such a walk D exists. We distinguish two cases:

(I) W is not entirely contained in a single biconnected component of H. In this case, there is an index i such that no biconnected component contains both w_{i-1} and w_{i+1}. Now consider $d_{i-1} \in \Gamma_H(w_{i-1})$ and $d_{i+1} \in \Gamma_H(w_{i+1})$. Note that $d_{i-1} \neq w_i$ and $d_{i+1} \neq w_i$ by the specification of D. Note further that $d_i \neq w_i$ as we do not allow self-loops in H. Consequently, there are two internally vertex disjoint 2-paths from w_{i-1} to w_{i+1}; one passes through d_{i-1}, d_i and d_{i+1}; and the other passes through w_i. This is a contradiction to the fact that no biconnected component contains both w_{i-1} and w_{i+1}.

(II) W is entirely contained in a biconnected component B. By Observation 5.77, the only possibility for this to be true is that B is an obstruction and W is a cycle in $\mathrm{Cy}(B)$. By the definition of obstructions (and $\mathrm{Cy}(B)$), W is thus an induced cycle of length q such that $q \geq 3$ and $q \neq 4$. The lemma will follow easily from the following claim.

Claim A: $D \cap (\bigcup_i W_i) = \emptyset$.

Proof: Assume for contradiction that $d_i \in W_j$ for some indices i and j. Note that $j \notin \{i-1, i+1\}$ by the specification of D. We cannot have $j = i$ since $d_i \in \Gamma_H(W_i)$ so $d_i \notin W_i$ (otherwise H has a self-loop). Since $j \notin \{i-1, i, i+1\}$, we have $q \geq 5$. We will show that H has a K_4-minor. Since d_i is adjacent to w_i, and it is not equal to w_{i-1} or w_{i+1} and since W is an induced cycle in B, we conclude that d_i is distinct from the vertices of W. H therefore contains a K_4-minor containing the vertex d_i and its three neighbours w_i, w_{j-1} and w_{j+1}. There are disjoint paths between these three vertices along the cycle W and d_i is not on these paths. See the following illustration.



This concludes the proof of Claim A. ■

We have assumed for contradiction that D exists, and proved Claim A. We obtain the contradiction by using Claim A to construct a K_4-minor in H. Claim A demonstrates that $W \cap D = \emptyset$. Now contract the walk D to a single vertex. This yields a vertex $d \notin W$ which is adjacent to all vertices of W. As W has length at least 3, we have found a K_4-minor as promised.

□

Lemma 5.81. *Let H be a connected K_4-minor-free graph. Let T be a closed suitable subtree of $\mathsf{BC}(H)$. Let B' be an obstruction that is a block of T. Then H has a hardness gadget.*

Proof. Let $W = (w_0, \ldots, w_{q-1}, w_0)$ be the output of $\textsc{Walk}(T, B')$. By Lemma 5.76, W is a closed walk with $q \geq 3$ and $q \neq 4$. Our goal is to use Lemma 5.13 to show that H has a hardness gadget. To this end, we identify the sets $\mathcal{C}_i$ of Lemma 5.13 with the sets $W_i = N_{W,H}(w_i)$. Let S be the set of all i such that W_i has even cardinality.

Claim A: *For every $i \in S$, there is an obstruction O_i such that, for $C_i = C(O_i, \Gamma_T(O_i))$, the following hold.*

- $w_{i-1}, w_i, w_{i+1} \in C_i$.
- $W_i = N_{C_i,H}(w_i)$.
- *Every vertex in W_i has degree 2 in O_i.*
- *w_i is an attachment point of (T, O_i).*

Proof: Fix $i \in S$. By the definition of W_i and the fact that $|W_i| > 1$, there is a biconnected component O_i of H that contains w_{i-1}, W_i, and w_{i+1}. Suppose, for contradiction, that O_i is an edge, diamond or impasse, then, by Observation 5.77, the walk (w_{i-1}, w_i, w_{i+1}) is a subpath of a path of the form $\mathrm{P}_H(\cdot)$. However, since $|W_i| \geq 2$, w_{i-1} and w_{i+1} have at least 2 common neighbours in H, this is a contradiction to Lemma 5.70.

We have established that O_i is an obstruction. Since T is a suitable subtree, $(O_i, \Gamma_T(O_i))$ is a suitable connector and, by Definition 5.58, $C_i = C(O_i, \Gamma_T(O_i))$ is a cycle with $\Gamma_T(O_i) = \{c \in C_i \mid \text{the cardinality of } N_{C_i,H}(c) \text{ is even}\}$. By Observation 5.77, (w_{i-1}, w_i, w_{i+1}) is a subwalk of a walk of the form $\mathrm{W}_{C_i}(\cdot)$. It follows that $w_{i-1}, w_i, w_{i+1} \in C_i$ and $W_i = N_{C_i,H}(w_i)$, as required.

As the cardinality of W_i is even and $C_i \in \mathrm{Cy}(O_i)$, by Definition 5.53, every vertex in W_i has degree 2 in O_i, as required.

The fact that the cardinality of W_i is even also implies that $w_i \in \Gamma_T(O_i)$ (since $\Gamma_T(O_i) = \{c \in C_i \mid \text{the cardinality of } N_{C_i,H}(c) \text{ is even}\}$). Thus, by Definition 5.71, w_i is either an exit or an attachment point of (T, O_i). However, by Observation 5.77, only the endpoints of $\mathrm{W}_{C_i}(\cdot)$ are exits, which means that w_i is an attachment point, as required. This finishes the proof of Claim A. ■

In the remainder of this proof, for each $i \in S$, let O_i and C_i be as stated in Claim A. Next we use the fact that, for each $i \in S$, w_i is an attachment point of

(T, O_i) to define a gadget $(\hat{J}_i, \hat{z}_i)$. Those gadgets will be used in the construction of the gadgets $(J_0, z_0), \dots, (J_{q-1}, z_{q-1})$ required by Lemma 5.13. Recall that, by definition of attachment points (Definition 5.71), for each $i \in S$, there is a (unique) maximal-length proper obstruction-free path $P_i = (b^i_0, B^i_1, b^i_1, B^i_2, \dots, B^i_{q_i}, b^i_{q_i})$ in T such that $w_i = b^i_{q_i}$ and b^i_0 is a leaf in T. As T is closed, we obtain that, for some block R of T, the vertex b^i_0 is R-closed, i.e., b^i_0 has precisely one descendant in $\mathsf{BC}(H)$ with respect to $<_R$. Moreover, this descendant must be an edge. We distinguish whether $q_i = 0$ or $q_i \geq 1$:

$q_i = 0$: We have $w_i = b^i_{q_i} = b^i_0$. Since $b^i_0 = w_i$ is R-closed, Lemma 5.64 ensures that it is also O_i-closed. Consequently, w_i has precisely three neighbours in H: The two neighbours in O_i (which are w_{i-1} and w_{i+1} — these are distinct by Lemma 5.78), as well as the other endpoint ℓ_i of the edge that is the unique descendant of w_i in $\mathsf{BC}(H)$.

We define $\hat{J}_i$ to be a single edge, one endpoint of which is z_i, and the other endpoint of which is pinned to w_i. Observe that

$$\{v \in V(H) \mid \left|\hom\Big((\hat{J}_i, \hat{z}_i) \to (H, v)\Big)\right| \text{ is odd.}\} = \{w_{i-1}, w_{i+1}, \ell_i\}\,.$$

This concludes the definition of $(\hat{J}_i, \hat{z}_i)$ in the case that $q_i = 0$.

$q_i \geq 1$: By Lemma 5.64, b^i_0 is B^i_0-closed. It follows that $|\Gamma_H(b^i_0) \setminus B^i_0| = 1$. Since T is a suitable subtree and P_i is obstruction-free, for each $j \in [q_i]$, $(B^i_j, \{b^i_{j-1}, b^i_j\})$ is a suitable connector in H and B^i_j is an edge, diamond or impasse. Thus, we can invoke Lemma 5.47. We obtain that at least one of the following is true:

- H has a hardness gadget.
- $B^i_{q_i}$ is an edge or a diamond and $(L_i, b^i_{q_i})$ is a good start in $B^i_{q_i}$ but not a good stop in $B^i_{q_i}$, where $L_i = \{b^i_{q_i-1}\}$.
- $B^i_{q_i}$ is an impasse, and $(L_i, b^i_{q_i})$ is a good start in $B^i_{q_i}$ but not a good stop in $B^i_{q_i}$, where $L_i = \{d_i\}$ and d_i is the unique common neighbour of $b^i_{q_i-1}$ and $b^i_{q_i}$ in H.

We are done in the first case, so suppose that one of other cases applies. By definition of good starts, we thus obtain a gadget $(\hat{J}_i, \hat{z}_i)$ such that, for $R_i = \Gamma_H(b^i_{q_i}) \setminus V(B^i_{q_i})$,

$$\{v \in V(H) \mid \left|\hom\Big((\hat{J}_i, \hat{z}_i) \to (H, v)\Big)\right| \text{ is odd.}\} = L_i \cup R_i.$$

Note that L_i and R_i are disjoint. Further, recall that $w_i = b^i_{q_i}$ and therefore $R_i \cap V(O_i) = \{w_{i-1}, w_{i+1}\}$ (by Claim A). As $(L_i, b^i_{q_i})$ is not a good stop in $B^i_{q_i}$, we have that R_i is of even cardinality, and thus $L_i \cup R_i$ is of odd cardinality. This concludes the definition of $(\hat{J}_i, \hat{z}_i)$ in the case that $q_i \geq 1$.

We now state the previously-established crucial property of the gadgets $(\hat{J}_i, \hat{z}_i)$ (which applies for all q_i, unless H has a hardness gadget).

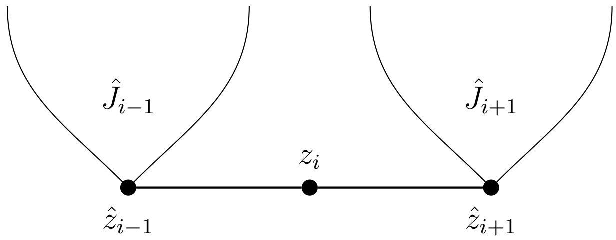


Figure 5.11: The gadget (J_i, z_i) for the case $i-1 \in S$ and $i+1 \in S$

Fact 1: *For every $i \in S$, there is a gadget $(\hat{J}_i, \hat{z}_i)$ such that the set*

$$\hat{\Omega}_i = \{v \in V(H) \mid \quad \left|\hom\Big((\hat{J}_i, \hat{z}_i) \to (H, v)\Big)\right| \text{ is odd.}\}$$

is a subset of $\Gamma_H(w_i)$, has odd cardinality, and contains precisely two vertices of O_i — the vertices w_{i-1} and w_{i+1}.

We proceed by defining for each $i \in \{0, \dots, q-1\}$ the gadgets (J_i, z_i) needed for Lemma 5.13. The definition of (J_i, z_i) depends on whether or not $i-1$ and $i+1$ are in S. (J_i, z_i) always contains the vertex z_i. Additionally, for $j \in \{i-1, i+1\}$, if $j \in S$ then (J_i, z_i) also contains a copy of the gadget $(\hat{J}_j, \hat{z}_j)$ and z_i is adjacent to $\hat{z}_j$. We provide an illustration of (J_i, z_i) for the case $i-1 \in S$ and $i+1 \in S$ in Figure 5.11.

Following the notation of Lemma 5.13, we set (for every $i \in \{0, \dots, q-1\}$):

$$\Omega_i = \{v \in V(H) \mid \quad |\hom((J_i, z_i) \to (H, v))| \text{ is odd.}\}$$

Claim B: *For all $i \in \{0, \dots, q-1\}$ the following holds true, unless H has a hardness gadget; indices are taken modulo q:*

- *If $i-1 \in S$ then $W_{i-1} \cap \Omega_i = \{w_{i-1}\}$, and if $i-1 \notin S$ then $W_{i-1} \cap \Omega_i = W_{i-1}$.*
- *If $i+1 \in S$ then $W_{i+1} \cap \Omega_i = \{w_{i+1}\}$, and if $i+1 \notin S$ then $W_{i+1} \cap \Omega_i = W_{i+1}$.*

So, $W_{i-1} \cap \Omega_i$ and $W_{i+1} \cap \Omega_i$ have odd cardinality.

Proof: We only show the first item; the second one is symmetric. We distinguish whether or not $i-1 \in S$:

(I) If $i-1 \in S$ then, by Claim A, w_{i-1} and w_i are contained in the cycle C_{i-1}, and $W_{i-1} = N_{C_{i-1},H}(w_{i-1})$. Since $C_{i-1} \in \mathrm{Cy}(O_{i-1})$, C_{i-1} is an induced cycle in O_{i-1} (hence in H) and is not a square. Therefore we can apply Corollary 5.52. It follows that H either has a hardness gadget, in which case we are done, or $\left|N_{C_{i-1},H}(w_i)\right| = 1$, i.e., $N_{C_{i-1},H}(w_i) = \{w_i\}$. This implies that w_i is not an exit of (T, O_{i-1}), and thus $w_{i+1} \in C_{i-1}$ and consequently $W_i = N_{C_{i-1},H}(w_i) = \{w_i\}$.

We are now able to prove that $W_{i-1} \cap \Omega_i = \{w_{i-1}\}$. First, for $v \in W_{i-1} \setminus \{w_{i-1}\}$, we show that $v \notin \Omega_i$ and hence $v \notin W_{i-1} \cap \Omega_i$. By the construction of J_i, it suffices to show that there is an even number of vertices in $\hat{\Omega}_{i-1}$ that are adjacent to v. Recall from Fact 1 that $\hat{\Omega}_{i-1} \subseteq \Gamma_H(w_{i-1})$. The vertex v has precisely two common neighbours with w_{i-1}, namely w_{i-2} and w_i (any others would lead to a K_4-minor in H induced by the vertices $\{v, w_{i-2}, w_{i-1}, w_i\}$). By Fact 1, we know that both of these are in $\hat{\Omega}_{i-1}$ and hence that there are two vertices in $\hat{\Omega}_{i-1}$ that are adjacent to v, as required.

It remains to show that $w_{i-1} \in \Omega_i$ and hence $w_{i-1} \in W_{i-1} \cap \Omega_i$. By the construction of J_i, it suffices to show that there is an odd number of vertices in $\hat{\Omega}_{i-1}$ that are adjacent to w_{i-1}, and, in case $i+1 \in S$, that there is an odd number of vertices in $\hat{\Omega}_{i+1}$ that are adjacent to w_{i-1}.

- By Fact 1, $\hat{\Omega}_{i-1} \subseteq \Gamma_H(w_{i-1})$. Hence every element of $\hat{\Omega}_{i-1}$ is adjacent to w_{i-1}. By Fact 1, $\hat{\Omega}_{i-1}$ has odd cardinality, as required.
- Suppose that $i+1 \in S$. By Fact 1, we have $\hat{\Omega}_{i+1} \subseteq \Gamma_H(w_{i+1})$ and $w_i \in \hat{\Omega}_{i+1}$. Furthermore, w_i is the only common neighbour of w_{i-1} and w_{i+1} in H by the fact that $W_i = \{w_i\}$. Hence w_i is the only vertex in $\hat{\Omega}_{i+1}$ that is adjacent to w_{i-1}, as required.

(II) Consider $i-1 \notin S$. Our goal is to show that $W_{i-1} \cap \Omega_i = W_{i-1}$. If $i+1 \notin S$, then J_i contains only the single vertex z_i and $\Omega_i = V(H)$ and we are finished.

Hence we can assume $i+1 \in S$. We first proceed as in Case (I) to obtain either a hardness gadget or $W_i = \{w_i\}$. By Claim A, w_i, w_{i+1} and w_{i+2} are contained in the cycle C_{i+1}, and $W_{i+1} = N_{C_{i+1},H}(w_{i+1})$. Since $C_{i+1} \in \mathrm{Cy}(O_{i+1})$, C_{i+1} is induced and not a square and therefore we can apply Corollary 5.52. It follows that H either has a hardness gadget, in which case we are done, or $\left|N_{C_{i+1},H}(w_i)\right| = 1$, i.e., $N_{C_{i+1},H}(w_i) = \{w_i\}$. This implies that w_i is not an exit of (T, O_{i+1}), and thus $w_{i-1} \in C_{i+1}$ and consequently $W_i = N_{C_{i+1},H}(w_i) = \{w_i\}$.

In order to show that $W_{i-1} \cap \Omega_i = W_{i-1}$ we show, for each $v \in W_{i-1}$, that $v \in \Omega_i$. By the construction of J_i ($i-1 \notin S$, $i+1 \in S$), it suffices to show that there is an odd number of vertices in $\hat{\Omega}_{i+1}$ that are adjacent to v. Recall from Fact 1 that $\hat{\Omega}_{i+1} \subseteq \Gamma_H(w_{i+1})$. By the fact that $v \in W_{i-1}$ and $w_{i+1} \in W_{i+1}$, from Lemma 5.79 we obtain that $\Gamma_H(v) \cap \Gamma_H(w_{i+1}) = W_i$. We have already established that $W_i = \{w_i\}$ and hence w_i is the only vertex in $\hat{\Omega}_{i+1}$ that is adjacent to w_{i-1}, as required.

This concludes the proof of Case (II) and of Claim B. ■
We prove one final claim before we can apply Lemma 5.13:

Claim C: *Unless H has a hardness gadget, there exists $k \in \{0, \dots, q-1\}$ such that both of the following are true; indices are taken modulo q:*

- *There are no edges between W_k and W_{k+3}.*
- *$(W_k \cup W_{k+2}) \cap \Omega_{k+1}$ and $(W_{k+1} \cup W_{k+3}) \cap \Omega_{k+2}$ are of even cardinality.*

Proof: We distinguish two cases.

(I) There is a biconnected component B that contains W. Consequently, by Observation 5.77, there is a cycle $C \in \mathrm{Cy}(B)$ such that $W = C$. Since $C \in \mathrm{Cy}(B)$ it has length $q = 3$ or $q \geq 5$. In this case, we choose $k = 0$. We first show that there is no edge between W_k and W_{k+3}:

- If $q = 3$, we show that for $u, v \in W_0$ there cannot be an edge between u and v. If $u = v$ there cannot be an edge since we do not allow self-loops in H. If $u \neq v$ there cannot be an edge, as otherwise u, v, w_1, w_2 induce a K_4-minor in H, contradicting the fact that H has none. Thus, there are no edges between W_0 and $W_{3 \bmod q} = W_0$.
- If $q \geq 5$, consider W_0 and $W_3 = W_{3 \bmod q}$. If $|W_0| = |W_3| = 1$ then there are no edges between W_0 and W_3 since C is induced by the definition of obstruction (Definition 5.53). So suppose $|W_0| > 1$ (the case $|W_3| > 1$ is symmetric). Since C is an induced cycle of length $q > 4$ in a biconnected graph B, we can apply Lemma 5.50 to find a separation (A, A') of H such that $C \setminus \{w_0\} \subseteq A$, $W_0 \subseteq A'$and $A \cap A' = \{w_q, w_1\}$. Since all of the vertices in $\bigcup_{i=1}^{q-1} W_i$ have neighbours in $C \setminus \{w_0\}$, this implies that w_{q-1} and w_1 are the only vertices in $\bigcup_{i=1}^{q-1} W_i$ that are adjacent to vertices in W_0. However, by Lemma 5.49, $W_0, \ldots W_{q-1}$ are pairwise disjoint and hence $w_{q-1}, w_1 \notin W_3$. So, there are no edges between W_0 and W_3, as required.

To establish the second bullet point, again use $k = 0$ and the fact that $W_0, \ldots W_{q-1}$ are pairwise disjoint. We have $|(W_0 \cup W_2) \cap \Omega_1| = |W_0 \cap \Omega_1| + |W_2 \cap \Omega_1|$. By Claim B, each of these terms is odd, so their sum is even. The same argument applies to $(W_1 \cup W_3) \cap \Omega_2$.

(II) W is not entirely contained in one biconnected component. If this is true, then by Observation 5.77, there exists an obstruction B with cycle $C \in \mathrm{Cy}(B)$ such that, for some $k \in \{0, \ldots, q-1\}$, w_k and w_{k+1} are contained in C, w_{k+1} is an exit of B (in particular, an articulation point), and w_{k+2} and w_{k+3} are not contained in B.

Since $w_{k+2} \neq w_{k+4}$ by Lemma 5.78, it follows that no $v \in W_{k+3}$ is in B, which implies that there is no edge between W_k and W_{k+3}, as required.

For the second item, observe that W_k and W_{k+2} must be disjoint, as w_k and w_{k+1} are in the biconnected component B, but w_{k+3} is not. We further claim that W_{k+1} and W_{k+3} are disjoint. To see this, observe first that $W_{k+1} = \{w_{k+1}\}$ since w_{k+1} is the only common neighbour of w_k and w_{k+2} as otherwise w_{k+2} would be contained in B. Then we have already established that no $v \in W_{k+3}$ is in B, which implies $w_{k+1} \notin W_{k+3}$.

Using the fact that W_k and W_{k+2} are disjoint, we conclude that $|(W_k \cup W_{k+2}) \cap \Omega_{k+1}| = (|W_k \cap \Omega_{k+1}| + |W_{k+2} \cap \Omega_{k+1}|)$. By Claim B, each of these terms is odd, so their sum is even. Using the fact that W_{k+1} and W_{k+3} are disjoint, the same is true for W_{k+1} and W_{k+3}.

■

We are finally able to invoke Lemma 5.13: Recall first, that $q \geq 3$ and $q \neq 4$ from the beginning of the proof. Recall that we identify the sets $\mathcal{C}_i$ of Lemma 5.13 with the sets W_i. Unless H has a hardness gadget (in which case we are finished) the following hold.

(L5.12.1) holds by Claim B.

(L5.12.2) and (L5.12.3) hold by Lemma 5.79.

(L5.12.4) is established by Lemma 5.80.

There is a k such that (L5.13.1) and (L5.13.2) hold by Claim C.

Consequently, all conditions are satisfied and we obtain a hardness gadget by Lemma 5.13. □

5.6.5 Proof of the Main Theorem

We can now prove Theorem 1.27, which we restate for convenience.

Theorem 1.27. Let H be a simple graph whose involution-free reduction H^* is K_4-minor free. If H^* contains at most one vertex, then $\oplus\textsc{Hom}(H)$ can be solved in polynomial time. Otherwise, $\oplus\textsc{Hom}(H)$ is $\oplus$P-complete and, assuming the randomised Exponential Time Hypothesis, it cannot be solved in time $\exp(o(|V(G)| + |E(G)|))$.

Proof. By Theorem 1.24, for every graph G, $|\hom(G \to H)| = |\hom(G \to H^*)| \bmod 2$. It is trivial to count homomorphisms to a graph with at most one vertex. Suppose that H^* has at least two vertices. Then it suffices to show that $\oplus\textsc{Hom}(H^*)$ is $\oplus$P-complete and that $\oplus\textsc{Hom}(H^*)$ cannot be solved in time $\exp(o(|G|))$, unless the rETH fails.

Since H^* is involution-free and contains at least 2 vertices, there is an involution-free connected component H' of H^* with at least 2 vertices as well: If H is disconnected, it has at least 2 connected components, and at least one of those two components cannot be a single vertex, as otherwise, we obtain a non-trivial involution by switching those vertices. Furthermore, a connected component of an involution-free graph cannot have a non-trivial involution, as otherwise, the entire graph would have a non-trivial involution.

Next we claim that H' has a hardness gadget: Assume first that H' has a biconnected component that is not an edge, a diamond, an obstruction, or an impasse. By Lemma 5.57, H' has a hardness gadget. In the remaining case, every biconnected component of H' is an edge, a diamond, an obstruction, or an impasse. By Lemma 5.66, there is a closed suitable subtree T of the block-cut tree $\mathsf{BC}(H')$. If no block of T is an obstruction, then H' has a hardness gadget by Lemma 5.67. Otherwise, H' has a hardness gadget by Lemma 5.81.

This allows us to invoke Theorem 5.7 and we obtain that $\oplus\textsc{Ret}(H')$ is $\oplus$P-hard and cannot be solved in time $\exp(o|J|)$, unless the rETH fails.

Since H' is involution-free, we can reduce $\oplus\textsc{Ret}(H')$ to $\oplus\textsc{Hom}(H')$ by Theorem 5.4, and we can reduce $\oplus\textsc{Hom}(H')$ to $\oplus\textsc{Hom}(H^*)$ by Lemma 5.5. These reductions are

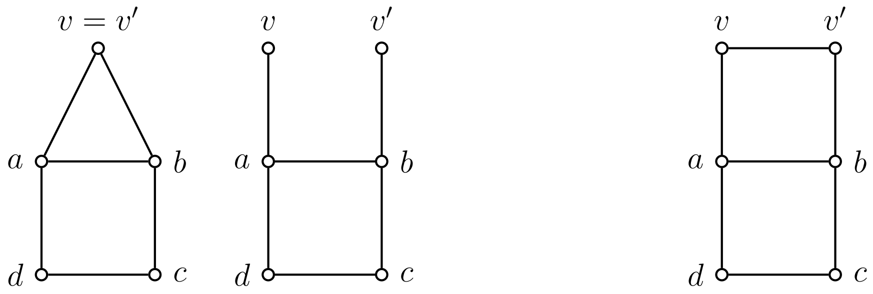


Figure 5.12: Illustration of the two cases in the proof of Lemma 5.82.

tight in the sense that any subexponential-time algorithm for $\oplus\text{Hom}(H^*)$ would yield a subexponential-time algorithm for $\oplus\text{Ret}(H')$; this is due to the fact that the size of the oracle queries in each reduction is bounded linearly in the input size. We thus obtain $\oplus$P-hardness of $\oplus\text{Hom}(H^*)$, and that $\oplus\text{Hom}(H^*)$ cannot be solved in time $\exp(o(|G|))$, unless the rETH fails. □

5.7 Counting Homomorphisms mod 2 to Graphs of Degree at most 3

We explored the possibilities for constructing hardness gadgets in graphs containing two squares that share one edge when we analysed K_4-minor-free and chordal bipartite graphs. It turns out that a similar strategy suffices to completely solve the case where H has degree at most 3. We start with the following lemma.

Lemma 5.82. *Let H be an involution-free graph of degree at most 3 that contains a square. Then H has a hardness gadget.*

Proof. Let $C = (a, b, c, d, a)$ be a square in H. Assume first that at least one of the edges $\{a, c\}$ or $\{b, d\}$ are present. W.l.o.g. let $\{a, c\}$ be present. Then a and c have degree 3 and thus, by assumption, no further neighbours. Thus (ac) is a non-trivial involution of H.

Now assume that none of $\{a, c\}$ or $\{b, d\}$ are edges of H. If both, a and c have degree 2 then (ac) is an involution. Similarly, if b and d have both degree 2, we obtain the involution (bd). W.l.o.g. we can thus assume that a and b have degree 3. Let v and v' be the neighbours of a and b, respectively, that are not contained in C. In what follows, we consider cases based on whether the edge $\{v, v'\}$ is present, and, if not, we differentiate between $v = v'$ and $v \neq v'$; an illustration is provided in Figure 5.12.

(I) $\{v, v'\} \notin E(H)$: This case corresponds to the two illustrations to the left of Figure 5.12. Note first that $\{v', d\}$ cannot be an edge of H, as otherwise, b and d both have neighbours $\{a, v', c\}$ (and no other neighbours, since they have degree 3), which means that (bd) is a non-trivial involution of H. Similarly, $\{v, c\}$ cannot be an edge of H, as otherwise (ac) is a non-trivial involution of H. Also, at least one of the edges $\{v, d\}$ and $\{v', c\}$ must not be present in H, as otherwise $(ad)(bc)$ is a non-trivial involution of H. W.l.o.g., assume that $\{v, d\}$ is not present. We construct a hardness gadget of H as follows:

- $I = \{a\}$.
- $S = \{b\}$.
- J_1 is a path of 4 vertices: The first vertex is a b-pin, the second vertex is y, and the fourth vertex is an a-pin.
- J_2 is a path of 3 vertices: The first vertex is an a-pin, the second vertex is z, and the third vertex is a c-pin.
- J_3 is just the edge $\{y, z\}$.

We first claim that $\Omega_y = \{v', a\}$. A vertex of H is in Ω_y if and only if it is adjacent to b and has an odd number of 2-paths to a. As H has degree at most three, the neighbours of b are precisely v', a and c. Note that a has degree precisely 3 and thus has an odd number of 2-paths to itself. Furthermore, there is only one 2-path from v' to a: This path contains b as internal vertex. There cannot be an additional 2-path from v' to a, since, in this case, the internal vertex must either be v, which is not possible as $\{v, v'\} \notin E(H)$, or d, which is not possible as $\{v, d\} \notin E(H)$. Finally, there are precisely two 2-paths from c to a: One has b as internal vertex, and the other has d as internal vertex. There cannot be a third one, as this 2-path would have v as internal vertex, but we ruled out the existence of the edge $\{v, c\}$. This shows that $\Omega_y = \{v', a\}$.

Our next claim is that $\Omega_z = \{b, d\}$. Observe that Ω_z contains precisely the common neighbours of a and c. Thus b and d are included in Ω_z. The only candidate for a third common neighbour would be v, but we ruled out the existence of the edge $\{v, c\}$.

Finally, we observe that $|\Sigma_{v',d}| = 0$ as $\{v', d\}$ is not an edge of H, and that $|\Sigma_{v',b}| = |\Sigma_{b,a}| = |\Sigma_{a,d}| = 1$.

(II) $\{v, v'\} \in E(H)$: This case corresponds to the illustration to the right of Figure 5.12. As in case (I), the edge $\{v, c\}$ is not present, as otherwise (ac) is a non-trivial involution, and that the edge $\{v', d\}$ is not present, as otherwise (bd) is a non-trivial involution. We construct a hardness gadget as follows:

- $I = \{a\}$.
- $S = \{b\}$.
- J_1 is a path of 3 vertices: The first vertex is a v-pin, the second vertex is y, and the third vertex is a b-pin.
- J_2 is a path of 3 vertices: The first vertex is an a-pin, the second vertex is z, and the third vertex is a c-pin.
- J_3 is just the edge $\{y, z\}$.

Note first that Ω_y contains precisely the common neighbours of v and b. Thus v' and a are contained in Ω_y. Recall further that c is not adjacent to v. As the degree of H is bounded by 3, we thus have $\Omega_y = \{v', a\}$. Similarly, we obtain that $\Omega_z = \{b, d\}$.

Finally, we observe that $|\Sigma_{v',d}| = 0$ as $\{v', d\}$ is not an edge of H, and that $|\Sigma_{v',b}| = |\Sigma_{b,a}| = |\Sigma_{a,d}| = 1$.

□

Theorem 5.83. *Let H be a graph whose involution-free reduction H^* has maximum degree at most 3. If H^* contains at most one vertex, then $\oplus\text{HOM}(H)$ can be solved in polynomial time. Otherwise, $\oplus\text{HOM}(H)$ is $\oplus$P-complete and, assuming the randomised Exponential Time Hypothesis, it cannot be solved in time $\exp(o(|G|))$.*

Proof. By Theorem 1.24, for every graph G, $|\text{hom}(G \to H)| = |\text{hom}(G \to H^*)| \bmod 2$. It is trivial to count homomorphisms to a graph with at most one vertex. Suppose that H^* has at least two vertices. Then it suffices to show that $\oplus\text{HOM}(H^*)$ is $\oplus$P-complete and that $\oplus\text{HOM}(H^*)$ cannot be solved in time $\exp(o(|G|))$, unless the rETH fails.

If H^* does not contain a square but has at least 2 vertices, then it has a hardness gadget as shown in [69]. If H^* contains a square, then it has a hardness gadget by Lemma 5.82.

By Theorem 5.7, we obtain that $\oplus\text{RET}(H^*)$ is $\oplus$P-hard and that it cannot be solved in time $\exp(o|J|)$, unless the rETH fails.

Finally, since H^* is involution-free, we can reduce $\oplus\text{RET}(H^*)$ to $\oplus\text{HOM}(H^*)$ by Theorem 5.4. As we have already noted, the size of the oracle queries in this reduction are bounded linearly in the input size, so the reduction proves that any subexponential-time algorithm for $\oplus\text{HOM}(H^*)$ would yield a subexponential-time algorithm for $\oplus\text{RET}(H^*)$, completing the proof. □

5.8 Counting List Homomorphisms modulo 2

Given graphs G and H together with a set of *lists* $\mathbf{S} = \{S_v \subseteq V(H) \mid v \in V(G)\}$, a *(list) homomorphism* from $(G, \mathbf{S})$ to H is a homomorphism h from G to H such that for each $v \in V(G)$ we have $h(v) \in S_v$. We use $\text{hom}((G, \mathbf{S}) \to H)$ to denote the set of homomorphisms from $(G, \mathbf{S})$ to H. List homomorphisms are a natural generalisation of both homomorphisms and retractions.

In this section we provide a complete complexity classification for the problem of counting list homomorphisms modulo 2 to a given graph H. The classification determines for which graphs H the problem is tractable. We strengthen the result by considering a wider class of graphs H than in the rest of this chapter (where we required H to be a simple graph, without self-loops or parallel edges). Let $\mathcal{H}$ be the set of all undirected graphs H which do not have parallel edges — self-loops are allowed.

Given a set S, let $\mathcal{P}(S)$ denote its power set. We consider the following problem, parameterised by a graph $H \in \mathcal{H}$ and by a set of lists $\mathcal{L} \subseteq \mathcal{P}(V(H))$.

Name: $\oplus\textsc{Hom}(H, \mathcal{L})$.
Input: A simple graph G and a collection of lists $\mathbf{S} = \{S_v \in \mathcal{L} \mid v \in V(G)\}$.
Output: $|\text{hom}((G, \mathbf{S}) \to H)| \bmod 2$.

The input G to $\oplus\textsc{Hom}(H, \mathcal{L})$ is assumed to be simple because this is standard in the field, and because it makes results stronger. However, this restriction is not important for our result — see Remark 5.87. Taking $\mathcal{L} = P(V(H))$, the problem $\oplus\textsc{Hom}(H, \mathcal{P}(V(H)))$ is the problem of counting list homomorphisms to H modulo 2. To simplify the notation, we also write $\oplus\textsc{LHom}(H)$ for this problem. The following lemma is well-known.

Lemma 5.84. *Let H be a graph in $\mathcal{H}$ that contains a walk (a, b, c, d) such that $a \neq c$, $b \neq d$, and $\{a, d\} \notin E(H)$. Let $\mathcal{L} \subseteq \mathcal{P}(V(H))$ be a set of lists with $\{\{a, c\}, \{b, d\}\} \subseteq \mathcal{L}$. Then $\oplus\textsc{Hom}(H, \mathcal{L})$ is $\oplus$P-complete.*

Proof. The problem $\oplus$BIS, of counting the independent sets of a bipartite graph, modulo 2, is known to be $\oplus$P-complete [46, Theorem 4.2.1]. We will reduce $\oplus$BIS to $\oplus\textsc{Hom}(H, \mathcal{L})$.

Let G be a bipartite graph (an input to $\oplus$BIS) with vertex partition $V(G) = (L, R)$. For each $v \in L$, let $S_v = \{a, c\}$ and for each $v \in R$ let $S_v = \{b, d\}$. We set $\mathbf{S} = \{S_v \mid v \in V(G)\}$. Then every homomorphism h from $(G, \mathbf{S})$ to H corresponds to an independent set in G (and vice versa), where $h(v) \in \{a, d\}$ means that v is *in* the independent set and $h(v) \in \{b, c\}$ means that v is *out* of the independent set. (Since $a \neq c$ and $b \neq d$ it is well-defined whether v is in or out.) Hence a single $\oplus\textsc{LHom}(H, \mathcal{L})$ oracle call with input $(G, \mathbf{S})$ returns the number of independent sets of G, modulo 2. □

Theorem 5.85. *Let H be a connected graph in $\mathcal{H}$ and let $\mathcal{L} \subseteq \mathcal{P}(V(H))$ be a set of lists with $\{S \subseteq V(H) \mid |S| = 2\} \subseteq \mathcal{L}$. If (i) H is a complete bipartite graph with no self-loops, or (ii) H is a complete graph in which every vertex has a self-loop, then $\oplus\textsc{Hom}(H, \mathcal{L})$ can be solved in polynomial time. Otherwise, $\oplus\textsc{Hom}(H, \mathcal{L})$ is $\oplus$P-complete.*

Proof. The easiness result comes from Dyer and Greenhill [42, Theorem 1.1]. (Dyer and Greenhill's result is stated for homomorphisms rather than for list homomorphisms, but it is easy to see, and well known, that it extends to list homomorphisms.) For the hardness part we consider four cases.

Case 1: H contains at least one looped and one unlooped vertex.

The problem $\oplus$IS, of counting the independent sets of a graph, modulo 2, is known to be $\oplus$P-complete [142]. In this case there is an easy reduction from $\oplus$IS to $\oplus\textsc{LHom}(H, \mathcal{L})$. To see this, note that, since H is connected, it contains a looped vertex a which is adjacent to an unlooped vertex b. Counting the homomorphisms from a graph G to $H[\{a,b\}]$ is well-known to be equivalent to counting the independent sets of G (see, e.g., [13]). Since $\{a,b\} \in \mathcal{L}$ we can use this list to restrict the image of homomorphisms to $\{a,b\}$, giving the desired reduction.

Case 2: H is a bipartite graph without self-loops but it is not a complete bipartite graph.

In this case, H contains a path (a,b,c,d) such that $\{a,d\} \notin E(H)$ so $\oplus\text{Hom}(H,\mathcal{L})$ is $\oplus$P-complete by Lemma 5.84.

Case 3: H is a graph without self-loops that contains a cycle of odd length.

Consider a shortest odd-length cycle C in H. Due to minimality, C has to be an induced cycle of H (any additional edge between vertices of C would give a shorter even-length cycle and a shorter odd-length cycle). If C is not a triangle, then C contains a path (a,b,c,d) such that $\{a,d\} \notin E(H)$. If otherwise C is a triangle (a,b,c,a), then $\{a,a\} \notin E(H)$ since H does not have self-loops. In both cases $\oplus\text{Hom}(H,\mathcal{L})$ is $\oplus$P-complete by Lemma 5.84.

Case 4: H is a graph with all self-loops present but H is not a complete graph.

In this case, H contains a path (a,b,c) where $\{a,c\} \notin E(H)$. Since $\{b,b\} \in E(H)$ we can apply Lemma 5.84 to the walk (a,b,b,c) to obtain $\oplus$P-completeness of $\oplus\text{Hom}(H,\mathcal{L})$.

□

The following complexity classification for the problem $\oplus\text{LHom}(H)$ follows easily from Theorem 5.85.

Theorem 5.86. *Let H be graph in $\mathcal{H}$. If every connected component H' of H satisfies one of the following*

1. *H' is a complete bipartite graph with no self-loops, or*

2. *H' is a complete graph in which every vertex has a self-loop,*

then $\oplus\text{LHom}(H)$ can be solved in polynomial time. Otherwise, $\oplus\text{LHom}(H)$ is $\oplus$P-complete.

Proof. The easiness result comes from Dyer and Greenhill [42, Theorem 1.1]. For the hardness part, let H' be a connected component of H that is not a complete bipartite graph with no self-loops and is not a complete graph in which every vertex has a self-loop. Let $\mathcal{L}$ be the set of all size-2 subsets of $V(H')$. From Theorem 5.85, $\oplus\text{Hom}(H',\mathcal{L})$ is $\oplus$P-complete. However, $\oplus\text{Hom}(H',\mathcal{L})$ reduces trivially to $\oplus\text{LHom}(H)$ — given an input $(G,\mathbf{S})$ to $\oplus\text{Hom}(H',\mathcal{L})$ simply return the number of (list) homomorphisms from $(G,\mathbf{S})$ to H, modulo 2.

□

Remark 5.87. Theorem 5.86 would be unchanged if we changed the definition of $\oplus$LHom(H) so that the input G can be any graph in $\mathcal{H}$ (so it need not be simple). A self-loop on a vertex v of G simply enforces the constraint that a homomorphism must map v to a vertex of H that has a self-loop. The same constraint can be enforced using a list.

Chapter 6

Conclusion and Open Questions

Die meisten Menschen wollen nicht eher schwimmen, als bis sie es können. Ist das nicht witzig? Natürlich wollen sie nicht schwimmen! Sie sind ja für den Boden geboren, nicht für's Wasser. Und natürlich wollen sie nicht denken; sie sind ja für's Leben geschaffen, nicht für's Denken! Ja, und wer denkt, wer das Denken zur Hauptsache macht, der kann es darin zwar weit bringen, aber er hat doch eben den Boden mit dem Wasser vertauscht, und einmal wird er ersaufen.

– Hermann Hesse, *Steppenwolf (1927)*

This thesis has established several complexity classifications for large classes of homomorphism counting problems. Throughout this work we have considered fixed right-hand side problems, that is, homomorphisms from an input graph to a graph that is a parameter of the problem. We have studied three different settings, namely, exact counting, approximate counting, and modular counting. We have also studied different types of homomorphisms, e.g., unrestricted homomorphisms, surjective homomorphisms, compactions, retractions and list homomorphisms.

In Part I, we have started off by investigating the exact counting complexity of counting surjective homomorphisms and counting compactions. We have seen that the exact counting framework offers only very few tractable cases. In particular, we have shown that exactly counting compactions has even fewer tractable cases than exactly counting surjective homomorphisms, or even exactly counting list homomorphisms. This inherent difficulty of exact counting problems encourages the study of relaxed problem formulations.

This has led us, in Part II, to the world of approximate counting, where we have encountered a reduction landscape that is different from that of exact counting. We have shown that both surjective homomorphism counts and compaction counts are at most as hard to approximate as retraction counts. It is open whether the other direction holds for these reductions, or whether the problems are possibly separated. We also currently do not know an AP-reduction or a separation between the surjective homomorphism counting problem and the compaction counting problem. With an eye to the respective decision problems, one might suspect that, for each graph H, #SHom(H) AP-reduces to #Comp(H). More generally, one might wonder whether the decision complexity landscape for these problems coincides with that of approximate counting.

Prior to this work, very little was known about the complexity of approximately counting retractions to a parameter graph H. In Part II, we have given an explicit classification for all square-free H. While approximation does not yield new polynomial-time solvable cases for retraction counting, it showcases an intriguing intermediate class of #BIS-equivalent problems. Of course, the nagging open question is: Can we classify the complexity for *all* graphs, including those with squares?

The restriction to graphs without squares is not so much due to limitations of the techniques we use as it is due to the combinatorial complexity of covering all cases, which so far can only be handled by combining a number of different methods. Nevertheless, it appears that new ideas will be needed in order to obtain a classification for all graphs. One aspect is the fact that, both for irreflexive *square-free* graphs and for reflexive *square-free* graphs, the classification for approximately counting retractions coincides with that of approximately counting list homomorphisms (compare Theorems 1.10 and 1.11). This allowed us to use several of the structures that induce hardness for counting list homomorphisms to show hardness results for the retraction problem. In the square-free case, discrepancies between the complexities of these two problems occur only for mixed graphs. However, it turns out that for graphs *with squares* there are separations also within the class of irreflexive graphs and within the class of reflexive graphs. It appears that several of the structures that induced hardness for square-free graphs do not induce hardness in general. We demonstrate this with a couple of examples.

Recall that an induced WR_3 (a reflexive star with 3 leaves) is a criterion for #SAT-hardness for approximately counting retractions to *square-free* graphs (see Lemma 4.38). It is also a criterion for #SAT-hardness for approximately counting list homomorphisms (for all graphs H) [63]. However, for the graph H_1 depicted in Figure 6.1 we have $\#\textsc{Ret}(H_1) \leq_{\mathrm{AP}} \#\mathrm{BIS}$ (see Appendix C), and the vertices a, b, c, d of H_1 induce a WR_3. H_1 also contains an induced reflexive 4-cycle (using the vertices e, f, g, h), which is another structure that implies #SAT-hardness for approximately counting list homomorphisms. Similarly, with an eye to Section 4.3.2, the graph H_2 in Figure 6.1 is interesting as it indicates that mixed triangles do not in general imply #SAT-hardness for approximately counting retractions. See Appendix C for a proof that $\#\textsc{Ret}(H_2) \leq_{\mathrm{AP}} \#\mathrm{BIS}$.

For irreflexive square-free graphs, we have used the existence of an induced J_3 as condition for #SAT-hardness (Lemma 3.6). Now consider the graph H_3 from Figure 6.1. We have $\#\textsc{Ret}(H_3) \leq_{\mathrm{AP}} \#\mathrm{BIS}$ (see Appendix C) and H_3 is an irreflexive graph with an induced J_3 (using the vertices $a, \ldots, g$). It is worth noting that H_3 contains several of the induced subgraphs that serve as indicators for #SAT-hardness of the approximate counting list homomorphism problem, namely X_2, X_3 and $T_2 = J_3$, as defined in [63, Fig. 2].

These examples show that the class of retraction counting problems that are AP-interreducible with #BIS appears to be quite substantial and nuanced, and at this point we do not have a conjecture how to characterise this class — it is not even

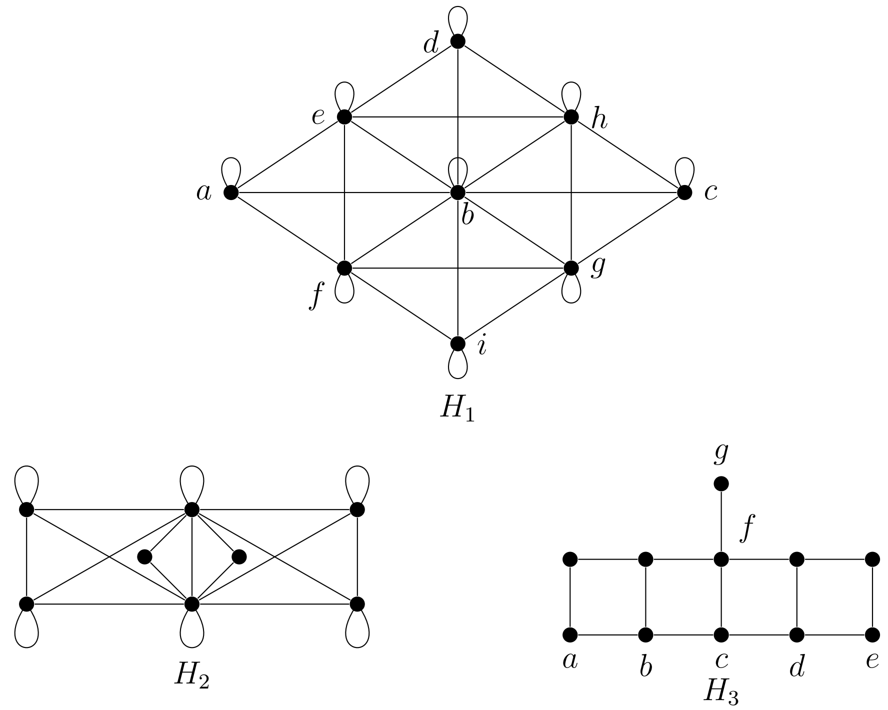


Figure 6.1: Some graphs with squares for which approximately counting retractions AP-reduces to #BIS.

clear that the complexity classification has the form of a trichotomy, even though that is our current guess.

Recall that one of the motivations for studying variants of the homomorphism counting problem was the hope of obtaining new insights about the intriguing unresolved complexity of approximately counting (unrestricted) homomorphisms. In that sense we have come full circle with the project as our work on retractions has also allowed us to establish new #BIS-easiness results for a large class of previously unresolved approximate counting homomorphism problems. On the list of open problems are also the approximate counting versions of the surjective homomorphism problem and the compaction problem — with interesting questions such as whether these frameworks contain an infinite complexity hierarchy.

Another problem worth investigating is that of approximately counting homomorphisms to directed graphs. For exact counting, a dichotomy is known due to Dyer, Goldberg and Paterson [40]. Not much is known for approximate counting. As we have shown, the #BIS-easiness framework from Section 3.2.2.1 can be applied to directed graphs, and it would be interesting to explore this further.

In Part III, we have investigated modular counting as another relaxation of the exact counting framework. This relaxation leads to a greater class of tractable problems and the main open problem is resolving the conjecture of Faben and Jerrum (Conjecture 1.25). We have introduced new gadgets with a more global flavour, and these gadgets have allowed us to resolve the conjecture for all graphs without a K_4-minor. Resolving the conjecture in its full generality still stands as an open problem. Furthermore, one might also consider the more general problem of counting modulo p, where p is any prime number.

Part IV

Appendices

Appendix A

From Chapter 2: Decomposition of $N^{\mathrm{comp}}\big(G \to K_{2,3}\big)$

In this appendix, we work through a long example to illustrate some of the definitions and ideas from Section 2.2.2. We do this by verifying the statement of Theorem 2.9 for the special case where $H = K_{2,3}$.

Of course, the theorem is already proved in the earlier sections of this paper, but we work through this example in order to help the reader gain familiarity with the definitions. For $H = K_{2,3}$ and a non-empty, irreflexive and connected graph G we want to prove

$$N^{\mathrm{comp}}\big(G \to H\big) = \sum_{J \in \mathcal{S}_H} \lambda_H(J) N\big(G \to J\big). \tag{A.1}$$

First, we set $\mathcal{S}_H = \{H_1, \dots, H_{10}\}$, cf. Figure A.1, as defined in Definition 2.7.

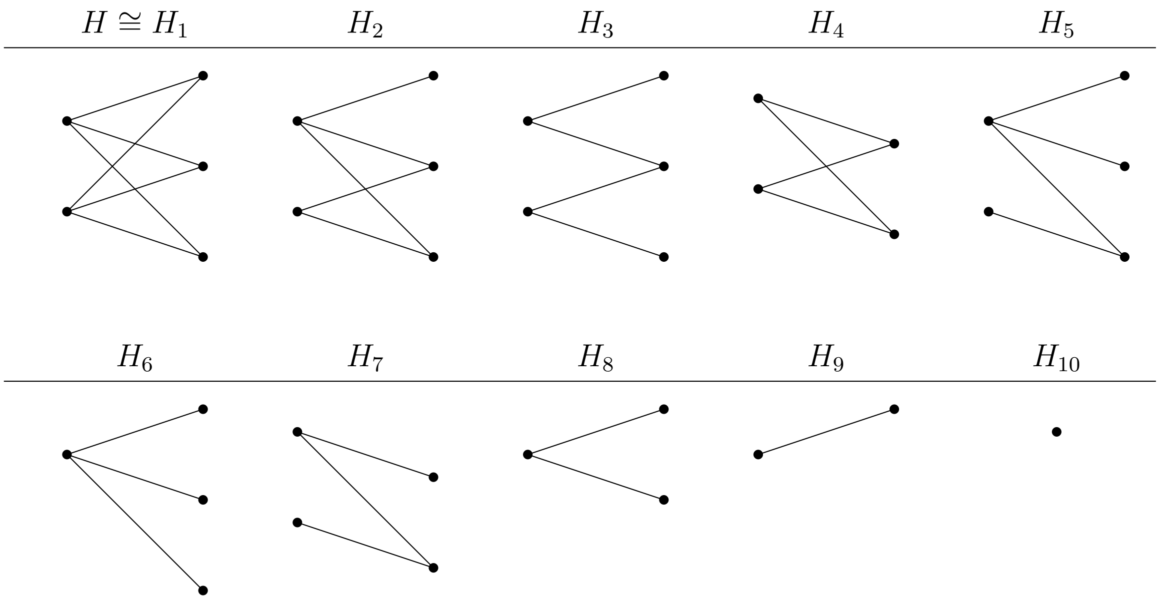


Figure A.1: $\mathcal{S}_H = \{H_1, \dots, H_{10}\}$

Next, we recall the definitions of μ_H and λ_H from Definitions 2.7 and 2.8. For $J \in \mathcal{S}_H$, $\mu_H(J)$ is the number of non-empty connected subgraphs of H that are isomorphic to J. Also, $\lambda_H(J) = 1$ if $J \cong H$. If otherwise J is isomorphic to some graph in $\mathcal{S}_H$ but $J \ncong H$, we have

$$\lambda_H(J) = -\sum_{\substack{H' \in \mathcal{S}_H \\ \text{s.t. } H' \ncong H}} \mu_H(H')\lambda_{H'}(J). \tag{A.2}$$

In order to verify (A.1), we have to determine $\lambda_H(J)$ for all $J \in S_H$. As $\lambda_H(J)$ is defined inductively by (A.2), we first determine $\lambda_{H'}(J)$ for all $H' \in \mathcal{S}_H$ with $H' \ncong H$.

We start with the graph H_{10} and determine $\lambda_{H_{10}}$. Clearly, H_{10} has only one connected subgraph and we can choose $\mathcal{S}_{H_{10}} = \{H_{10}\}$. Recall that $\lambda_{H_{10}}(J) = 0$ for all graphs J that are not isomorphic to any graph in $\mathcal{S}_{H_{10}}$, i.e. not isomorphic to H_{10} in this case. By definition we have

$$\mu_{H_{10}}(H_{10}) = 1 \quad \text{as well as} \quad \lambda_{H_{10}}(H_{10}) = 1, \text{ see Table A.1.}$$

This conforms with our intuition as for the single vertex graph H_{10}, it clearly holds that

$$N^{\text{comp}}\big(G \to H_{10}\big) = N\big(G \to H_{10}\big). \tag{A.3}$$

Thus, we have now verified (A.1) for $H = H_{10}$.

Using this information, we consider the graph H_9 next and determine μ_{H_9} and λ_{H_9} for $\mathcal{S}_{H_9} = \{H_9, H_{10}\}$, see Table A.2. H_9 contains two connected subgraphs that are isomorphic to H_{10}, therefore $\mu_{H_9}(H_{10}) = 2$. Then, from (A.2) we obtain

$$\lambda_{H_9}(H_{10}) = -\sum_{H' \in \{H_{10}\}} \mu_{H_9}(H')\lambda_{H'}(H_{10}) = -2.$$

Plugging this into (A.1) for $H = H_9$, we get

$$\begin{aligned} N^{\text{comp}}\big(G \to H_9\big) &= \sum_{J \in \mathcal{S}_{H_9}} \lambda_{H_9}(J) N\big(G \to J\big) \\ &= N\big(G \to H_9\big) - 2N\big(G \to H_{10}\big). \end{aligned} \tag{A.4}$$

Now let us verify this expression. Recall that G is connected. The central idea behind our approach is that every homomorphism from G to H_9 is a compaction onto some connected subgraph H' of H_9. Furthermore, $\mu_{H_9}(H')$ tells us how many such subgraphs there are that are isomorphic to H'. Thus,

$$\begin{aligned} N\big(G \to H_9\big) &= \mu_{H_9}(H_9) \cdot N^{\text{comp}}\big(G \to H_9\big) + \mu_{H_9}(H_{10}) \cdot N^{\text{comp}}\big(G \to H_{10}\big) \\ &= N^{\text{comp}}\big(G \to H_9\big) + 2N^{\text{comp}}\big(G \to H_{10}\big). \end{aligned}$$

Rearranging and using the fact that $N^{\text{comp}}\big(G \to H_{10}\big) = N\big(G \to H_{10}\big)$ from (A.3):

$$\begin{aligned} N^{\text{comp}}\big(G \to H_9\big) &= N\big(G \to H_9\big) - 2N^{\text{comp}}\big(G \to H_{10}\big) \\ &= N\big(G \to H_9\big) - 2N\big(G \to H_{10}\big). \end{aligned}$$

Thus, we have now proved (A.4) which in turn proves (A.1) for $H = H_9$.

Using (A.3) and (A.4) we can now go on to find (see Table A.3) that

$$N^{\mathrm{comp}}\big(G \to H_8\big) = N\big(G \to H_8\big) - 2N\big(G \to H_9\big) + N\big(G \to H_{10}\big)$$

and so on.

This gives the intuition behind the formal definitions of μ_H and λ_H. For completeness, we give the values for all graphs H_1 through H_{10} in Tables A.1 through A.10. From Table A.10 we can conclude that for $H = K_{2,3}$ the statement of Theorem 2.9 gives

$$\begin{aligned} N^{\mathrm{comp}}\big(G \to K_{2,3}\big) = N\big(G \to K_{2,3}\big) &- 6N\big(G \to H_2\big) + 6N\big(G \to H_3\big) \\ &+ 3N\big(G \to H_4\big) + 6N\big(G \to H_5\big) - 2N\big(G \to H_6\big) \\ &- 12N\big(G \to H_7\big) + 3N\big(G \to H_8\big). \end{aligned}$$

Table A.1: Decomposition of H_{10}

H'	H_{10}
$\mu_{H_{10}}(H')$	1
$\lambda_{H_{10}}(H')$	1

Table A.2: Decomposition of H_9

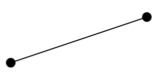
H'	H_9	H_{10}
		
$\mu_{H_9}(H')$	1	2
$\lambda_{H_9}(H')$	1	-2

Table A.3: Decomposition of H_8

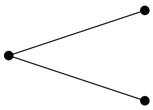
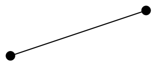
H'	H_8	H_9	H_{10}
			
$\mu_{H_8}(H')$	1	2	3
$\lambda_{H_8}(H')$	1	−2	1

Table A.4: Decomposition of H_7

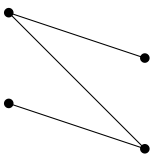
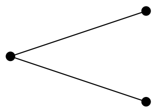
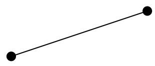
H'	H_7	H_8	H_9	H_{10}
				
$\mu_{H_7}(H')$	1	2	3	4
$\lambda_{H_7}(H')$	1	−2	1	0

Table A.5: Decomposition of H_6

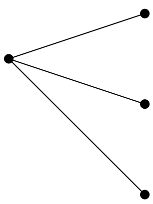
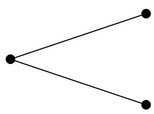
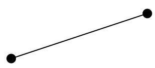
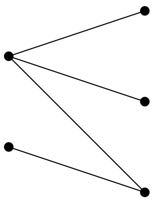
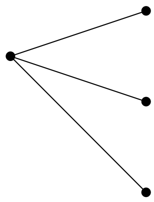
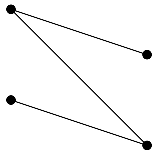
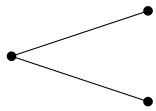
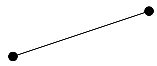
H'	H_6	H_8	H_9	H_{10}
				
$\mu_{H_6}(H')$	1	3	3	4
$\lambda_{H_6}(H')$	1	−3	3	−1

Table A.6: Decomposition of H_5

H'	H_5	H_6	H_7	H_8	H_9
					
$\mu_{H_5}(H')$	1	1	2	4	4
$\lambda_{H_5}(H')$	1	-1	-2	3	-1

H'	H_{10}
$\mu_{H_5}(H')$	5
$\lambda_{H_5}(H')$	0

Table A.7: Decomposition of H_4

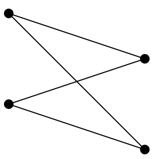
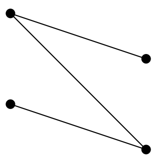
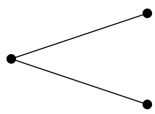
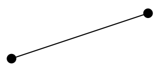
H'	H_4	H_7	H_8	H_9	H_{10}
					
$\mu_{H_4}(H')$	1	4	4	4	4
$\lambda_{H_4}(H')$	1	-4	4	0	0

Table A.8: Decomposition of H_3

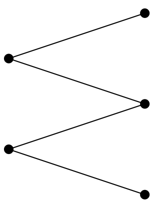
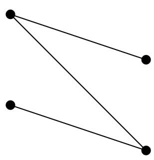
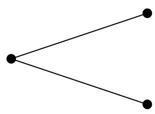
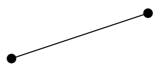
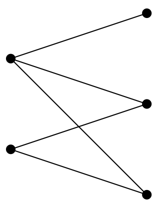
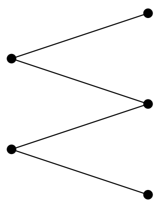
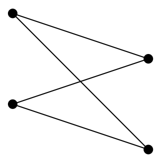
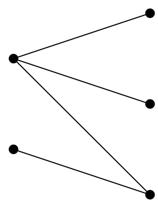
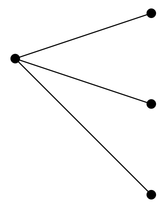
H'	H_3	H_7	H_8	H_9	H_{10}
					
$\mu_{H_3}(H')$	1	2	3	4	5
$\lambda_{H_3}(H')$	1	-2	1	0	0

Table A.9: Decomposition of H_2

H'	H_2	H_3	H_4	H_5	H_6
					
$\mu_{H_2}(H')$	1	2	1	2	1
$\lambda_{H_2}(H')$	1	-2	-1	-2	1

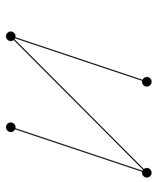
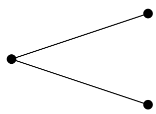
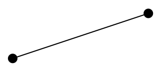
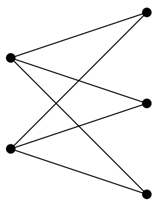
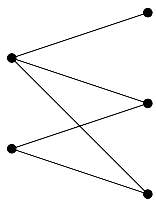
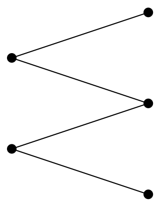
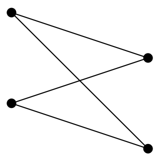
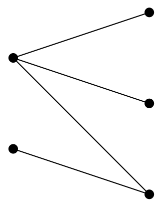
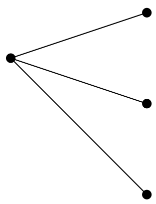
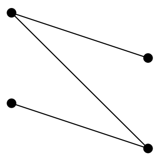
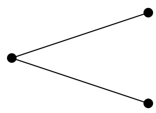
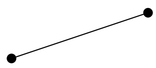
H'	H_7	H_8	H_9	H_{10}
				
$\mu_{H_2}(H')$	6	6	5	5
$\lambda_{H_2}(H')$	6	-3	0	0

Table A.10: Decomposition of $H_1 = K_{2,3}$

H'	H_1	H_2	H_3	H_4	H_5
					
$\mu_{H_1}(H')$	1	6	6	3	6
$\lambda_{H_1}(H')$	1	−6	6	3	6

H'	H_6	H_7	H_8	H_9	H_{10}
					
$\mu_{H_1}(H')$	2	12	9	6	5
$\lambda_{H_1}(H')$	−2	−12	3	0	0

Appendix B

From Chapter 3: Proof of Lemma 3.20 from [58]

As noted on page 81, Lemma 3.20 is actually subsumed by Lemma 4.33. Here, we present our original proof from [58], which does not rely on definitions from Chapter 4.

In order to show #SAT-hardness we use a reduction from counting large cuts in a graph G. We use graph gadgets to model these cuts. We replace each vertex of G by a graph J such that the number of homomorphisms from J to H is dominated by exactly two "types" of homomorphisms. These two types encode the two parts of a cut. In Table 1 we give all types that represent a significant share of the set of homomorphisms. In Lemma B.14 we show how to choose parameters of the graph J to ensure that only 2 significant types remain. In the proof of Lemma 3.20 we verify another desired property, which is that the two types interact in an "anti-ferromagnetic" way to ensure that large cuts dominate.

At this point we introduce the gadget graph J and introduce some of its properties. Note that a similar but simpler gadget has been used in [37] and [108].

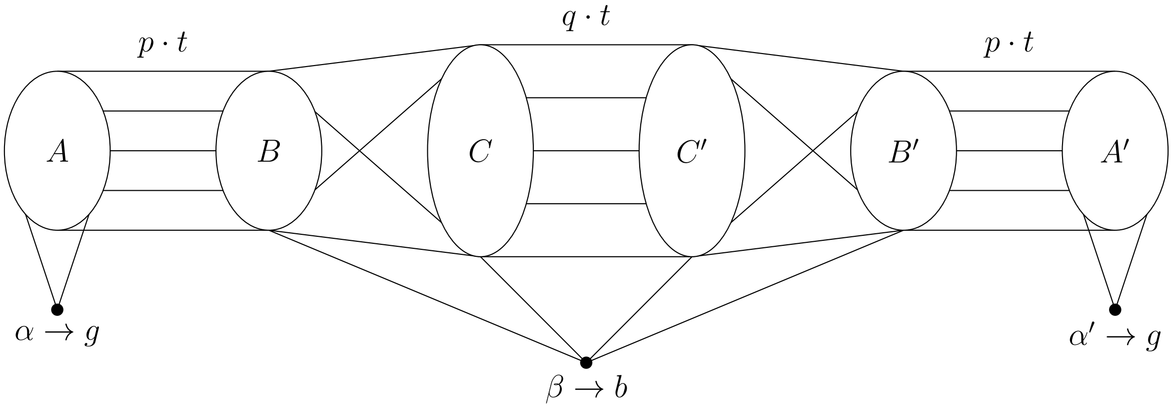


Figure B.1: The graph J. A label of the form $v \to u$ means that the vertex $v \in V(J)$ is pinned to $u \in V(H)$ since $S_v = \{u\}$.

Definition B.1. For sets X and Y we define $X \times Y = \{\{x, y\} \mid x \in X, y \in Y\}$ as an undirected version of the usual definition of the Cartesian product.

Definition B.2. We now define the graph J, as visualised in Figure B.1. Let p, q and t be positive integers — these are parameters of J. Let A, A', B and B' be independent sets of size $p \cdot t$ and let C and C' be independent sets of size $q \cdot t$. These six sets are pairwise disjoint. In addition, we introduce vertices α, α' and β that are distinct from each other and the remaining vertices. The vertex set of J is the union of $\{\alpha, \alpha', \beta\}$ and the sets A, A', B, B', C and C'. As displayed in Figure B.1, the edge set of J is defined as follows. The set of edges $\mathcal{M}_1$ between the vertices of A and B forms a perfect matching (every vertex in A is adjacent to exactly one vertex in B and vice versa). The set of edges $\mathcal{M}_2$ between the vertices of C and C' and the edges $\mathcal{M}_3$ between the vertices of A' and B' form perfect matchings respectively. Then

$$E(J) = \bigcup_{i \in [3]} \mathcal{M}_i \cup \Big(B \times C\Big) \cup \Big(B' \times C'\Big)$$
$$\cup \Big(\{\alpha\} \times A\Big) \cup \Big(\{\alpha'\} \times A'\Big) \cup \Big(\{\beta\} \times (B \cup C \cup C' \cup B')\Big).$$

This completes the definition of the graph J.

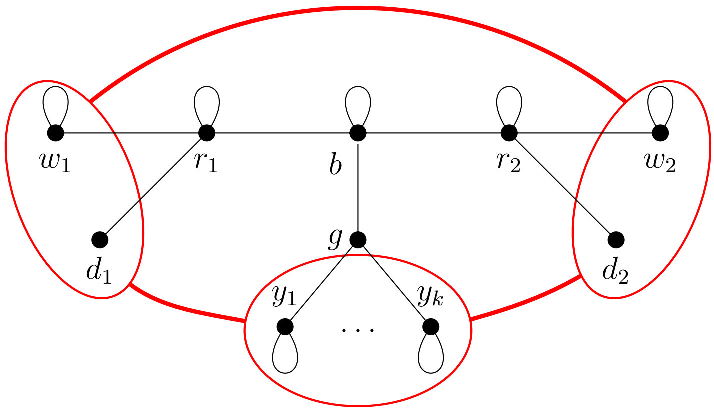


Figure B.2: The graph H_k. Circled sets of vertices are independent sets of possibly looped vertices. Sets of vertices that are connected by a thick red edge have a complete set of edges between them.

For any positive integer k let H_k be the graph as shown in Figure B.2. The vertex set of H_k is $\{w_1, d_1, r_1, w_2, d_2, r_2, b, g, y_1, \dots, y_k\}$. All of these vertices are looped except for d_1, d_2 and g. The non-loop edges of H_k are the edges in

$$\{\{w_1, r_1\}, \{w_2, r_2\}, \{d_1, r_1\}, \{d_2, r_2\}, \{r_1, b\}, \{r_2, b\}, \{b, g\}, \{g, y_1\}, \dots, \{g, y_k\}\},$$

together with those in $\{w_1, d_1\} \times \{w_2, d_2\}$, $\{w_1, d_1\} \times \{y_1, \dots, y_k\}$ and $\{w_2, d_2\} \times \{y_1, \dots, y_k\}$. The significance of this graph will become clear in the proof of Lemma 3.20.

For a graph J we define the vertex lists $S_\alpha = \{g\}$, $S_{\alpha'} = \{g\}$, and $S_\beta = \{b\}$. Also, for all $v \in V(J) \setminus \{\alpha, \alpha', \beta\}$, we define $S_v = V(H_k)$. Finally, we let $\mathbf{S}_J = \{S_v \mid v \in V(J)\}$. In order to investigate the number of homomorphisms from $(J, \mathbf{S}_J)$ to H_k, we set up the following notation. Suppose that U and V are subsets of $V(J)$ and that h is a homomorphism $h \in \mathcal{H}((J, \mathbf{S}_J), H_k)$. We define

- $h(V) = \{h(x) \mid x \in V\}$ and
- $h(U, V) = \{(h(x), h(y)) \mid x \in U, v \in V, \{x, y\} \in E(J)\}$.

We say that $(h(A, B), h(C, C'), h(B', A'))$ is the *type* of h. We will partition the set $\mathcal{H}((J, \mathbf{S}_J), H_k)$ into different classes by type. Formally, a type is a tuple $T = (T_1, T_2, T_3)$ where each T_i is a subset of $\{(x, y) \mid x \in V(H_k), y \in V(H_k), \{x, y\} \in E(H_k)\}$. The type of a homomorphism gives a lot of information. Given a type $T = (T_1, T_2, T_3)$, let $A(T) = \{x \mid \exists y\, (x, y) \in T_1\}$, $B(T) = \{y \mid \exists x\, (x, y) \in T_1\}$, $C(T) = \{x \mid \exists y\, (x, y) \in T_2\}$, $C'(T) = \{y \mid \exists x\, (x, y) \in T_2\}$, $B'(T) = \{x \mid \exists y\, (x, y) \in T_3\}$, and $A'(T) = \{y \mid \exists x\, (x, y) \in T_3\}$. If a homomorphism $h \in \mathcal{H}((J, \mathbf{S}_J), H_k)$ has type T then it is clear from the definition of J that $h(A) = A(T)$, $h(B) = B(T)$, $h(C) = C(T)$, $h(C') = C'(T)$, $h(B') = B'(T)$ and $h(A') = A'(T)$. A type T is called *non-empty* if there exists a homomorphism from $(J, \mathbf{S}_J)$ to H_k that has type T, otherwise it is called empty. The following observation follows from the definition of J.

Observation B.3. *A type $T = (T_1, T_2, T_3)$ is non-empty if and only if*

(1) T_1, T_2 and T_3 are non-empty,

(2) $B(T) \cup C(T) \cup C'(T) \cup B'(T) \subseteq \Gamma(b)$,

(3) $A(T) \cup A'(T) \subseteq \Gamma(g)$,

(4) $B(T) \times C(T) \subseteq E(H_k)$ and $B'(T) \times C'(T) \subseteq E(H_k)$.

Given a type $T = (T_1, T_2, T_3)$ we define $N(T)$ to be the number of homomorphisms in $\mathcal{H}((J, \mathbf{S}_J), H_k)$ that have type T. We also set $\widehat{N}(T) = |T_1|^{pt}|T_2|^{qt}|T_3|^{pt}$. In Lemma B.5 we show that, for non-empty T, $\widehat{N}(T)$ is a close approximation to $N(T)$.

We use the following technical fact. Let $f_{\mathrm{sur}}(a, b)$ be the number of surjective functions from a set of a elements to a set of b elements.

Lemma B.4 ([37, Lemma 18])**.** *If a and b are positive integers and $a \geq 2b \ln b$, then*

$$b^a\left(1 - \exp\left(-\frac{a}{2b}\right)\right) \leq f_{\mathrm{sur}}(a, b) \leq b^a.$$

Lemma B.5. *Let p and q be positive integers. There exists a positive integer t_0 such that for all $t \geq t_0$ and all non-empty types T of the corresponding graph J we have*

$$\frac{\widehat{N}(T)}{2} \leq N(T) \leq \widehat{N}(T).$$

Proof. Let $T = (T_1, T_2, T_3)$ be a non-empty type. Then

$$N(T) = f_{\text{sur}}(p \cdot t, |T_1|) \cdot f_{\text{sur}}(q \cdot t, |T_2|) \cdot f_{\text{sur}}(p \cdot t, |T_3|). \tag{B.1}$$

For fixed p and q and sufficiently large t_0 we know from Lemma B.4 that for all $t \geq t_0$ we have

$$1 - \exp\left(-\frac{p \cdot t}{2|T_1|}\right) \geq (1/2)^{1/3},$$

an analogous bound holds for the other two factors in Equation (B.1). The statement of the lemma then directly follows from Lemma B.4. □

Definition B.6. We say that a type $T = (T_1, T_2, T_3)$ is *maximal* if it is non-empty and every type $T' = (T'_1, T'_2, T'_3)$ with $T' \neq T$, $T_1 \subseteq T'_1$, $T_2 \subseteq T'_2$ and $T_3 \subseteq T'_3$ is empty.

Using this definition of maximality we prove that the number of homomorphisms in $\mathcal{H}((J, \mathbf{S}_J), H_k)$ that have a maximal type is exponentially larger as a function of t than the number of homomorphisms that have non-maximal types. Note that the precise value of the fraction $\frac{31+12k}{32+12k}$ that appears in the following lemmas is not important, we only need it to be smaller than 1. This particular bound uses the fact that, for any type (T_1, T_2, T_3), the sets T_1, T_2 and T_3 have cardinality at most $2|E(H_k)| = 32 + 12k$.

Constraint B.7. In our proofs we will need the fact that the parameters p and q of J are sufficiently large with respect to the number of edges in H_k. In particular, we require that $p, q \geq 2|E(H_k)| = 32 + 12k$.

Lemma B.8. *Let T be a non-empty type that is not maximal. Then there exists a non-empty type T^* such that $\widehat{N}(T) \leq \left(\frac{31+12k}{32+12k}\right)^t \widehat{N}(T^*)$.*

Proof. Let $T = (T_1, T_2, T_3)$ be a non-empty type that is not maximal. Then there exists a non-empty type $T^* = (T^*_1, T^*_2, T^*_3)$ with $T^* \neq T$ and $T_i \subseteq T^*_i$ for $i \in [3]$. Since $T^* \neq T$ there exists an index $i \in [3]$ such that $T_i \subsetneq T^*_i$, i.e. $|T_i| \leq |T^*_i| - 1$. Then (using the fact that $p, q \geq 1$)

$$\frac{\widehat{N}(T)}{\widehat{N}(T^*)} = \frac{|T_1|^{pt}|T_2|^{qt}|T_3|^{pt}}{|T^*_1|^{pt}|T^*_2|^{qt}|T^*_3|^{pt}} \leq \left(\frac{|T^*_i| - 1}{|T^*_i|}\right)^t \leq \left(\frac{2|E(H_k)| - 1}{2|E(H_k)|}\right)^t \leq \left(\frac{31 + 12k}{32 + 12k}\right)^t.$$

□

Definition B.9. Let $E = E(H_k)$. For all $X \subseteq V(H_k)$ and $Y \subseteq V(H_k)$ we set $E(X, Y) = \{(x, y) \mid x \in X, y \in Y, \{x, y\} \in E\}$.

For a set of vertices S in a graph H recall the definition of the set of common neighbours $\Gamma(S)$ and the set of all neighbours $\Phi(S)$ from Section 3.1.2.

Lemma B.10. *Let $T = (T_1, T_2, T_3)$ be a maximal type. Then*

(1) $T_1 = E(A(T), B(T))$, $T_2 = E(C(T), C'(T))$ and $T_3 = E(B'(T), A'(T))$. Also,

(2) $C(T) = \Gamma\big(\Gamma(C(T)) \cap \Gamma(b)\big) \cap \Gamma(b)$ and $C'(T) = \Gamma\big(\Gamma(C'(T)) \cap \Gamma(b)\big) \cap \Gamma(b)$.

(3) $B(T) = \Gamma(C(T)) \cap \Gamma(b)$ *and* $B'(T) = \Gamma(C'(T)) \cap \Gamma(b)$ *.*

(4) $A(T) = \Phi(B(T)) \cap \Gamma(g)$ *and* $A'(T) = \Phi(B'(T)) \cap \Gamma(g)$*.*

Proof. Let $T = (T_1, T_2, T_3)$ be a non-empty type.

Proof of (1): It is clear from the definitions that $T_1 \subseteq E(A(T), B(T))$, $T_2 \subseteq E(C(T), C'(T))$ and $T_3 \subseteq E(B'(T), A'(T))$. Suppose that T_2 is a strict subset of $E(C(T), C'(T))$. We will show that T is not maximal. To this end, consider the type $T^* = (T_1, E(C(T), C'(T)), T_3)$. Note that $A(T^*) = A(T)$, $B(T^*) = B(T)$, $C(T^*) = C(T)$, $C'(T^*) = C'(T)$, $B'(T^*) = B'(T)$ and $A'(T^*) = A'(T)$. Using Observation B.3 and the fact that T is non-empty, we conclude that T^* is non-empty. Using the definition of maximality (comparing T to T^*) we conclude that T is not maximal. Similarly, if T_1 is a strict subset of $E(A(T), B(T))$ or if T_3 is a strict subset of $E(B'(T), A'(T))$ then T is not maximal.

Proof of (2): Let $X = \Gamma\big(\Gamma(C(T)) \cap \Gamma(b)\big)$ and $S = X \cap \Gamma(b)$. If $y \in \Gamma(C(T)) \cap \Gamma(b)$ then y is certainly adjacent to everything in $C(T)$, so $C(T) \subseteq X$. Since $C(T) \subseteq \Gamma(b)$ by Observation B.3, we conclude that $C(T)$ is a subset of S. Similarly, defining $X' = \Gamma\big(\Gamma(C'(T)) \cap \Gamma(b)\big)$ and $S' = X' \cap \Gamma(b)$, we have $C'(T) \subseteq S'$. Thus, $T_2 \subseteq E(S, S')$. Consider the type $T^* = (T_1, E(S, S'), T_3)$.

- We first show that T^* is non-empty. Note that $A(T^*) = A(T)$, $B(T^*) = B(T)$, $A'(T^*) = A'(T)$ and $B'(T^*) = B'(T)$. Also, $C(T^*) \subseteq S \subseteq \Gamma(b)$ and $C'(T^*) \subseteq S' \subseteq \Gamma(b)$. Using Observation B.3 and the fact that T is non-empty, we must check that $B(T) \times C(T^*) \subseteq E(H_k)$ and $B'(T) \times C'(T^*) \subseteq E(H_k)$. To do this, we will check that $B(T) \times S \subseteq E(H_k)$ and $B'(T) \times S' \subseteq E(H_k)$.

  We start with the first of these. Since T is non-empty, Observation B.3 guarantees that $B(T) \subseteq \Gamma(C(T)) \cap \Gamma(b)$. So it suffices to show that $\big(\Gamma(C(T)) \cap \Gamma(b)\big) \times S \subseteq E(H_k)$, which follows from the definition of S. The proof that $B'(T) \times S' \subseteq E(H_k)$ is similar. We have shown that T^* is non-empty.

- We next show that $C(T^*) = S$. We have already established that $C(T^*) \subseteq S$. The vertex b is adjacent to everything in $\Gamma(b)$ so it is adjacent to everything in the subset $\Gamma(C'(T)) \cap \Gamma(b)$ hence $b \in X'$. Since b has a loop, this implies $b \in S'$. By the definition of T^* it follows that $S \subseteq C(T^*)$, and hence $C(T^*) = S$, as required. We can similarly show that $C'(T^*) = S'$.

Suppose that $C(T)$ is a strict subset of S. Comparing T to T^*, we find that T_2 is a strict subset of $E(S, S')$ so T is not maximal. Similarly, if $C'(T)$ is a strict subset of S' then T is not maximal.

Proof of (3): It is immediate from Observation B.3 that $B(T) \subseteq \Gamma(C(T)) \cap \Gamma(b)$ and $B'(T) \subseteq \Gamma(C'(T)) \cap \Gamma(b)$.

Suppose that $B(T)$ is a strict subset of $\Gamma(C(T)) \cap \Gamma(b)$. We will show that T is not maximal. To this end, let v be any vertex in $\Gamma(C(T)) \cap \Gamma(b) \setminus B(T)$ and consider the type $T^* = (T_1 \cup \{(b, v)\}, T_2, T_3)$. Observation B.3 shows that T^* is non-empty, so T is not maximal. Similarly, if $B'(T)$ is a strict subset of $\Gamma(C'(T)) \cap \Gamma(b)$ then T is not maximal.

Proof of (4): It is immediate from Observation B.3 and the definition of a type that $A(T) \subseteq \Phi(B(T)) \cap \Gamma(g)$ and $A'(T) \subseteq \Phi(B'(T)) \cap \Gamma(g)$. If either of these subset inclusions is strict then, as in the proof of (3), it is straightforward to see that T is not maximal. □

Lemma B.11. *Let T be a maximal type. Then $C(T)$ and $C'(T)$ are both in the set*

$$\{\{b\}, \{r_1, b\}, \{r_2, b\}\{r_1, r_2, b, g\}\}.$$

Proof. We will prove this for $C(T)$. The argument for $C'(T)$ is the same. From Observation B.3, $C(T)$ is a (not necessarily strict) subset of $\Gamma(b) = \{r_1, r_2, b, g\}$ (and it is non-empty).

- If $g \in C(T)$ then $\Gamma(C(T)) \cap \Gamma(b) = \{b\}$ so, by item (2) of Lemma B.10, $C(T) = \Gamma(b) = \{r_1, r_2, b, g\}$.
- If $r_1 \in C(T)$ and $r_2 \in C(T)$ then $\Gamma(C(T)) \cap \Gamma(b) = \{b\}$ so, again, $C(T) = \Gamma(b) = \{r_1, r_2, b, g\}$.
- If $C(T) = \{r_1\}$ then $\Gamma(C(T)) \cap \Gamma(b) = \{r_1, b\}$ so, by item (2) of Lemma B.10, $C(T) = \{r_1, b\}$, which is a contradiction.
- Similarly, the case $C(T) = \{r_2\}$ gives a contradiction.

This covers all possible cases. □

Definition B.12. For $i \in [6]$ let $X_i \subseteq V(H_k)$. We say that the types $(E(X_1, X_2), E(X_3, X_4), E(X_5, X_6))$ and $(E(X_6, X_5), E(X_4, X_3), E(X_2, X_1))$ are *symmetric* to each other.

Note that if T and T' are symmetric to each other it holds that $N(T) = N(T')$.

Lemma B.13. *All maximal types are listed in Table B.1 (except for those that are symmetric to a listed type). Furthermore, for each listed type T we give the corresponding value for $\widehat{N}(T)$.*

Proof. First, Lemma B.11 gives the 4 possibilities for $C(T)$ and $C'(T)$. Up to symmetry, this gives the 10 possibilities listed in the table.

Next, for each of the 10 possibilities, we use items (3) and (4) of Lemma B.10 to compute the corresponding sets $A(T)$, $B(T)$, $B'(T)$ and $A'(T)$.

Now item (1) of Lemma B.10 guarantees that $T_1 = E(A(T), B(T))$, $T_2 = E(C(T), C'(T))$ and $T_3 = E(B'(T), A'(T))$. So

$$\widehat{N}(T) = |E(A(T), B(T))|^{pt} |E(C(T), C'(T))|^{qt} |E(B'(T), A'(T))|^{pt}.$$

These quantities are all computed in the table. □

Let $T_1, \ldots, T_{10}$ be the types as given in Table B.1.

Table B.1: Maximal types of the homomorphisms in $\mathcal{H}(J, \mathbf{S}_J), H_k)$.

	$A(T)$	$B(T)$	$C(T)$	$C'(T)$	$B'(T)$	$A'(T)$	$\widehat{N}(T)$
T_1	$\{b\} \cup \mathcal{Y}$	$\{r_1, r_2, b, g\}$	$\{b\}$	$\{b\}$	$\{r_1, r_2, b, g\}$	$\{b\} \cup \mathcal{Y}$	$(4+k)^{pt} \cdot 1^{qt} \cdot (4+k)^{pt}$
T_2	$\{b\} \cup \mathcal{Y}$	$\{r_1, r_2, b, g\}$	$\{b\}$	$\{r_1, b\}$	$\{r_1, b\}$	$\{b\}$	$(4+k)^{pt} \cdot 2^{qt} \cdot 2^{pt}$
T_3	$\{b\} \cup \mathcal{Y}$	$\{r_1, r_2, b, g\}$	$\{b\}$	$\{r_2, b\}$	$\{r_2, b\}$	$\{b\}$	$(4+k)^{pt} \cdot 2^{qt} \cdot 2^{pt}$
T_4	$\{b\} \cup \mathcal{Y}$	$\{r_1, r_2, b, g\}$	$\{b\}$	$\{r_1, r_2, b, g\}$	$\{b\}$	$\{b\}$	$(4+k)^{pt} \cdot 4^{qt} \cdot 1^{pt}$
T_5	$\{b\}$	$\{r_1, b\}$	$\{r_1, b\}$	$\{r_2, b\}$	$\{r_2, b\}$	$\{b\}$	$2^{pt} \cdot 3^{qt} \cdot 2^{pt}$
T_6	$\{b\}$	$\{r_1, b\}$	$\{r_1, b\}$	$\{r_1, b\}$	$\{r_1, b\}$	$\{b\}$	$2^{pt} \cdot 4^{qt} \cdot 2^{pt}$
T_7	$\{b\}$	$\{r_2, b\}$	$\{r_2, b\}$	$\{r_2, b\}$	$\{r_2, b\}$	$\{b\}$	$2^{pt} \cdot 4^{qt} \cdot 2^{pt}$
T_8	$\{b\}$	$\{r_1, b\}$	$\{r_1, b\}$	$\{r_1, r_2, b, g\}$	$\{b\}$	$\{b\}$	$2^{pt} \cdot 6^{qt} \cdot 1^{pt}$
T_9	$\{b\}$	$\{r_2, b\}$	$\{r_2, b\}$	$\{r_1, r_2, b, g\}$	$\{b\}$	$\{b\}$	$2^{pt} \cdot 6^{qt} \cdot 1^{pt}$
T_{10}	$\{b\}$	$\{b\}$	$\{r_1, r_2, b, g\}$	$\{r_1, r_2, b, g\}$	$\{b\}$	$\{b\}$	$1^{pt} \cdot 9^{qt} \cdot 1^{pt}$

Note: Recall that p, q and t are the parameters of J where p and q satisfy Constraint B.7. Each line corresponds to a type $\Big(E\big(A(T), B(T)\big), E\big(C(T), C'(T)\big), E\big(B'(T), A'(T)\big)\Big)$. To shorten the notation we set $\mathcal{Y} = \{y_i \mid i \in [k]\}$.

Lemma B.14. *Let k be a positive integer. Then there is a $\gamma \in (0,1)$ and positive integers p and q that satisfy Constraint B.7 such that, for all $i \in [10]$ except $i = 4$ and all positive integers t, we have $\widehat{N}(T_i) \le \gamma^t \widehat{N}(T_4)$.*

Proof. We choose integers $p, q \ge 32 + 12k$ (p and q satisfy Constraint B.7) such that

$$\log_4(4+k) < \frac{q}{p} < \log_{9/4}(4+k). \tag{B.2}$$

This is possible as $\log_4(4+k) < \log_{9/4}(4+k)$ for all $k > 0$. Suppose that T and T' are types listed in Table B.1 which are distinct from T_4 and have the property that $\widehat{N}(T') < \widehat{N}(T)$. Then the sought-for bound automatically holds for T' if it holds for T.

We check the sought-for bound for each $i \in [10]$, $i \neq 4$:

T_1: $\frac{\widehat{N}(T_1)}{\widehat{N}(T_4)} = (4+k)^{pt}(1/4)^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $(4+k)^p/4^q < 1$. This is true as $\log_4(4+k) < \frac{q}{p}$ by (B.2).

T_2 (and T_3): $\frac{\widehat{N}(T_2)}{\widehat{N}(T_4)} = 2^{pt}(1/2)^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $2^p/2^q < 1$. This is true as $q > p$ by (B.2).

T_5: $\widehat{N}(T_5) < \widehat{N}(T_6)$.

T_6 (and T_7): $\frac{\widehat{N}(T_6)}{\widehat{N}(T_4)} = (4/(4+k))^{pt} < \gamma^t$ is fulfilled for $4/(4+k) \le 4/5 < \gamma < 1$.

T_8 (and T_9): $\frac{\widehat{N}(T_8)}{\widehat{N}(T_4)} = (2/(4+k))^{pt}(3/2)^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $(2/(4+k))^p(3/2)^q < 1$. This is true as $\frac{q}{p} < \log_{9/4}(4+k) < \log_{3/2}((4+k)/2)$ by (B.2) and for all $k > 0$.

T_{10}: $\frac{\widehat{N}(T_{10})}{\widehat{N}(T_4)} = (1/(4+k))^{pt}(9/4)^{qt} < \gamma^t$ is fulfilled for some sufficiently large $\gamma < 1$ if and only if $(1/(4+k))^p(9/4)^q < 1$. This is true as $\frac{q}{p} < \log_{9/4}(4+k)$ by (B.2).

□

In the statement of the lemma we refer to the graph H'_k as depicted in Figure B.3.

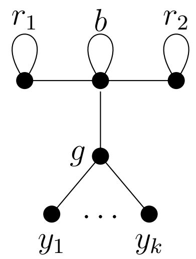


Figure B.3: The graph H'_k.

Lemma 3.20. *Let H be graph that has a looped vertex b such that, for some positive integer k, H'_k (see Figure 3.8) is a subgraph of $H[\Gamma^2(b)]$, and $H[\Gamma^2(b)]$ in turn is a subgraph of H_k (see Figure 3.7). Then* #SAT $\leq_{\mathrm{AP}}$ #RET(H).

Proof. We use a reduction from #LARGECUT, which is known to be #SAT-hard (see [37]). A *cut* of a graph G is a partition of $V(G)$ into two subsets (the order of this pair is ignored) and the size of a cut is the number of edges that have exactly one endpoint in each of these two subsets.

Name: #LARGECUT.
Input: An integer $K \geq 1$ and a connected graph G in which every cut has size at most K.
Output: The number of size-K cuts in G.

Let G and K be an input to #LARGECUT, n be the number of vertices of G and $\varepsilon \in (0,1)$ be the parameter of the desired precision of approximation in the AP-reduction. From G we construct an input $(G', \mathbf{S})$ to #RET(H) by introducing vertex and edge gadgets. By the assumption of the lemma, the vertex b of H has $\Gamma(b) = \{b, r_1, r_2, g\}$ where b, r_1 and r_2 are looped and g is not and $\Gamma(g) = \{b, y_1, \ldots, y_k\}$ with $k \geq 1$.

Let p, q be positive integers that are chosen such that they fulfil Constraint B.7 and (B.2). Note that p and q only depend on k which is a parameter of the fixed graph H and therefore do not depend on the input G. We will define the parameter t of the gadget graph J to be $t = n^4$. We also define a new parameter $s = n + 1$.

For each vertex $v \in V(G)$ we introduce a vertex gadget G'_v which is a graph J with parameters p, q and t as given in Definition B.2. We denote the corresponding sets A, B, C, C', B', A' by $A_v, B_v, C_v, C'_v, B'_v$ and A'_v, respectively. It is fine to keep the notation for the remaining vertices as α, α' and β as technically these vertices can be thought of as identical vertices over all gadgets because of their pinning. We say that two gadgets G'_u and G'_v are adjacent if u and v are adjacent in G.

We connect vertex gadgets as follows. For every edge $e = \{u, v\} \in E(G)$ we introduce an edge gadget as follows. We introduce two size-s independent sets, denoted by S_e and S'_e. We set $V'_e = S_e \cup S'_e$. As shown in Figure B.4 we construct the set of edges

$$E'_e = (C_u \times S_e) \cup (C'_u \times S'_e) \cup (C_v \times S'_e) \cup (C'_v \times S_e) \cup (\{\beta\} \times S_e) \cup (\{\beta\} \times S'_e).$$

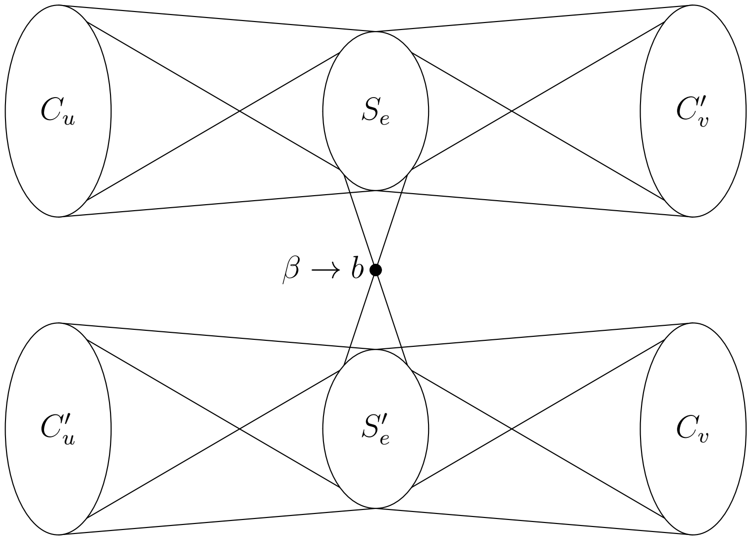


Figure B.4: The edge gadget for the edge $e = \{u, v\}$.

Putting the pieces together, G' is the graph with

$$V(G') = \bigcup_{v \in V(G)} V(G'_v) \cup \bigcup_{e \in E(G)} V'_e \qquad \text{and} \qquad E(G') = \bigcup_{v \in V(G)} E(G'_v) \cup \bigcup_{e \in E(G)} E'_e.$$

Finally, we define the vertex lists $S_\alpha = S_{\alpha'} = \{g\}$, $S_\beta = \{b\}$ and $S_v = V(H)$ for all $v \in V(G') \setminus \{\alpha, \alpha', \beta\}$. Then $\mathbf{S} = \{S_v \mid v \in V(G')\}$. This completes the definition of the instance $(G', \mathbf{S})$.

The pinning of the vertex β (via the list S_β) ensures that every homomorphism from $(G', \mathbf{S})$ to H is a homomorphism from $(G', \mathbf{S})$ to $H[\Gamma^2(b)]$. By the assumption of the lemma, $H[\Gamma^2(b)]$ is a subgraph of H_k. We make a case distinction based on the graph $H[\Gamma^2(b)]$.

Case 1: $H[\Gamma^2(b)] = H_k$**.** Let h be a homomorphisms from $(G', \mathbf{S})$ to H, v be some vertex of G and G'_v be the corresponding vertex gadget. Then by our definition of $(G', \mathbf{S})$ we observe that $h|_{V(G'_v)}$ corresponds to a homomorphism from $(J, \mathbf{S}_J)$ to H_k and therefore has a type.

We say that a homomorphism from $(G', \mathbf{S})$ to H is *full* if its restriction to each vertex gadget is either of type T_4 (from Table B.1) or of its symmetric type (let us call it T'_4). Each full homomorphism h defines a cut as it partitions $V(G)$ into those vertices v for which $h|_{G'_v}$ has type T_4 and those for which $h|_{G'_v}$ has type T'_4. We say that a full homomorphism is *K-large* if the size of the corresponding cut is equal to K, otherwise we say that the homomorphism is *K-small*. Consider a full homomorphism h from $(G', \mathbf{S})$ to H_k.

- For an edge $e = \{u, v\}$ of G suppose that $h|_{G'_u}$ has type T_4 and $h|_{G'_v}$ has type T'_4. Note that by the definition of the edge gadget, we have $h(S_e) \subseteq \Gamma(h(C_u)) \cap \Gamma(h(C'_v))$. Then the vertices in S_e can be mapped to any of the 4 neighbours of b, whereas all vertices in S'_e have to be mapped to b (since $h(S'_e) \subseteq \Gamma(h(C'_u)) \cap \Gamma(h(C_v))$ where $C'_u = C_v = \{r_1, r_2, b, g\}$ and b is the sole common neighbour of r_1, r_2, b and g).
- Suppose instead that $h|_{G'_u}$ and $h|_{G'_v}$ have the same type T_4 or T'_4. Then the homomorphism h has to map the vertices in both S_e and S_v to b.

Thus, every pair of adjacent gadgets of different types contributes a factor of 4^s to the number of full homomorphisms, whereas every pair of adjacent gadgets of the same type only contributes a factor of 1. Recall the definition of $N(T)$ as the number of homomorphisms from $(J, \mathbf{S}_J)$ to H_k that have type T. Then for $\ell \geq 1$ every size-ℓ cut of G arises in $2 \cdot N(T_4)^n \cdot 4^{s\ell}$ ways as a full homomorphism from $(G', \mathbf{S})$ to H_k.

Let N be the number of solutions to #LargeCut with input G and K (our goal is to approximate this number). We partition the homomorphisms from $(G', \mathbf{S})$ to H_k into three different sets. Z^* is the number of K-large (full) homomorphisms, Z_1 is the number of homomorphisms that are full but K-small and Z_2 is the number of non-full homomorphisms. Then we have $N = Z^*/(2N(T_4)^n 4^{sK})$ and $N\big((G', \mathbf{S}) \to H\big) = N\big((G', \mathbf{S}) \to H_k\big) = Z^* + Z_1 + Z_2$. Thus it remains to show that $(Z_1 + Z_2)/(2N(T_4)^n 4^{sK}) \leq 1/4$ for our choice of p, q, t and s. Under this assumption we then have $N\big((G', \mathbf{S}) \to H\big)/(2N(T_4)^n 4^{sK}) \in [N, N + 1/4]$ and a single oracle call to determine $N\big((G', \mathbf{S}) \to H\big)$ with precision $\delta = \varepsilon/21$ suffices to determine N with the sought-for precision as demonstrated in [37, Proof of Theorem 3].

Now we prove $(Z_1 + Z_2)/(2N(T_4)^n 4^{sK}) \leq 1/4$. As there are at most 2^n ways to assign a type T_4 or T'_4 to the n vertex gadgets in G' we have $Z_1 \leq 2^n \cdot N(T_4)^n \cdot 4^{s(K-1)}$. We next obtain the following bound since $s = n + 1$:

$$\frac{Z_1}{2N(T_4)^n 4^{sK}} \leq \frac{2^n N(T_4)^n 4^{s(K-1)}}{2N(T_4)^n 4^{sK}} = \frac{2^n}{2 \cdot 4^s} \leq \frac{1}{8}.$$

We obtain a similar bound for Z_2. From Lemmas B.8, B.13 and B.14 we know that for our choice of p and q there exists $\gamma \in (0, 1)$ such that for every type T that is not T_4 or T'_4 we have $\widehat{N}(T) \leq \gamma^t \widehat{N}(T_4)$. Using Lemma B.5 this gives $N(T) \leq 2\gamma^t N(T_4)$ for sufficiently large t with respect to p, q and k (which only depend on H but not on the input G). Since $t = n^4$ we can assume that t is sufficiently large with respect to p and q as otherwise the input size is bounded by a constant (in which case we can solve #LargeCut in constant time).

For each type $T = (T_1, T_2, T_3)$, the cardinality of each set T_i is bounded above by $2|E(H_k)| = 32 + 12k$ and hence there are at most $\big(2^{32+12k}\big)^3$ different types.

Furthermore, as H_k has $8+k$ vertices, there are at most $(8+k)^{2sn^2}$ possible functions from the at most $2sn^2$ vertices in $\bigcup_{e\in E(G)}(S_e \cup S'_e)$ to vertices in H_k. Since $t = n^4$ and $s = n+1$ we obtain

$$\frac{Z_2}{2N(T_4)^n 4^{sK}} \le \frac{\left(2^{32+12k}\right)^{3n} \cdot N(T_4)^{n-1} \cdot 2\gamma^t N(T_4) \cdot (8+k)^{2sn^2}}{2N(T_4)^n 4^{sK}}$$
$$= \gamma^t \cdot \frac{\left(2^{32+12k}\right)^{3n}(8+k)^{2sn^2}}{4^{sK}} \le \frac{1}{8}.$$

The last inequality holds for sufficiently large n as

$$\frac{\left(2^{32+12k}\right)^{3n}(8+k)^{2sn^2}}{4^{sK}} \le C^{n^3}$$

for some positive constant C that only depends on H, but not on the input G, whereas $t = n^4$. **(End of Case 1)**

Case 2: $H[\Gamma^2(b)] \neq H_k$**.** By the assumption of the lemma, $H[\Gamma^2(b)]$ is a subgraph of H_k. Let $\mathcal{H}'$ be the set of homomorphisms in $\mathcal{H}((J, \mathbf{S}_J), H_k)$ that are homomorphisms from J to $H[\Gamma^2(b)]$. Then for each type T the number of homomorphisms in $\mathcal{H}'$ of type T is at most the number of homomorphisms in $\mathcal{H}((J, \mathbf{S}_J), H_k)$ that have type T.

Note that the type T_4 (and its symmetric type) only uses vertices and edges from H'_k and we know that H'_k is a subgraph of $H[\Gamma^2(b)]$ by the assumption of the lemma. Therefore each homomorphism which is of type T_4 is also in $\mathcal{H}'$ (their number remains unchanged). The analysis is then analogous to that of Case 1. (The number of K-large and K-small homomorphisms stays the same whereas the number of non-full homomorphisms can only decrease as we only need to consider a subset of the previous types and the number of homomorphisms that have a particular type can only decrease.) **(End of Case 2)**

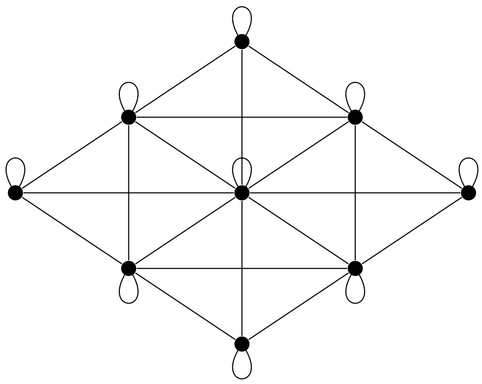
□

Appendix C

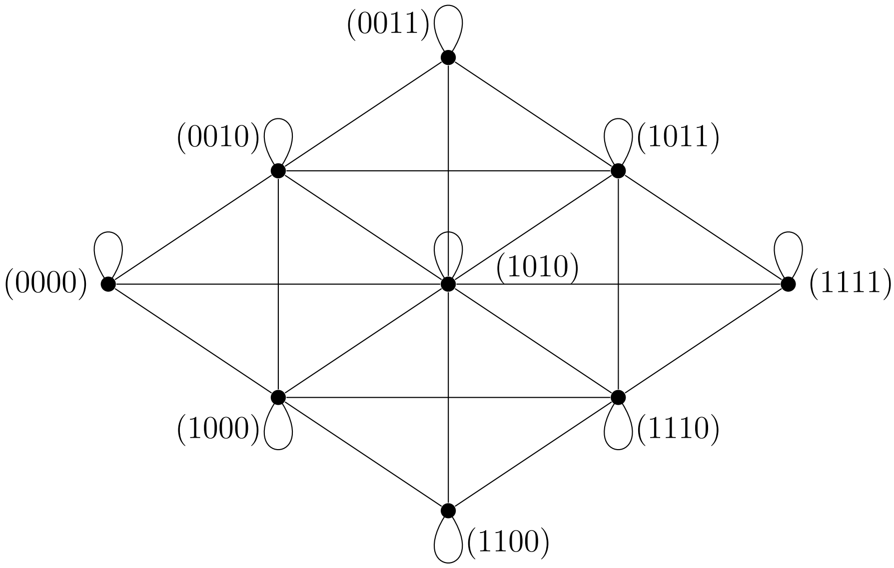
Some #BIS-Easiness Results for Graphs with Squares

In this appendix we use the definitions and notation from Chapter 3. We give #BIS-easiness results for the problem $\#\text{Ret}(H)$ for some graphs H of particular interest (explained in Chapter 6).

Lemma C.1. *For the following graph H we have $\#\text{Ret}(H) \leq_{\text{AP}} \#\text{BIS}$:*



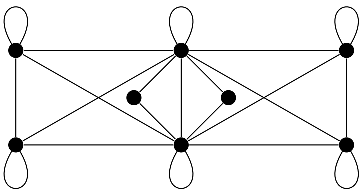
Proof. Let $X = \{x_1, x_2, x_3, x_4\}$. Let $I_\text{v} = (X, C_\text{v})$ and $I_\text{e} = (X, C_\text{e})$, where $C_\text{v} = C_\text{e} = \{\text{Imp}(x_2, x_1), \text{Imp}(x_4, x_3)\}$. Then $H_{I_\text{v}, I_\text{e}}$ is the following graph (each vertex v corresponds to a satisfying assignment σ_v of I_v and is labelled with $(\sigma_v(x_1)\, \sigma_v(x_2)\, \sigma_v(x_3)\, \sigma_v(x_4))$):



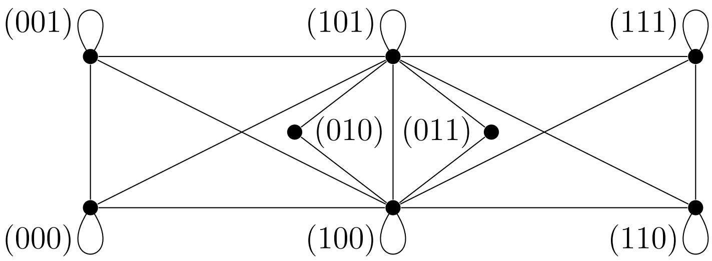
H_{I_v,I_e} is isomorphic to H, and the result follows from Lemma 4.1.

□

Lemma C.2. *For the following graph H we have* $\#\text{RET}(H) \leq_{\text{AP}} \#\text{BIS}$*:*



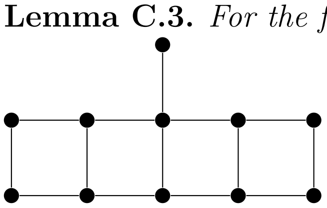
Proof. Let $X = \{x_1, x_2, x_3\}$. Let $I_v = (X, C_v)$, where $C_v = \emptyset$, and let $I_e = (X, C_e)$, where $C_e = \{\text{Imp}(x_2, x_1)\}$. Then H_{I_v,I_e} is the following graph (each vertex v corresponds to a satisfying assignment σ_v of I_v and is labelled with $(\sigma_v(x_1)\,\sigma_v(x_2)\,\sigma_v(x_3))$):



H_{I_v,I_e} is isomorphic to H, and the result follows from Lemma 4.1.

□

Lemma C.3. *For the following graph H we have* $\#\text{RET}(H) \leq_{\text{AP}} \#\text{BIS}$*:*



In order to show Lemma C.3 we use the fact that the #BIS-easiness technique from Section 3.2.2.1 can be strengthened when applied to irreflexive bipartite graphs. That strengthening is possible due to the fact that for bipartite H we can also assume bipartite graphs G as input. We can then use different vertex instances of #CSP({Imp}) to encode the two parts of the bipartition, which gives more modelling power. This strengthening has been used before by Kelk [108, Section 7.3.4] in the context of counting downsets of a partial order. We transfer that idea to the more general framework of constraint satisfaction problems. For completeness we now state the details, which closely follow the presentation of Section 3.2.2.1. Afterwards we show which instances to use in order to encode the graph from Lemma C.3.

Definition C.4. Let $I_{\mathrm{L}} = (X_L, C_{\mathrm{L}})$, $I_{\mathrm{R}} = (X_R, C_{\mathrm{R}})$, $I_{\mathrm{LR}} = (X_L \cup X_R, C_{\mathrm{LR}})$, and $I_{\mathrm{RL}} = (X_L \cup X_R, C_{\mathrm{RL}})$ be instances of #CSP({Imp}) such that every constraint in C_{LR} is of the form $\mathrm{Imp}(x, y)$ with $x \in X_L$ and $y \in X_R$, and every constraint in C_{RL} is of the form $\mathrm{Imp}(x, y)$ with $x \in X_R$ and $y \in X_L$. We define the bipartite graph $H_{I_{\mathrm{L}},I_{\mathrm{R}},I_{\mathrm{LR}},I_{\mathrm{RL}}}$ as follows. The vertices of $H_{I_{\mathrm{L}},I_{\mathrm{R}},I_{\mathrm{LR}},I_{\mathrm{RL}}}$ form a bipartition (L_H, R_H), where the vertices in L_H are the satisfying assignments of I_{L}, and the vertices of R_H are the satisfying assignments of I_{R}. Given any assignments $\sigma_\ell \in L_H$ and $\sigma_r \in R_H$, there is an edge $\{\sigma_\ell, \sigma_r\}$ in $H_{I_{\mathrm{L}},I_{\mathrm{R}},I_{\mathrm{LR}},I_{\mathrm{RL}}}$ if and only if the following holds:

- For every constraint $\mathrm{Imp}(x, y)$ in I_{LR}, we have $\sigma_\ell(x) \Rightarrow \sigma_r(y)$.
- For every constraint $\mathrm{Imp}(x, y)$ in I_{RL}, we have $\sigma_r(x) \Rightarrow \sigma_\ell(y)$.

Lemma C.5. *Let* $I_{\mathrm{L}} = (X_L, C_{\mathrm{L}})$, $I_{\mathrm{R}} = (X_R, C_{\mathrm{R}})$, $I_{\mathrm{LR}} = (X_L, C_{\mathrm{LR}})$, *and* $I_{\mathrm{RL}} = (X_L, C_{\mathrm{RL}})$ *be instances of* #CSP({Imp}) *such that every constraint in* C_{LR} *is of the form* $\mathrm{Imp}(x, y)$ *with* $x \in X_L$ *and* $y \in X_R$*, and every constraint in* C_{RL} *is of the form* $\mathrm{Imp}(x, y)$ *with* $x \in X_R$ *and* $y \in X_L$*. Then* $\#\textsc{Ret}(H_{I_{\mathrm{L}},I_{\mathrm{R}},I_{\mathrm{LR}},I_{\mathrm{RL}}}) \leq_{\mathrm{AP}} \#\mathrm{BIS}$.

Proof. We first show the reduction from $\#\textsc{Ret}(H_{I_{\mathrm{L}},I_{\mathrm{R}},I_{\mathrm{LR}},I_{\mathrm{RL}}})$ to $\#\mathrm{CSP}(\{\mathrm{Imp}, \delta_0, \delta_1\})$. The result then follows from Lemma 3.9. Let $(G, \mathbf{S})$ be an instance of $\#\textsc{Ret}(H_{I_{\mathrm{L}},I_{\mathrm{R}},I_{\mathrm{LR}},I_{\mathrm{RL}}})$. If G is not bipartite (which can be recognised in polynomial time) then $N\big((G, \mathbf{S}) \to H_{I_{\mathrm{L}},I_{\mathrm{R}},I_{\mathrm{LR}},I_{\mathrm{RL}}}\big) = 0$ (and we are done). So, suppose that G is a bipartite graph, i.e., G is of the form $G = ((L_G, R_G), E)$ (such a bipartition can be found in polynomial time). Let (L_H, R_H) be the bipartition defined in Definition C.4. Further, let $\mathcal{H}((L_G, R_G, \mathbf{S}), (L_H, R_H))$ be the set of homomorphisms from $(G, \mathbf{S})$ to $H_{I_{\mathrm{L}},I_{\mathrm{R}},I_{\mathrm{LR}},I_{\mathrm{RL}}}$ that map vertices in L_G to vertices in L_H and vertices in R_G to vertices in R_H. For a desired approximation to $N\big((G, \mathbf{S}) \to H_{I_{\mathrm{L}},I_{\mathrm{R}},I_{\mathrm{LR}},I_{\mathrm{RL}}}\big)$ with precision ε it suffices to show that we can obtain an approximation to $|\mathcal{H}((L_G, R_G, \mathbf{S}), (L_H, R_H))|$ with precision ε (as we can rename the parts of the bipartition of $V(G)$ in order to approximate $|\mathcal{H}((R_G, L_G, \mathbf{S}), (L_H, R_H))|$).

From $(L_G, R_G, \mathbf{S})$ we create an instance I of $\#\mathrm{CSP}(\{\mathrm{Imp}, \delta_0, \delta_1\})$ as follows. The set of variables of I is $(L_G \times X_L) \cup (R_G \times X_R)$ and the set of constraints C of I is constructed as follows.

(1) For each $v \in L_G$ and each constraint $\mathrm{Imp}(x, y) \in I_{\mathrm{L}}$, we add the constraint $\mathrm{Imp}((v, x), (v, y))$ to C.

(2) For each $v \in R_G$ and each constraint $\mathrm{Imp}(x, y) \in I_{\mathrm{R}}$, we add the constraint $\mathrm{Imp}((v, x), (v, y))$ to C.

(3) For each edge $\{u, v\} \in E(G)$ with $u \in L_G$ and $v \in R_G$, and for each constraint $\mathrm{Imp}(x, y) \in I_{\mathrm{LR}}$, we add the constraint $\mathrm{Imp}((u, x), (v, y))$ to C. Further, for each constraint $\mathrm{Imp}(x, y) \in I_{\mathrm{RL}}$, we add the constraint $\mathrm{Imp}((v, x), (u, y))$ to C.

(4) For each $v \in L_G$ with $|S_v| = 1$ let τ be the (only) element of S_v. If $\tau \in R_H$ then $|\mathcal{H}((L_G, R_G, \mathbf{S}), (L_H, R_H))| = 0$ (and we are done). Otherwise, if $\tau \in L_H$, for each $x \in X_L$ we proceed as follows. If $\tau(x) = 0$ then add the constraint $\delta_0((v, x))$ to C. Otherwise, add the constraint $\delta_1((v, x))$ to C.

(5) For each $v \in R_G$ with $|S_v| = 1$ let τ be the (only) element of S_v. If $\tau \in L_H$ then $|\mathcal{H}((L_G, R_G, \mathbf{S}), (L_H, R_H))| = 0$ (and we are done). Otherwise, if $\tau \in R_H$, for each $x \in X_R$ we proceed as follows. If $\tau(x) = 0$ then add the constraint $\delta_0((v, x))$ to C. Otherwise, add the constraint $\delta_1((v, x))$ to C.

To complete the reduction from $\#\mathrm{RET}(H_{I_{\mathrm{L}}, I_{\mathrm{R}}, I_{\mathrm{LR}}, I_{\mathrm{RL}}})$ to $\#\mathrm{CSP}(\{\mathrm{Imp}, \delta_0, \delta_1\})$, we will show that there is a bijection between the satisfying assignments of I and the homomorphisms in $\mathcal{H}((L_G, R_G, \mathbf{S}), (L_H, R_H))$. This bijection ensures that the number of satisfying assignments of I is equal to $|\mathcal{H}((L_G, R_G, \mathbf{S}), (L_H, R_H))|$. Hence the approximation to $|\mathcal{H}((L_G, R_G, \mathbf{S}), (L_H, R_H))|$ can be achieved using a single oracle call to $\#\mathrm{CSP}(\{\mathrm{Imp}, \delta_0, \delta_1\})$ with the desired accuracy ε.

To establish the bijection, we present an (invertible) map from satisfying assignments of I to homomorphisms in $\mathcal{H}((L_G, R_G, \mathbf{S}), (L_H, R_H))$. The map is constructed as follows. Let σ be any satisfying assignment of I.

- For every vertex $v \in L_G$, define a function $\sigma_v \colon X_L \to \{0, 1\}$ as follows. For all $x \in X_L$, let $\sigma_v(x) = \sigma((v, x))$. The constraints added to C in item (1) ensure that, since σ is a satisfying assignment of I, the assignment σ_v is a satisfying assignment of I_{L}. Thus, σ_v is a vertex of L_H.

- For every vertex $v \in R_G$, define a function $\sigma_v \colon X_R \to \{0, 1\}$ as follows. For all $x \in X_R$, let $\sigma_v(x) = \sigma((v, x))$. The constraints added to C in item (2) ensure that, since σ is a satisfying assignment of I, the assignment σ_v is a satisfying assignment of I_{R}. Thus, σ_v is a vertex of R_H.

- Next, we will argue that the function from $V(G)$ to $V(H_{I_{\mathrm{L}}, I_{\mathrm{R}}, I_{\mathrm{LR}}, I_{\mathrm{RL}}})$ that maps every vertex $v \in V(G)$ to σ_v is a homomorphism from $(G, \mathbf{S})$ to $H_{I_{\mathrm{L}}, I_{\mathrm{R}}, I_{\mathrm{LR}}, I_{\mathrm{RL}}}$.

  – Consider an edge $\{u, v\}$ of G with $u \in L_G$ and $v \in R_G$. We must show that $\{\sigma_u, \sigma_v\}$ is an edge of $H_{I_{\mathrm{L}}, I_{\mathrm{R}}, I_{\mathrm{LR}}, I_{\mathrm{RL}}}$. Using Definition C.4, this is equivalent to showing that,

    ∗ for every constraint $\mathrm{Imp}(x, y)$ in I_{LR}, we have $\sigma_u(x) \Rightarrow \sigma_v(y)$, and

    ∗ for every constraint $\mathrm{Imp}(x, y)$ in I_{RL}, we have $\sigma_v(x) \Rightarrow \sigma_u(y)$.

  Using the construction of σ_u and σ_v, this is equivalent to showing that,

 * for every constraint $\mathrm{Imp}(x,y)$ in I_{LR}, we have $\sigma(u,x) \Rightarrow \sigma(v,y)$, and
 * for every constraint $\mathrm{Imp}(x,y)$ in I_{RL}, we have $\sigma(v,x) \Rightarrow \sigma(u,y)$.

 This is ensured by the fact that σ is a satisfying assignment of I, so it satisfies the constraints added in item (3).

- Consider a vertex $v \in V(G)$ with $S_v = \{\tau\}$. We must show that $\sigma_v = \tau$. This is ensured by the constraints added in items (4) and (5).

Starting from the satisfying assignment σ of I, we produced a homomorphism in $\mathcal{H}((L_G, R_G, \mathbf{S}), (L_H, R_H))$, namely the homomorphism that maps every vertex $v \in V(G)$ to σ_v. To finish the proof, we need only note that this construction is invertible — given any homomorphism in $\mathcal{H}((L_G, R_G, \mathbf{S}), (L_H, R_H))$ we let σ_v denote the image of v under this homomorphism. Given the collection $\{\sigma_v \mid v \in V(G)\}$, we construct an assignment σ from $(L_G \times X_L) \cup (R_G \times X_R)$ to $\{0,1\}$ by inverting the above construction: For every $(v,x) \in (L_G \times X_L) \cup (R_G \times X_R)$, let $\sigma((v,x)) = \sigma_v(x)$. We must then check that σ is satisfying.

- For each $v \in L_G$, the assignment σ satisfies the relevant constraints added in item (1) because σ_v is a vertex of L_H, hence a satisfying assignment of I_{L}.

- For each $v \in R_G$, the assignment σ satisfies the relevant constraints added in item (2) because σ_v is a vertex of R_H, hence a satisfying assignment of I_{R}.

- For each $\{u,v\} \in E(G)$ with $u \in L_G$ and $v \in R_G$, and each constraint $\mathrm{Imp}((u,x),(v,y))$ with $\mathrm{Imp}(x,y) \in I_{\mathrm{LR}}$ added to C in item (3), σ satisfies the constraint because $\{\sigma_u, \sigma_v\}$ is an edge of $H_{I_{\mathrm{L}},I_{\mathrm{R}},I_{\mathrm{LR}},I_{\mathrm{RL}}}$ (so $\sigma_u(x) \implies \sigma_v(y)$). Analogously, for each constraint $\mathrm{Imp}((v,x),(u,y))$ with $\mathrm{Imp}(x,y) \in I_{\mathrm{RL}}$ added to C in item (3), σ satisfies the constraint because $\sigma_v(x) \implies \sigma_u(y)$.

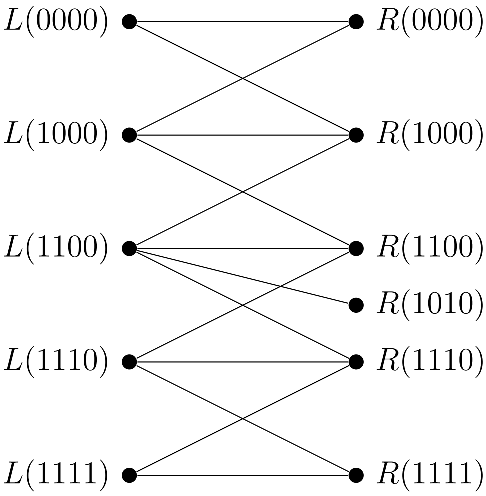
- Finally, for any $s \in \{0,1\}$, consider a constraint $\delta_s((v,x))$ introduced in items (4) or (5). The procedure in items (4) or (5) ensures that, for some τ with $S_v = \{\tau\}$, we have $\tau(x) = s$. Thus, because the homomorphism has $\sigma_v = \tau$ the constraint $\sigma((v,x)) = s$ is satisfied by σ.

□

With Lemma C.5 we can now prove Lemma C.3.

Proof of Lemma C.3. Let $X_L = \{x_{L1}, x_{L2}, x_{L3}, x_{L4}\}$ and $X_R = \{x_{R1}, x_{R2}, x_{R3}, x_{R4}\}$. Let $I_{\mathrm{L}} = (X_L, C_{\mathrm{L}})$, where $C_{\mathrm{L}} = \{\mathrm{Imp}(x_{Lj}, x_{Li}) \mid i,j \in [4], i < j\}$, and let $I_{\mathrm{R}} = (X_R, C_{\mathrm{R}})$, where $C_{\mathrm{R}} = \{\mathrm{Imp}(x_{Rj}, x_{Ri}) \mid i,j \in [4], i < j\} \setminus \{\mathrm{Imp}(x_{R3}, x_{R2})\}$. Let $I_{\mathrm{LR}} = (X_L \cup X_R, C_{\mathrm{LR}})$ with $C_{\mathrm{LR}} = \{\mathrm{Imp}(x_{Lj}, x_{Ri}) \mid i,j \in [4], i < j\}$, and let $I_{\mathrm{RL}} = (X_L \cup X_R, C_{\mathrm{RL}})$ with $C_{\mathrm{LR}} = \{\mathrm{Imp}(x_{Rj}, x_{Li}) \mid i,j \in [4], i < j\}$.

Then $H_{I_{\mathrm{L}},I_{\mathrm{R}},I_{\mathrm{LR}},I_{\mathrm{RL}}}$ is the following graph, where each vertex $v \in L_G$ corresponds to a satisfying assignment σ_v of I_{L} and is labelled with $L(\sigma_v(x_{L1})\, \sigma_v(x_{L2})\, \sigma_v(x_{L3})\, \sigma_v(x_{L4}))$; and each vertex $v \in L_R$ corresponds to a satisfying assignment σ_v of I_{R} and is labelled with $R(\sigma_v(x_{R1})\, \sigma_v(x_{R2})\, \sigma_v(x_{R3})\, \sigma_v(x_{R4})))$



$H_{I_{\mathrm{L}},I_{\mathrm{R}},I_{\mathrm{LR}},I_{\mathrm{RL}}}$ is isomorphic to H, and the result follows from Lemma C.5.

□

Bibliography

[1] Miriam Backens. A new holant dichotomy inspired by quantum computation. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 16:1–16:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[2] Miriam Backens, Andrei Bulatov, Leslie Ann Goldberg, Colin McQuillan, and Stanislav Živný. Boolean approximate counting CSPs with weak conservativity, and implications for ferromagnetic two-spin. *Journal of Computer and System Sciences*, 109:95–125, 2020.

[3] Miriam Backens and Leslie Ann Goldberg. Holant clones and the approximability of conservative holant problems. *ACM Transactions on Algorithms (TALG)*, 16(2):1–55, 2020.

[4] Ivona Bezáková, Andreas Galanis, Leslie Ann Goldberg, and Daniel Štefankovič. Inapproximability of the independent set polynomial in the complex plane. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1234–1240. ACM, 2018.

[5] M. Bíró, M. Hujter, and Zs. Tuza. Precoloring extension. I. Interval graphs. *Discrete Math.*, 100(1-3):267–279, 1992. Special volume to mark the centennial of Julius Petersen's "Die Theorie der regulären Graphs", Part I.

[6] Manuel Bodirsky, Jan Kára, and Barnaby Martin. The complexity of surjective homomorphism problems—a survey. *Discrete Appl. Math.*, 160(12):1680–1690, 2012.

[7] Hans L. Bodlaender, Klaus Jansen, and Gerhard J. Woeginger. Scheduling with incompatible jobs. *Discrete Appl. Math.*, 55(3):219–232, 1994.

[8] Jan Bok, Richard Brewster, Tomás Feder, Pavol Hell, and Nikola Jedličková. List homomorphism problems for signed graphs. *arXiv preprint arXiv:2005.05547*, 2020.

[9] Magnus Bordewich. On the approximation complexity hierarchy. *Approximation and Online Algorithms*, pages 37–46, 2011.

[10] Christian Borgs, Jennifer Chayes, Jeff Kahn, and László Lovász. Left and right convergence of graphs with bounded degree. *Random Structures Algorithms*, 42(1):1–28, 2013.

[11] Christian Borgs, Jennifer Chayes, László Lovász, Vera T. Sós, and Katalin Vesztergombi. Counting graph homomorphisms. In *Topics in discrete mathematics*, volume 26 of *Algorithms Combin.*, pages 315–371. Springer, Berlin, 2006.

[12] Karol Borsuk. Sur les rétractes. *Fundamenta Mathematicae*, 17(1):152–170, 1931.

[13] Graham R. Brightwell and Peter Winkler. Graph homomorphisms and phase transitions. *J. Combin. Theory Ser. B*, 77(2):221–262, 1999.

[14] Andrei Bulatov and Martin Grohe. The complexity of partition functions. *Theoretical Computer Science*, 348(2-3):148–186, 2005.

[15] Andrei A Bulatov. The complexity of the counting constraint satisfaction problem. *Journal of the ACM (JACM)*, 60(5):1–41, 2013.

[16] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *58th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2017*, pages 319–330. IEEE Computer Soc., Los Alamitos, CA, 2017.

[17] Andrei A Bulatov, Martin Dyer, Leslie Ann Goldberg, Mark Jerrum, and Colin McQuillan. The expressibility of functions on the boolean domain, with applications to counting CSPs. *Journal of the ACM (JACM)*, 60(5):1–36, 2013.

[18] Andrei A. Bulatov and Dániel Marx. The complexity of global cardinality constraints. *Log. Methods Comput. Sci.*, 6(4), 2010.

[19] Robert Burton and Jeffrey E Steif. Non-uniqueness of measures of maximal entropy for subshifts of finite type. *Ergodic Theory and Dynamical Systems*, 14(2):213–235, 1994.

[20] Jin-Yi Cai and Xi Chen. A decidable dichotomy theorem on directed graph homomorphisms with non-negative weights. *computational complexity*, 28(3):345–408, 2019.

[21] Jin-Yi Cai, Xi Chen, and Pinyan Lu. Graph homomorphisms with complex values: A dichotomy theorem. *SIAM Journal on Computing*, 42(3):924–1029, 2013.

[22] Jin-yi Cai, Pinyan Lu, and Mingji Xia. Holant problems and counting CSP. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 715–724. ACM, 2009.

[23] Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In John E. Hopcroft, Emily P. Friedman, and Michael A. Harrison, editors, *Proceedings of the 9th Annual ACM Symposium on Theory of Computing, STOC 1977, May 4-6, 1977, Boulder, Colorado, USA*, pages 77–90. ACM, 1977.

[24] Hubie Chen. An algebraic hardness criterion for surjective constraint satisfaction. *Algebra universalis*, 72(4):393–401, 2014.

[25] Hubie Chen. Homomorphisms are indeed a good basis for counting: Three fixed-template dichotomy theorems, for the price of one. *arXiv preprint arXiv:1710.00234*, 2017.

[26] Hubie Chen. Algebraic global gadgetry for surjective constraint satisfaction. *arXiv preprint arXiv:2005.11307*, 2020.

[27] Hubie Chen, Radu Curticapean, and Holger Dell. The exponential-time complexity of counting (quantum) graph homomorphisms. In Ignasi Sau and Dimitrios M. Thilikos, editors, *Graph-Theoretic Concepts in Computer Science - 45th International Workshop, WG 2019, Vall de Núria, Spain, June 19-21, 2019, Revised Papers*, volume 11789 of *Lecture Notes in Computer Science*, pages 364–378. Springer, 2019.

[28] Xi Chen, Martin Dyer, Leslie Ann Goldberg, Mark Jerrum, Pinyan Lu, Colin McQuillan, and David Richerby. The complexity of approximating conservative counting CSPs. *Journal of Computer and System Sciences*, 81(1):311–329, 2015.

[29] Radu Curticapean, Holger Dell, and Dániel Marx. Homomorphisms are a good basis for counting small subgraphs. In *STOC'17—Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 210–223. ACM, New York, 2017.

[30] Holger Dell. Note on "The Complexity of Counting Surjective Homomorphisms and Compactions". *arXiv preprint arXiv:1710.01712*, 2017.

[31] Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlen. Exponential time complexity of the permanent and the Tutte polynomial. *ACM Trans. Algorithms*, 10(4):21:1–21:32, 2014.

[32] Josep Díaz, Maria Serna, and Dimitrios M. Thilikos. Recent results on parameterized H-coloring. In *Graphs, morphisms and statistical physics*, volume 63 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 65–85. Amer. Math. Soc., Providence, RI, 2004.

[33] Josep Díaz, Maria J. Serna, and Dimitrios M. Thilikos. Counting H-colorings of partial k-trees. *Theor. Comput. Sci.*, 281(1-2):291–309, 2002.

[34] Reinhard Diestel. *Graph Theory, 5th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2016.

[35] PL Dobruschin. The description of a random field by means of conditional probabilities and conditions of its regularity. *Theory of Probability & Its Applications*, 13(2):197–224, 1968.

[36] Richard J Duffin. Topology of series-parallel networks. *Journal of Mathematical Analysis and Applications*, 10(2):303–318, 1965.

[37] Martin Dyer, Leslie Ann Goldberg, Catherine Greenhill, and Mark Jerrum. The relative complexity of approximate counting problems. *Algorithmica*, 38(3):471–500, 2004.

[38] Martin Dyer, Leslie Ann Goldberg, and Mark Jerrum. Counting and sampling H-colourings. *Inform. and Comput.*, 189(1):1–16, 2004.

[39] Martin Dyer, Leslie Ann Goldberg, and Mark Jerrum. An approximation trichotomy for Boolean ♯CSP. *J. Comput. System Sci.*, 76(3-4):267–277, 2010.

[40] Martin Dyer, Leslie Ann Goldberg, and Mike Paterson. On counting homomorphisms to directed acyclic graphs. *Journal of the ACM (JACM)*, 54(6):27–es, 2007.

[41] Martin Dyer and Catherine Greenhill. Random walks on combinatorial objects. In *Surveys in combinatorics, 1999 (Canterbury)*, volume 267 of *London Math. Soc. Lecture Note Ser.*, pages 101–136. Cambridge Univ. Press, Cambridge, 1999.

[42] Martin Dyer and Catherine Greenhill. The complexity of counting graph homomorphisms. *Random Structures & Algorithms*, 17(3-4):260–289, 2000.

[43] Martin Dyer and Catherine Greenhill. On Markov chains for independent sets. *Journal of Algorithms*, 35(1):17–49, 2000.

[44] Martin Dyer and Catherine Greenhill. Corrigendum: The complexity of counting graph homomorphisms. *Random Structures & Algorithms*, 25(3):346–352, 2004.

[45] Martin Dyer and David Richerby. An effective dichotomy for the counting constraint satisfaction problem. *SIAM Journal on Computing*, 42(3):1245–1274, 2013.

[46] John Faben. The complexity of counting solutions to generalised satisfiability problems modulo k. *arXiv preprint arXiv:0809.1836*, 2008.

[47] John Faben. *The Complexity of Modular Counting in Constraint Satisfaction Problems.* PhD thesis, Queen Mary University of London, 2012.

[48] John Faben and Mark Jerrum. The complexity of parity graph homomorphism: An initial investigation. *Theory of Computing*, 11(2):35–57, 2015.

[49] Tomás Feder and Pavol Hell. List homomorphisms to reflexive graphs. *J. Combin. Theory Ser. B*, 72(2):236–250, 1998.

[50] Tomas Feder, Pavol Hell, and Jing Huang. List homomorphisms and circular arc graphs. *Combinatorica*, 19(4):487–505, 1999.

[51] Tomas Feder, Pavol Hell, and Jing Huang. Extension problems with degree bounds. *Discrete Appl. Math.*, 157(7):1592–1599, 2009.

[52] Tomás Feder, Pavol Hell, Peter Jonsson, Andrei Krokhin, and Gustav Nordh. Retractions to pseudoforests. *SIAM Journal on Discrete Mathematics*, 24(1):101–112, 2010.

[53] Tomás Feder and Moshe Y Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[54] Jiří Fiala and Jan Kratochvíl. Locally constrained graph homomorphisms—structure, complexity, and applications. *Computer Science Review*, 2(2):97–111, 2008.

[55] Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. *SIAM J. Comput.*, 33(4):892–922, 2004.

[56] Jacob Focke, Leslie Ann Goldberg, Marc Roth, and Stanislav Živný. Counting homomorphisms to K_4-minor-free graphs, modulo 2. *arXiv preprint arXiv:2006.16632*, 2020.

[57] Jacob Focke, Leslie Ann Goldberg, and Stanislav Živný. The complexity of counting surjective homomorphisms and compactions. *SIAM Journal on Discrete Mathematics*, 33(2):1006–1043, 2019.

[58] Jacob Focke, Leslie Ann Goldberg, and Stanislav Živný. The complexity of approximately counting retractions. *ACM Transactions of Computation Theory*, 12(3):Art. 15, 43, 2020.

[59] Jacob Focke, Leslie Ann Goldberg, and Stanislav Živný. The complexity of approximately counting retractions to square-free graphs. *arXiv preprint arXiv:1907.02319*, 2019.

[60] Peter Fulla, Hannes Uppman, and Stanislav Živný. The complexity of Boolean surjective general-valued CSPs. *ACM Transactions on Computation Theory (TOCT)*, 11(1):1–31, 2018.

[61] Andreas Galanis, Leslie Ann Goldberg, and Andrés Herrera-Poyatos. The complexity of approximating the complex-valued Potts model. *arXiv preprint arXiv:2005.01076*, 2020.

[62] Andreas Galanis, Leslie Ann Goldberg, and Mark Jerrum. Approximately counting H-colorings is #BIS-hard. *SIAM J. Comput.*, 45(3):680–711, 2016.

[63] Andreas Galanis, Leslie Ann Goldberg, and Mark Jerrum. A complexity trichotomy for approximately counting list H-colorings. *ACM Trans. Comput. Theory*, 9(2):Art. 9, 22, 2017.

[64] Andreas Galanis, Leslie Ann Goldberg, and James Stewart. Fast algorithms for general spin systems on bipartite expanders. *arXiv preprint arXiv:2004.13442*, 2020.

[65] Andreas Galanis, Daniel Štefankovič, and Eric Vigoda. Inapproximability of the partition function for the antiferromagnetic Ising and hard-core models. *Combinatorics, Probability and Computing*, 25(4):500–559, 2016.

[66] Andreas Galanis, Daniel Štefankovič, Eric Vigoda, and Linji Yang. Ferromagnetic Potts model: Refined #BIS-hardness and related results. *SIAM J. Comput.*, 45(6):2004–2065, 2016.

[67] Hans-Otto Georgii. *Gibbs measures and phase transitions*, volume 9. Walter de Gruyter, 2011.

[68] Andreas Göbel, Leslie Ann Goldberg, and David Richerby. The complexity of counting homomorphisms to cactus graphs modulo 2. *ACM Trans. Comput. Theory*, 6(4):17:1–17:29, 2014.

[69] Andreas Göbel, Leslie Ann Goldberg, and David Richerby. Counting homomorphisms to square-free graphs, modulo 2. *ACM Trans. Comput. Theory*, 8(3):Art. 12, 29, 2016.

[70] Andreas Göbel, J. A. Gregor Lagodzinski, and Karen Seidel. Counting homomorphisms to trees modulo a prime. In Igor Potapov, Paul G. Spirakis, and James Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, volume 117 of *LIPIcs*, pages 49:1–49:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[71] Andreas Göbel (Andreas Gkompel Magkakis). *Counting, modular counting and graph homomorphisms*. PhD thesis, University of Oxford, 2016.

[72] Leslie Ann Goldberg, Martin Grohe, Mark Jerrum, and Marc Thurley. A complexity dichotomy for partition functions with mixed signs. *SIAM Journal on Computing*, 39(7):3336–3402, 2010.

[73] Leslie Ann Goldberg and Mark Jerrum. The complexity of ferromagnetic Ising with local fields. *Combinatorics, Probability and Computing*, 16(1):43–61, 2007.

[74] Leslie Ann Goldberg and Mark Jerrum. Approximating the partition function of the ferromagnetic Potts model. *Journal of the ACM (JACM)*, 59(5):1–31, 2012.

[75] Leslie Ann Goldberg and Mark Jerrum. Approximating the Tutte polynomial of a binary matroid and other related combinatorial polynomials. *Journal of Computer and System Sciences*, 79(1):68–78, 2013.

[76] Leslie Ann Goldberg and Mark Jerrum. The complexity of approximately counting tree homomorphisms. *ACM Trans. Comput. Theory*, 6(2):Art. 8, 31, 2014.

[77] Leslie Ann Goldberg, Mark Jerrum, and Mike Paterson. The computational complexity of two-state spin systems. *Random Structures Algorithms*, 23(2):133–154, 2003.

[78] Leslie Ann Goldberg, Steven Kelk, and Mike Paterson. The complexity of choosing an H-coloring (nearly) uniformly at random. *SIAM J. Comput.*, 33(2):416–432, 2004.

[79] L. M. Goldschlager and I. Parberry. On the construction of parallel computers from various bases of Boolean functions. *Theor. Comput. Sci.*, 43:43–58, 1986.

[80] Petr A. Golovach, Matthew Johnson, Barnaby Martin, Daniël Paulusma, and Anthony Stewart. Surjective H-colouring: New hardness results. *Computability*, 8(1):27–42, 2019.

[81] Petr A. Golovach, Bernard Lidický, Barnaby Martin, and Daniël Paulusma. Finding vertex-surjective graph homomorphisms. *Acta Inform.*, 49(6):381–394, 2012.

[82] Petr A. Golovach, Daniël Paulusma, and Jian Song. Computing vertex-surjective homomorphisms to partially reflexive trees. *Theoret. Comput. Sci.*, 457:86–100, 2012.

[83] Martin Charles Golumbic and Clinton F. Goss. Perfect elimination and chordal bipartite graphs. *Journal of Graph Theory*, 2(2):155–163, 1978.

[84] Artem Govorov, Jin-Yi Cai, and Martin Dyer. A dichotomy for bounded degree graph homomorphisms with nonnegative weights. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 66:1–66:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[85] Carla Groenland, Karolina Okrasa, Pawel Rzazewski, Alex Scott, Paul Seymour, and Sophie Spirkl. H-colouring P_t-free graphs in subexponential time. *Discrete Applied Mathematics*, 267:184–189, 2019.

[86] Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM (JACM)*, 54(1):1–24, 2007.

[87] Martin Grohe and Marc Thurley. Counting homomorphisms and partition functions. *Model Theoretic Methods in Finite Combinatorics*, 558:243–292, 2011.

[88] Wolfgang Gutjahr, Emo Welzl, and Gerhard Woeginger. Polynomial graph-colorings. *Discrete Applied Mathematics*, 35(1):29–45, 1992.

[89] Olle Häggström. Random-cluster analysis of a class of binary lattice gases. *Journal of statistical physics*, 91(1-2):47–74, 1998.

[90] Frank Harary. *Graph theory.* Addison-Wesley Publishing Co., Reading, Mass.-Menlo Park, Calif.-London, 1969.

[91] Frank Harary and Allen Schwenk. Trees with Hamiltonian square. *Mathematika*, 18:138–140, 1971.

[92] P. Hell and D. J. Miller. Graphs with forbidden homomorphic images. *Annals of the New York Academy of Sciences*, 319(1):270–280, 1979.

[93] P. Hell and J. Nešetřil. Homomorphisms of graphs and of their orientations. *Monatsh. Math.*, 85(1):39–48, 1978.

[94] Pavol Hell. *Rétractions de graphes.* PhD thesis, Université de Montréal, 1972.

[95] Pavol Hell. Absolute retracts in graphs. In *Graphs and Combinatorics*, pages 291–301. Springer, 1974.

[96] Pavol Hell and Jaroslav Nešetřil. On the complexity of H-coloring. *J. Combin. Theory Ser. B*, 48(1):92–110, 1990.

[97] Pavol Hell and Jaroslav Nešetřil. Counting list homomorphisms for graphs with bounded degrees. In *Graphs, morphisms and statistical physics*, volume 63 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 105–112. Amer. Math. Soc., Providence, RI, 2004.

[98] Pavol Hell and Jaroslav Nešetřil. *Graphs and Homomorphisms*, volume 28 of *Oxford Lecture Series in Mathematics and its Applications.* Oxford University Press, Oxford, 2004.

[99] Pavol Hell and Jaroslav Nešetřil. Colouring, constraint satisfaction, and complexity. *Computer Science Review*, 2(3):143–163, 2008.

[100] Pavol Hell and Ivan Rival. Absolute retracts and varieties of reflexive graphs. *Canad. J. Math.*, 39(3):544–567, 1987.

[101] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[102] Ernst Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925.

[103] Klaus Jansen and Petra Scheffler. Generalized coloring for tree-like graphs. *Discrete Appl. Math.*, 75(2):135–155, 1997.

[104] Mark Jerrum. A very simple algorithm for estimating the number of k-colorings of a low-degree graph. *Random Structures & Algorithms*, 7(2):157–165, 1995.

[105] Mark R. Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoret. Comput. Sci.*, 43(2-3):169–188, 1986.

[106] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.

[107] Amirhossein Kazeminia and Andrei A. Bulatov. Counting homomorphisms modulo a prime number. In *44th International Symposium on Mathematical Foundations of Computer Science*, volume 138 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 59, 13. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019.

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

[109] Ekkehard G. Köhler. *Graphs without asteroidal triples*. PhD thesis, Technische Universität Berlin, 1999.

[110] Phokion G. Kolaitis and Moshe Y. Vardi. Conjunctive-query containment and constraint satisfaction. *J. Comput. Syst. Sci.*, 61(2):302–332, 2000.

[111] Jan Kratochvíl and András Sebő. Coloring precolored perfect graphs. *J. Graph Theory*, 25(3):207–215, 1997.

[112] Richard E Ladner. On the structure of polynomial time reducibility. *Journal of the ACM (JACM)*, 22(1):155–171, 1975.

[113] Benoît Larose. Algebra and the complexity of digraph CSPs: a survey. In *The constraint satisfaction problem: complexity and approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 267–285. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2017.

[114] Benoit Larose, Barnaby Martin, and Daniël Paulusma. Surjective H-colouring over reflexive digraphs. *ACM Transactions on Computation Theory (TOCT)*, 11(1):1–21, 2018.

[115] R. M. R. Lewis. *A Guide to Graph Colouring - Algorithms and Applications*. Springer, 2016.

[116] L. Lovász. Operations with structures. *Acta Math. Acad. Sci. Hungar.*, 18:321–328, 1967.

[117] László Lovász. Graph minor theory. *Bull. Amer. Math. Soc. (N.S.)*, 43(1):75–86, 2006.

[118] László Lovász. *Large Networks and Graph Limits*, volume 60 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Providence, RI, 2012.

[119] Barnaby Martin and Daniël Paulusma. The computational complexity of disconnected cut and $2K_2$-partition. *J. Combin. Theory Ser. B*, 111:17–37, 2015.

[120] Dániel Marx. Graph colouring problems and their applications in scheduling. *Periodica Polytechnica Electrical Engineering*, 48(1-2):11–16, 2004.

[121] Dániel Marx. Parameterized coloring problems on chordal graphs. *Theoret. Comput. Sci.*, 351(3):407–424, 2006.

[122] Gregor Matl and Stanislav Živný. Using a min-cut generalisation to go beyond Boolean surjective VCSPs. *Algorithmica*, pages 1–29, 2020.

[123] Michael Mitzenmacher and Eli Upfal. *Probability and Computing*. Cambridge University Press, Cambridge, second edition, 2017.

[124] Reza Naserasr, Edita Rollová, and Éric Sopena. Homomorphisms of signed graphs. *Journal of Graph Theory*, 79(3):178–212, 2015.

[125] Committee on Mathematical Challenges from Computational Chemistry et al. *Mathematical challenges from theoretical/computational chemistry*. National Academies Press, 1995.

[126] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley Publishing Company, Reading, MA, 1994.

[127] Christos H Papadimitriou and Stathis K Zachos. Two remarks on the power of counting. In *Theoretical Computer Science*, pages 269–275. Springer, 1982.

[128] Viresh Patel and Guus Regts. Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials. *SIAM Journal on Computing*, 46(6):1893–1919, 2017.

[129] Erwin Pesch. *Retracts of graphs*, volume 110 of *Mathematical Systems in Economics*. Athenäum Verlag GmbH, Frankfurt am Main, 1988.

[130] R. B. Potts. Some generalized order-disorder transformations. *Proc. Cambridge Philos. Soc.*, 48:106–109, 1952.

[131] Wolfgang M. Schmidt. *Diophantine approximations and Diophantine equations*, volume 1467 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1991.

[132] Claus-Peter Schnorr. Optimal algorithms for self-reducible problems. In S. Michaelson and Robin Milner, editors, *Third International Colloquium on Automata, Languages and Programming, University of Edinburgh, UK, July 20-23, 1976*, pages 322–337. Edinburgh University Press, 1976.

[133] Uwe Schöning. A uniform approach to obtain diagonal sets in complexity classes. *Theoretical Computer Science*, 18(1):95 – 103, 1982.

[134] Alistair Sinclair, Piyush Srivastava, and Marc Thurley. Approximation algorithms for two-state anti-ferromagnetic spin systems on bounded degree graphs. *Journal of Statistical Physics*, 155(4):666–686, 2014.

[135] Steven Skiena. *The Algorithm Design Manual, Second Edition.* Springer, 2008.

[136] Allan Sly. Computational transition at the uniqueness threshold. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 287–296. IEEE, 2010.

[137] Allan Sly and Nike Sun. Counting in two-spin models on d-regular graphs. *Annals of Probability*, 42(6):2383–2416, 2014.

[138] Marc Thurley. *The complexity of partition functions.* PhD thesis, Humboldt-Universität zu Berlin, 2009.

[139] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.

[140] Zsolt Tuza. Graph colorings with local constraints—a survey. *Discuss. Math. Graph Theory*, 17(2):161–228, 1997.

[141] Leslie G Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[142] Leslie G Valiant. Accidental algorthims. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 509–517. IEEE, 2006.

[143] Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(3):85–93, 1986.

[144] Narayan Vikas. Computational complexity of compaction to reflexive cycles. *SIAM Journal on Computing*, 32(1):253–280, 2002.

[145] Narayan Vikas. Compaction, retraction, and constraint satisfaction. *SIAM J. Comput.*, 33(4):761–782, 2004.

[146] Narayan Vikas. A complete and equal computational complexity classification of compaction and retraction to all graphs with at most four vertices and some general results. *J. Comput. System Sci.*, 71(4):406–439, 2005.

[147] Narayan Vikas. Algorithms for partition of some class of graphs under compaction and vertex-compaction. *Algorithmica*, 67(2):180–206, 2013.

[148] Narayan Vikas. Computational complexity relationship between compaction, vertex-compaction, and retraction. In *Combinatorial algorithms*, volume 10765 of *Lecture Notes in Comput. Sci.*, pages 154–166. Springer, Cham, 2018.

[149] Dror Weitz. Counting independent sets up to the tree threshold. In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, STOC 2006, Seattle, WA, USA, May 21-23, 2006*, pages 140–149. ACM, 2006.

[150] Benjamin Widom and John S. Rowlinson. New model for the study of liquid–vapor phase transitions. *The Journal of Chemical Physics*, 52(4):1670–1684, 1970.

[151] Dmitriy Zhuk. No-rainbow problem is NP-hard. *arXiv preprint arXiv:2003.11764*, 2020.

[152] Dmitriy Zhuk. A proof of the CSP dichotomy conjecture. *Journal of the ACM (JACM)*, 67(5):1–78, 2020.

[153] David Zuckerman. On unapproximable versions of NP-complete problems. *SIAM Journal on Computing*, 25(6):1293–1304, 1996.