Approximate counting via complex zero-free regions and spectral independence

Andrés Herrera poyatos

Balliol College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2023

To my parents and my brothers.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Leslie Goldberg and Andreas Galanis, for introducing me to the fascinating field of approximate counting and for their invaluable time, guidance and expertise. Working under the supervision of such exceptionally talented researchers has been an enlightening opportunity that I am truly grateful for.

I would also like to extend my thanks to Stefan Kiefer and Standa Živný for serving as examiners for my various DPhil examinations, and their excellent questions and feedback during those. Additionally, I would like to thank in advance my DPhil viva examiners for taking the time to review this thesis.

On a personal note, I am deeply grateful to my office mates for creating an uplifting and stimulating environment in our shared space. A special thanks goes to Marc, with whom I have shared countless of maths discussions, coffee breaks and board games nights during the four years that we have been office mates. I would also like to acknowledge the incredible friends that I have made during my time in Oxford, with a special thanks to my Balliol friends for their constant support, particularly as housemates during the pandemic.

Last but not the least, I would like to express my deepest gratitude to my parents, for their never-ending support and instilling in me their passion for mathematics, and to my brothers, David and Sergio. Your presence in my life has been a constant source of inspiration and encouragement and I have missed you dearly since I moved to Oxford.

Abstract

This thesis investigates fundamental problems in approximate counting that arise in the field of statistical mechanics. Building upon recent advancements in the area, our research aims to enhance our understanding of the computational complexity of sampling from the Ising and Potts models, as well as the random k-SAT model.

The q-state Potts model is a spin model in which each particle is randomly assigned a spin (out of q possible spins), where the probability of a certain assignment depends on how many adjacent particles present the same spin. The edge interaction of the model is a parameter that quantifies the strength of interaction between two adjacent particles. The Ising model corresponds to the Potts model with $q = 2$. Sampling from these models is inherently connected to approximating the partition function of the model, a graph polynomial that encodes several aggregate thermodynamic properties of the system. In addition to classical connections with quantum computing and phase transitions in statistical physics, recent work in approximate counting has shown that the behaviour in the complex plane of these partition functions, and more precisely the location of zeros, is strongly connected with the complexity of the approximation problem, even for positive real-valued parameters. Thus, following this trend in both statistical physics and algorithmic research, we allow the edge interaction to be any complex number.

First, we study the complexity of approximating the partition function of the q-state Potts model and the closely related Tutte polynomial for complex values of the underlying parameters. Previous work in the complex plane by Goldberg and Guo focused on $q = 2$; for $q > 2$, the behaviour in the complex plane is not as well understood and most work applies only to the real-valued Tutte plane. Our main result is a complete classification of the complexity of the approximation problems for all non-real values of the parameters, by establishing #P-hardness results that apply even when restricted to planar graphs. Our techniques apply to all $q \geq 2$ and further complement/refine previous results both for the Ising model and the Tutte plane, answering in particular a question raised by Bordewich, Freedman, Lovász and Welsh in the context of quantum computations.

Secondly, we investigate the complexity of approximating the partition function $Z_{\text{Ising}}(G;\beta)$ of the Ising model in terms of the relation between the edge interaction β and a parameter Δ which is an upper bound on the maximum degree of the input graph G. In this thesis we establish both new tractability and inapproximability results. Our tractability results show that $Z_{\text{Ising}}(-;\beta)$ has an FPTAS when $\beta \in \mathbb{C}$ and $|\beta - 1|/|\beta + 1| < \tan(\pi/(4\Delta - 4))$. The core of the proof is showing that there are no inputs G that make the partition function 0 when β is in this range. Our result significantly extends the known zero-free region of the Ising model (and hence the known approximation results). Our intractability results show that it is #P-hard to approximate $Z_{\text{Ising}}(-;\beta)$ when $\beta \in \mathbb{C}$ is an algebraic number such that $\beta \notin \mathbb{R} \cup \{i, -i\}$ and $|\beta - 1|/|\beta + 1| > 1/\sqrt{\Delta - 1}$. These are the first results to show intractability of approximating

$Z_{\text{Ising}}(-,\beta)$ on bounded degree graphs with complex β. Moreover, we demonstrate situations in which zeros of the partition function imply hardness of approximation in the Ising model.

Finally, we exploit the recently successful framework of spectral independence to analyse the mixing time of a Markov chain, and we apply it in order to sample satisfying assignments of k-CNF formulas. Our analysis leads to a nearly linear-time algorithm to approximately sample satisfying assignments in the random k-SAT model when the density of the random formula $\alpha = m/n$ scales exponentially with k, where n is the number of variables and m is the number of clauses. The best previously known sampling algorithm for the random k-SAT model applies when the density $\alpha = m/n$ of the formula is less than $2^{k/300}$ and runs in time $n^{\exp(\Theta(k))}$. Our algorithm achieves a significantly faster running time of $n^{1+o_k(1)}$ and samples satisfying assignments up to density $\alpha \leq 2^{0.039k}$. The main challenge in our setting is the presence of many variables with unbounded degree, which causes significant correlations within the formula and impedes the application of relevant Markov chain methods from the bounded-degree setting.

Declaration of authorship

If not explicitly stated otherwise, all the results presented in this thesis are new contributions. 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 journals. Chapters 2 and 3 are based on the following papers, which are co-authored with my supervisors Andreas Galanis and Leslie Ann Goldberg:

[51] Andreas Galanis, Leslie Ann Goldberg, and Andrés Herrera-Poyatos. The complexity of approximating the complex-valued potts model. *Comput. Complexity*, 31(1):Paper No. 2, 2022. doi:10.1007/s00037-021-00218-x.
∘ A preliminary version of this work appeared in MFCS: Andreas Galanis, Leslie Ann Goldberg, and Andrés Herrera-Poyatos. The complexity of approximating the complex-valued potts model. In *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 36:1–36:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2020.36.

[47] Andreas Galanis, Leslie A. Goldberg, and Andres Herrera-Poyatos. The complexity of approximating the complex-valued Ising model on bounded degree graphs. *SIAM J. Discrete Math.*, 36(3):2159–2204, 2022. doi:10.1137/21M1454043.

Chapter 4 is based on an updated version of the following work, which is co-authored with my suvervisors and Heng Guo.

[48] Andreas Galanis, Leslie Ann Goldberg, Heng Guo, and Andrés Herrera-Poyatos. Fast sampling of satisfying assignments from random k-sat. *arXiv preprint*, 2022. arXiv:2206.15308.

The version of [48] presented in this thesis includes results on the geometry of the space of satisfying assignments of random k-CNF formulas, that have been developed in conjunction with Zongchen Chen, Nitya Mani and Ankur Moitra. The proofs of these geometry results presented here are my own. This extended version of [48] has been submitted to Random Structures and Algorithms.

Contents

List of definitions and notation

Here we gather the notation and definitions that are used repeatedly in each chapter of this thesis. If some notation or definition is not here, then it is only used in one specific section of a chapter (and it is defined in that section).

Chapter 1

Notation	Description	Reference
$\#\mathrm{P}$	The class of counting problems	Page 12
hom	The graph homomorphism partition function	Eq. (1.3), page 15
$Z_{\mathcal{I}}$	The independent set polynomial	Page 15
$\mathrm{size}(G)$	The size of a graph G	Page 15
Z_{Potts}	The Potts model partition function	Eq. (1.4), page 15
Z_{Ising}	The Ising model partition function	Page 16
Z_{Tutte}	The Tutte polynomial	Eq. (1.5), page 16
$\Phi(k,n,m)$	A random k-CNF formula with n var. and m clauses	Page 18
α	The density of the formula Φ, so $\alpha = m/n$	Page 18
FPRAS	Fully-polynomial randomised approximation scheme	Page 19
FPTAS	Fully-polynomial deterministic approximation scheme	Page 19
arg	The principal argument of a complex number	Page 22
Arg	The set of arguments of a complex number	Page 22
ε_Δ	The number $\tan(\pi/(4(\Delta-1)))$	Theorem 1.5
δ_Δ	The number $\max\left\{\sin\left(\frac{\alpha}{2}\right)\cos\left(\Delta\frac{\alpha}{2}\right) : 0 < \alpha < \frac{2\pi}{3\Delta}\right\}$	Page 26
h_β	The Mobius function $h_\beta(z) = (\beta z+1)/(\beta+z)$	Page 29
w.h.p.	Stands for "with high probability"	Page 31
d_{TV}	The total variation distance between two distributions	Page 31
$\|\Lambda\|_1$	Hamming weight	Definition 1.9
D-connectivity	Connectivity of assignments of a k-CNF formula	Definition 1.9
$f(n)$-loose	Looseness in a k-CNF formula	Definition 1.11

Computational problems

Notation	Description	Reference
Factor-K-NormPotts(q,y)	Norm approx. problem for Potts	Page 22
Factor-K-NormPlanarPotts(q,y)	Same restricted to planar graphs	Page 22
Distance-ρ-ArgPotts(q,y)	Argument approx. problem for Potts	Page 22

DISTANCE-ρ-ARGPLANARPOTTS(q, y)	Same restricted to planar graphs	Page 22
FACTOR-K-NORMISING(y)	Norm approx. problem for Ising	Page 22
DISTANCE-ρ-ARGISING(q, y)	Argument approx. problem for Ising .	Page 22
FACTOR-K-NORMTUTTE(q, γ)	Norm approx. problem for Tutte	Page 22
FACTOR-K-NORMPLANARTUTTE(q, γ)	Same restricted to planar graphs	Page 22
DISTANCE-ρ-ARGTUTTE(q, γ)	Argument approx. problem for Tutte	Page 22
DISTANCE-ρ-ARGPLANARTUTTE(q, γ)	Same restricted to planar graphs	Page 22
SIGNTUTTE(q, γ)	Sign problem for Tutte polynomial ..	Page 22
SIGNPLANARTUTTE(q, γ)	Same restricted to planar graphs	Page 22
ISINGNORM(β, Δ, K)	Norm approx. problem for b.d. Ising	Page 28
ISINGARG(β, Δ, ρ)	Same problem for argument	Page 28

Chapter 2

Notation	Description	Reference
theta graph	Graph with two terminals joined by paths	Definition 2.1
series-parallel graph ...	Graph constructed from series-parallel operations	Definition 2.1
$\mathsf{P}_{\mathbb{C}}$	Set of poly-time computable numbers	Page 38
$\mathsf{P}_{\mathbb{R}}$	The set $\mathsf{P}_{\mathbb{C}} \cap \mathbb{R}$	Page 38
$Z_{st}(G; q, \gamma)$	Tutte poly. for connected terminals s and t	Page 40
$Z_{s\|t}(G; q, \gamma)$	Tutte poly. for non-connected terminals s and t .	Page 40
γ-implement	See definition	Page 40
shift	See definition	Page 41
parallel composition ...	Operation between two graphs	Page 41
series composition	Operation between two graphs	Page 41
$\Theta_{(l_1,\ldots,l_m)}$	Theta graph with m paths of lengths $l_1, \ldots, l_m$ ..	Page 42
$w(\ldots)$	Weight implemented by a Theta graph	Page 42
Root of unity	Complex root of $z^k = 1$ for some k	Page 43
poly-time approx. shift	See definition	Definition 2.10

Chapter 3

Notation	Description	Reference
$\mathbb{A}$, $\mathbb{C}_{\mathbb{A}}$	The sets of real and complex algebraic numbers	Page 93
SAW tree	Tree of self-avoiding walks of a graph	Page 95
$Z_v^j(G; \beta)$	Ising model partition function with a pinning on vertex v	Definition 3.6
$R(T, v; \beta)$	The ratio $Z_v^1(T; \beta)/Z_v^0(T; \beta)$ for a tree T	Definition 3.6
$F_{\beta,k}(z_1, \ldots, z_k)$	Recursion for Ising model on trees	Page 96

$B(x,r)$	Open disk with centre x and radius r	Page 96
$\overline{B}(x,r)$	Closed disk with centre x and radius r	Page 96
$C(x,r)$	Circle with centre x and radius r	Page 96
$Z^{jk}_{st}(H;\beta)$	Ising model p.f. with two pinnings	Page 97
$I_{st}(H;\beta)$	The interaction matrix of the Ising model	Page 97
(Δ,β)-implements	See definition	Definition 3.7
$\widehat{\mathbb{C}}$	The Riemann sphere	Page 99
$d(z,w)$	Cordal metric	Page 99
Mobius map	Rational map of degree one	Page 99
multiplier	Derivative at a fixpoint	Page 99
fixpoint	Point ω with $f(\omega)=\omega$	Page 99
Julia set	See definiton	Page 99
neighbourhood of x	Set containing a ball with centre x	Page 100
exceptional point	Point with finite number of iterations on f	Page 100
$\mathcal{R}(\delta)$	Set of $z\in\mathbb{C}$ with $\lvert(z-1)/(z+1)\rvert\leq\delta$	Definition 3.15
Ising program	See the definition	Definition 3.25
g_β	The map $g_\beta(z)=h_\beta(h_\beta(z))$	Definition 3.25
Mobius program	See the definition	Definition 3.27
program-approximable	See definition	Definition 3.28
desenly program-approx.	See definition	Definition 3.29

Chapter 4

Notation	**Description**	**Reference**
$\mathcal{V}$	The set of variables of Φ	Page 140
$\mathcal{C}$	The set of clauses of Φ	Page 140
$\mathrm{var}(c)$	The variables in the clause c	Page 140
$\mathrm{var}(S)$	The union of $\mathrm{var}(c)$ over $c\in S$	Page 140
ξ	Our sampling algorithm has error at most $n^{-\xi}$	Theorem 1.8
Δ_r	The high-degree threshold, set to $\lceil 2^{(r_0-\delta)k}\rceil$	Definition 4.1
high-degree	See definition	Definition 4.1
r-distributed	See definition	Definition 4.3
$(r,r_\mathrm{m},r_\mathrm{a},r_\mathrm{c})$-marking	See definition	Definition 4.3
r_0, r_1, δ	$r_0=0.117841$, $r_1=0.227092$ and $\delta=0.00001$	Definition 4.3
$\mathcal{V}_\mathrm{m}$, $\mathcal{V}_\mathrm{a}$, $\mathcal{V}_\mathrm{c}$	The sets of marked, auxiliary and control variables	Definition 4.3
Ω^*	The set of all assignments $\mathcal{V}\to\{\mathsf{F},\mathsf{T}\}$	Definition 4.4
Ω	The set of satisfying assignments of Φ	Definition 4.4
μ_A	The uniform distribution over $A\subseteq\Omega*$	Definition 4.4
Φ^Λ	The formula Φ simplified under Λ	Definition 4.4

$\mathcal{V}^\Lambda$, $\mathcal{C}^\Lambda$	The variables and clauses of Φ^Λ	Definition 4.4
Ω^Λ	The set of satisfying assignments of Φ^Λ	Definition 4.4
$\mu\|_V$	The marginal distribution of μ on V	Definition 4.6
ε-uniform	See definition	Definition 4.7
$T_{\text{mix}}(\rho, \varepsilon)$	Mixing time of the block Glauber dynamics	Page 143
b-marginally bounded	See definition	Page 143
$\mathcal{I}^\Lambda(u \to v)$	The influence of u on v (under Λ)	Page 144
η-spectrally independent	See definition	Page 144
G_Φ	The dependency graph of $\mathcal{C}$	Definition 4.11
H_Φ	The dependency graph of $\mathcal{V}$	Definition 4.13
$\mathcal{C}_{\text{good}}(r)$, $\mathcal{C}_{\text{bad}}(r)$	Good and bad clauses, a partition of $\mathcal{C}$	Page 151
$\mathcal{V}_{\text{good}}(r)$, $\mathcal{V}_{\text{bad}}(r)$	Good and bad variables, a partition of $\mathcal{V}$	Page 151
$\Phi_{\text{good}}(r)$	The formula with all good var. and clauses of Φ	Definition 4.19
$\Phi_{\text{bad}}(r)$	The formula with all bad var. and clauses of Φ	Definition 4.19

Chapter 1

Introduction and contributions

Approximate counting is a branch of computational complexity and randomised algorithms that has seen a growing interest in the past decade due to its connections to sampling, statistical mechanics and random systems. Before delving into approximate counting, let us briefly introduce the class of *counting problems*, which can be understood as the counting analogue to NP. Formally, an exact counting problem is a computational problem of the form "compute $f(x)$", where $f(x)$ is the number of accepting paths (on input x) of a non-deterministic Turing machine that runs in polynomial time. We use #P to denote the class of all counting problems, which was introduced by Valiant [115]. As an example, a canonical counting problem is #SAT, the problem of counting the number of satisfying assignments of a boolean formula in conjunctive normal form (CNF). By the celebrated Cook-Levin theorem, any counting problem has a polynomial-time reduction to #SAT. Thus, we say that #SAT is #P-*complete*. Other relevant #P-complete problems are counting the number of independent sets of a graph or counting the number of satisfying assignments of a formula in disjunctive normal form, denoted #DNF, see, for instance, [121]. Several more #P-complete (or #P-hard) problems will come up in this thesis.

The goal of approximate counting is understanding the inherent difficulty of computing approximate answers to counting problems that would be #P-hard to solve exactly. For a precise notion of approximability, we refer to Section 1.2. A classic example of fundamental relevance in theoretical computer science is, for $k \geq 3$, the problem of approximately counting the number of satisfying assignments of a k-CNF formula (a formula in conjunctive normal form where every clause has exactly k literals), which is hard unless NP = RP. Interestingly, this contrasts to the complexity of the problem of approximating the number of satisfying assignments of a formula in disjunctive normal form (#DNF), which turns out to have an elementary polynomial-time algorithm. The complexity of several relevant approximate counting problems still remains elusive see, for instance, [39].

Some of the most famous problems in approximate counting arise in statistical mechanics. This is the case of some of the problems that we study in this thesis, including the problems of approximating the partition function of spin systems such as the hard-core, Ising and Potts models, which we will formally introduce in Section 1.1. As we will see, approximating partition functions is inherently connected to sampling from the distributions of these spin systems, which is of particular importance in statistical mechanics. Abstractly, for a finite set V of "particles" and an integer $q \geq 2$, we want to sample from a probability distribution over the *configurations* $\sigma \colon V \to [q]$, where $[q] := \{1, 2, \ldots, q\}$ is the set of possible *spins*. We highlight that this abstract setting also includes as a particular case the problem of uniformly sampling satisfying assignments

of CNF formulas. In fact, we will work in this general setting at some stages in this thesis as some of our methods exploit properties of spin distributions with local constrains. The computational complexity of approximating these partition functions turns out to be related to the physical phase transitions that these models present on bounded-degree graphs, which has drawn a lot of attention to approximate counting and has motivated a lot of work in the area. We will briefly delve into this connection in Section 1.2 of this introduction. Interestingly, these partition functions occasionally admit fully-polynomial randomised approximation algorithms, serving as examples of problems whose exact version is #P-hard but the approximate version is tractable [8]. Other times the problem of approximating a partition function can be shown to be hard, sometimes even as hard as the exact version. The study of the complexity of approximating partition functions is a growing area where new techniques to envisage efficient approximation algorithms or to show hardness are being developed.

The recent increasing interest in approximate counting has led to a remarkable progress, and a number of innovative approaches have emerged, sometimes connecting approximate counting to other seemingly unrelated areas of mathematics such as complex analysis and complex dynamics [101, 8, 15, 16], or sometimes promoting a resurgence of classic techniques such as Markov Chain Monte Carlo algorithms for spin systems via the spectral independence method [7, 28]. As a consequence of these new techniques, a lot of old open problems have started to fall. This thesis is devoted to exploiting some of these exciting approaches to further understand some of the classic problems in approximate counting, including the complexity of approximating the partition functions of the Ising and Potts model on complex parameters, as well as developing approximate counting algorithms for the number of satisfying assignments of random k-CNF formulas, and understanding the boundaries of efficient computation within these models.

The rest of this introduction is organised as follows. In Section 1.1, we further motivate the work presented in this thesis from the point of view of statistical mechanics, and introduce the approximate counting problems for which we present new results in this work. In Section 1.2 we introduce the concept of approximation schemes and briefly describe some of the most powerful techniques to obtain such algorithms or prove hardness of approximation, highlighting the connection between physical phase transitions and computational phase transitions in spin systems. In Section 1.3 we state our results on the Potts model on complex parameters, where we give a complete map of approximability of the partition function. In Section 1.4 we state our results on the Ising model on bounded degree graphs, including novel tractability and inapproximability results. In Section 1.5 we state our results on random k-CNF formulas, including the first almost-uniform sampler of satisfying assignments based on spectral independence arguments. Finally, in Section 1.6 we present the organisation of the rest of this thesis.

1.1 Partition functions: from statistical mechanics to approximate counting

Partition functions arise naturally in physics, mathematics and computer science, and their study has revealed connections among all these disciplines. The exact definition of partition function varies depending on the area of study; in approximate counting we are interested in partition functions from a combinatorial perspective. Before introducing the combinatorial definition of the partition functions that we consider in this thesis, let us motivate this concept from the lenses of statistical mechanics.

In statistical mechanics of discrete systems we model particles as vertices of an undirected graph $G = (V, E)$ (possibly with loops or multiple edges between the same vertices) that captures the interactions among them. These particles may correspond, for example, to magnets, the molecules of a gas, or atoms in spin glasses. Each particle is in one out of q possible states/spins, where $q \geq 2$. For instance, in the case of ferromagnetic systems, these states corresponds to plus or minus ($q = 2$). A *configuration* is a map $\sigma \colon V \to [q]$, assigning each particle to a spin. Let Ω be the set of all configurations. The *Gibbs distribution* of the system is a probability distribution over Ω that assigns to each state a probability that is a function of that state's energy, denoted $H(\sigma)$, and the temperature of the system T. We also call $H(\sigma)$ the *Hamiltonian* of the state, and we will define it explicitly later in this section. More precisely, the probability of a configuration σ under the Gibbs distribution, denoted $\mu(\sigma)$, is proportional to $\exp(-H(\sigma)/(cT))$, where T is the temperature of the system and c is the Boltzmann constant. The *partition function* of the system is the normalising factor of the Gibbs distribution, that is,

$$Z(G) = \sum_{\sigma \in \Omega} \exp(-H(\sigma)/(cT)). \tag{1.1}$$

The Hamiltonian/energy of a configuration depends on two interactions: the interaction energies between adjacent particles based on their spins, and the interaction between the system particles and an external field, which acts on each particle based on the particle's spin. The interaction energies are represented by a symmetric $q \times q$ matrix K, so the entry $K_{i,j} \in \mathbb{R}$ measures the interaction energy between the spins i and j. The external field has associated a vector M, whose entry $M_i \in \mathbb{R}$ indicates how the field acts on spin i. The overall energy of a configuration σ is then $H(\sigma) = \sum_{\{u,v\} \in E} K_{\sigma(u),\sigma(v)} + \sum_{v \in V} M_{\sigma(v)}$.

This definition of the Gibbs distribution arises as the solution of an optimisation problem; the Gibbs distribution is the distribution that maximises the entropy of the system, subject to certain normalisation constraints. For more information about the Gibbs distribution from a statistical mechanics point of view we refer to [53]. Some relevant aggregate thermodynamic variables of the system, such as the entropy, total energy or free energy, can be expressed in terms of the partition function and its derivatives. Moreover, if we can approximate the partition function $Z(G)$ efficiently, we can approximate efficiently any probability $\mu(\sigma)$ and, as a consequence, we can strongly simulate the system. Thus, the problems of computing and approximating partition functions are of particular relevance in statistical mechanics.

By setting $Y_{i,j} = \exp(-K_{i,j}/(cT))$ and $Z_j = \exp(-M_j/(cT))$, from (1.1) we find that

$$Z(G) = \sum_{\sigma\colon V\to[q]} \prod_{\{u,v\}\in E} Y_{\sigma(u)\sigma(v)} \prod_{v\in V} Z_{\sigma(v)}. \tag{1.2}$$

Even though in our derivation $Y_{i,j}, Z_j$ are positive numbers, generally the partition function $Z(G)$ can be seen as a multivariate graph polynomial with variables Y_{ij} and Z_j, which may take complex values. In fact, (1.2) comprises many well-known graph polynomials as particular cases. For example, when Z is a vector of ones, we obtain the *graph homomorphism partition function* [8, Chapter 7],

$$\hom(G;Y) = \sum_{\sigma\colon V\to[q]} \prod_{\{u,v\}\in E} Y_{\sigma(u)\sigma(v)}, \tag{1.3}$$

which is of particular interest in theoretical computer science, see, for instance, [45, 40]. Another relevant example is the case when $q = 2$ and, for $\lambda \in \mathbb{C}$,

$$Y = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} \lambda \\ 1 \end{bmatrix}.$$

This instance is famously known as the *independent set polynomial* or the partition function of the hard-core model [15], and is denoted $Z_{\mathcal{I}}(G;\lambda)$. One can easily find that

$$Z_{\mathcal{I}}(G;\lambda) = \sum_{I\in\mathcal{I}(G)} \lambda^{|I|},$$

where $\mathcal{I}(G)$ is the set of all the independent sets of a graph G. The parameter λ is known as the *activity* of the model. Note that $Z_{\mathcal{I}}(G;1)$ counts the number of independent sets of the graph G, whose computation is known to be #P-hard. Hence, computing the polynomial $Z_{\mathcal{I}}(G;\lambda)$ is #P-hard. Under typical circumstances this is the case with partition functions, it is unrealistic to write Z as a sum of monomials in a polynomial number of computations on $\text{size}(G) := |V(G)| + |E(G)|$.

1.1.1 The Potts model and the Tutte polynomial

The q-state Potts model is a classic model of ferromagnetism in statistical mechanics [105, 121] that can be seen as a particular example of the graph homomorphism partition function. From a physics point of view, this model arises when the energy between interacting particles behaves as follows: for a constant $\theta > 0$, the energy between two adjacent particles is θ if the particles' spins are the same, and the energy is $-\theta$ if they are different. It is not difficult to check that under our combinatorial notation, following a derivation along the lines of (1.1) and letting $y = \exp(-2\theta/(cT))$, this model is equivalent to the case when Z is a vector of ones and the matrix Y is such that $Y_{ii} = y$ and $Y_{ij} = 1$ for every i, j with $i \neq j$, see [121] for details. We denote by $Z_{\text{Potts}}(G;q,y)$ the partition function of the Potts model with parameter $y \in \mathbb{C}$, and we find that

$$Z_{\text{Potts}}(G;q,y) = \sum_{\sigma\colon V\to[q]} y^{m(\sigma)}, \tag{1.4}$$

where $m(\sigma)$ denotes the number of monochromatic edges of G under σ. The parameter y is known as the *edge interaction*; note that the Gibbs distribution of the model is only defined when y is a positive real. This partition function includes several relevant combinatorial quantities as a particular case. For example, $Z_{\text{Potts}}(G; q, 0)$ counts the number of proper q-colourings of the graph G (recall that these are the q-colourings such that two adjacent vertices have distinct colours), that is, $Z_{\text{Potts}}(G; q, 0)$ is the evaluation of the chromatic polynomial of G on a positive integer q. We refer to [121] for more connections between the chromatic polynomial and the Potts model. The case $q = 2$ is commonly known as the *Ising model*, and we write $Z_{\text{Ising}}(G; y)$ to denote its partition function.

The Ising/Potts models have an extremely useful generalisation to non-integer values of q via the so-called "random-cluster" formulation of the Tutte polynomial. For complex numbers q and γ, the Tutte polynomial of a graph $G = (V, E)$ is given by

$$Z_{\text{Tutte}}(G; q, \gamma) = \sum_{A \subseteq E} q^{k(A)} \gamma^{|A|}, \tag{1.5}$$

where $k(A)$ denotes the number of connected components in the graph (V, A) (isolated vertices do count). When q is an integer with $q \geq 2$, we have $Z_{\text{Potts}}(G; q, y) = Z_{\text{Tutte}}(G; q, y - 1)$, see, for instance, [109]. From a computational point of view, the Tutte polynomial encompasses other relevant combinatorial quantities, such as the number of nowhere-zero q-flows of G (which corresponds to $Z_{\text{Tutte}}(G; q, -q)$ up to an easily computable factor) or the number of spanning subgraphs of G, that is, the number of subgraphs (V, A) of G with $k(A) = k(E)$ (which coincides with the limit $\lim_{q \to 0} Z_{\text{Tutte}}(G; q, 1)/q^{k(E)}$). The Tutte polynomial on planar graphs is particularly relevant in quantum computing since it corresponds to the Jones polynomial of an "alternating link" [121, Chapter 5], and polynomial-time quantum computation can be simulated by additively approximating the Jones polynomial at certain roots of unity. This connection between the Tutte polynomial and quantum computation will be relevant in Section 1.3 and Chapter 2, see also [20] for details.

1.1.2 Hardness of exact computation of partition functions

Due to the combinatorial relevance of the Tutte polynomial and its connections to fundamental problems in theoretical computer science, computational complexity questions involving the Tutte polynomial have been the focus of a long series of publications, see [109] for a state of the art. For example, an interesting question that will come up in Chapter 2 of this thesis is the problem of determining the sign of $Z_{\text{Tutte}}(G; q, \gamma)$ [59]. From the point of view of counting complexity, one of the most natural questions is the following one: for a fixed pair $(q, \gamma) \in \mathbb{C}^2$, how hard is it to evaluate $Z_{\text{Tutte}}(G; q, \gamma)$ at (q, γ) and an input graph G? This question was addressed by Jaeger, Vertigan and Welsh [73], concluding that the evaluation problem is #P-hard for almost all pairs (q, γ), see Section 2.5.3 for the precise result. For example, evaluating $Z_{\text{Ising}}(G; y)$ is #P-hard except when $y \in \{0, \pm 1, \pm i\}$, where evaluation can be performed in polynomial time. This kind of dichotomy also applies to the Potts model, where a few more exceptions / easy

points arise. Similar results exist for the graph homomorphism partition function, although the reductions are significantly more convoluted. This led researchers to focus on real parameters first [40], and it took a lot of work to fully resolve the complexity of evaluation on complex parameters, see [27]. Further questions arise, such as if these hardness results hold for certain relevant families of graphs, for instance, planar graphs [116], or for certain modifications of the partition function, see, for example, [45]. When it comes to approximate counting, the complexity map for these partition functions is, at the time of writing, not fully resolved. In this thesis we make significant progress on the Ising and Potts model.

1.1.3 Constraint satisfaction problems, statistical mechanics and the random k-SAT model

We finish this section by highlighting the connections between k-SAT and statistical mechanics, which have been exploited several times in the specialised literature to obtain predictions about phase transitions in satisfiability problems [94]. To introduce this connection, first we consider the concept of constraint satisfaction problem. Let $V = \{v_1, \dots, v_n\}$ denote a collection of variables and let $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$ be the set of the respective domains. A constraint C is a pair (t, ρ) where t is a tuple of variables, called the constraint scope, and ρ is a relation on their corresponding domains, called the constraint relation. A constraint satisfaction problem (CSP) is a triple $(V, \mathcal{D}, \mathcal{C})$, where V and $\mathcal{D}$ are as above and $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$ is set of constraints. A solution to $(V, \mathcal{D}, \mathcal{C})$ is a mapping that assigns to each variable v_i a value from D_i so that the mapping satisfies all constraints, that is, for each constraint C_j, the image of the constraint scope is a member of the constraint relation. For example, we can see the problem of finding a proper q-colouring as a CSP; for a graph $G = (V, E)$, let the domain of the variables be $[q]$ and consider for each edge $(u, v) \in E$ the constraint $\{(i, j) : i, j \in [q], i \neq j\}$. Then a q-colouring is a solution of the CSP if and only if it is a proper q-colouring. Another obvious example is k-SAT, where each constraint of the CSP is determined by a clause of the k-CNF formula.

We remark that the derivation of partition function given in (1.1) can be reproduced in the setting of CSPs with a different definition of Hamiltonian. Let $\Phi = (V, \mathcal{D}, \mathcal{C})$ be a CSP. For an assignment σ of each variable to an element in its respective domain, which can be seen as a tuple in $\prod_{j=1}^{n} D_j$, we let $H_\Phi(\sigma)$ be the number of constraints of Φ that are not satisfied by σ. This gives rise to the partition function $Z(\Phi; \theta) = \sum_{\sigma \in \prod_{j=1}^{n} D_j} e^{-\theta H_\Phi(\sigma)}$. When Φ corresponds to the problem of finding a proper q-colouring of a graph G, we note that $Z(\Phi; \theta)$ coincides with the partition function of the Potts model on G with change of variables $y = \exp(-\theta)$. On a different note, the quantity $Z(\Phi; \theta)$ converges to the number of solutions of Φ as θ converges to ∞. As a consequence, certain methods of statistical mechanics that were developed to analyse Gibbs distributions and partition functions can be applied to gain information about the number of solutions of a CSP and, in particular, the number of satisfying assignments of a k-CNF formula. We will see an example of this in Chapter 4, where we use a version of the simulated annealing method originated in statistical mechanics [18, 71] to approximate the number of satisfying

assignments of a k-CNF Φ under certain conditions.

Methods and predictions from statistical mechanics have been particularly successful when analysing the random k-SAT model, which has played a key role as foundational model in the study of randomised algorithms. For integers $k, n, m \geq 2$, the random formula $\Phi = \Phi(k, n, m)$ is a k-CNF formula chosen uniformly at random from the set of formulae with n Boolean variables and m clauses, where each clause has k literals (repetitions allowed). Here, we consider the sparse regime where the density of the formula, $\alpha = m/n$, is bounded by an absolute constant. An important question is determining the probability that the random formula is satisfiable as a function of its density (in the limit $n \to \infty$). Interestingly, for all sufficiently large k, the probability that Φ is satisfiable drops abruptly from 1 to 0 when the density α crosses a certain threshold $\alpha_\star(k)$. Recently there has been tremendous progress in establishing this phase transition (which was originally predicted by the replica symmetric method of statistical mechanics), concluding that $\alpha_\star(k) = 2^k \log 2 - \frac{1}{2}(1 + \log 2) + o_k(1)$ as $k \to \infty$ [38, 34]. Despite the good progress on pinning down this phase transition, finding satisfying assignments for densities up to $\alpha_\star$ poses severe challenges. In fact, the best known algorithm [30] for finding a satisfying assignment of a random formula Φ succeeds up to densities $(1 + o_k(1))\frac{2^k}{k} \log k$, and going beyond such densities is a major open problem with links to phase transitions [1].

From a probabilistic viewpoint, the analysis of the partition function of the random k-SAT model (i.e. the number of satisfying assignments of the formula) depends on subtle properties of the solution set $\Omega = \Omega_\Phi$ consisting of the satisfying assignments of Φ [2, 32, 108, 95]. In this direction, there has been substantial work on finding the so-called free energy of the model, i.e., the asymptotic value of the quantity $\frac{1}{n}\mathbf{E}[\log(1 + |\Omega|)]$. Computing the k-SAT free energy is a difficult problem which is still open (roughly, the difficulty comes from the asymmetry of the model and the unbounded degrees), but there have been results for closely related models including the permissive version of the model [32, 95, 35], the regular k-SAT model [36], and the regular NAE-SAT model [99, 108]. Very recently, a formula for the free energy of the 2-SAT model was given in [2]. In this thesis one of our goals is understanding the geometry of the space of solutions of random k-CNF formulas with the aim of developing fast sampling algorithms that lead to an efficient approximation of the partition function for densities below the threshold $\alpha_*(k)$. Before presenting our results, we introduce the concept of approximation schemes.

1.2 Fully polynomial approximation schemes in spin systems

In this section we briefly overview some of the most successful techniques to come up with approximation algorithms for partition functions, which serves as a motivation for the work presented in this thesis. First, let us define the notion of approximation that we use in approximate counting. Let f be a function from $\{0,1\}^*$ to the positive real numbers. A randomised approximation scheme for f is a probabilistic algorithm that takes as an input a string x and a rational number $\varepsilon \in (0,1)$, and produces as output a random variable Y such that $\mathbb{P}\left[e^{-\varepsilon} \leq Y/f(x) \leq e^{\varepsilon}\right] \geq 3/4$. The choice of $3/4$ in this definition is only due to convenience

– via a standard bootstrapping argument we can swap $3/4$ by any $\delta \in (0,1)$. This definition of approximation scheme can be extended to the case when $f\colon \{0,1\}^* \to \mathbb{C}$ by requiring instead that, with probability at least $3/4$, the output Y of the algorithm is a complex number with $Y = f(x)e^z$ for some $z \in \mathbb{C}$ with $|z| \leq \varepsilon$, see for instance [16]. If this algorithm runs in polynomial time in $|x|$ and $1/\varepsilon$, we say that it is a *fully polynomial randomised approximation scheme* (usually abbreviated as FPRAS). If, moreover, this algorithm is actually deterministic, that is, it always computes $y \in \mathbb{C}$ with $y = f(x)e^z$ for some $z \in \mathbb{C}$ with $|z| \leq \varepsilon$, we call it *fully polynomial-time approximation scheme* (abbreviated as FPTAS). A reader that is not familiar with approximate counting may be surprised that multiplicative approximations are chosen when defining approximation schemes. We remark that this definition arises naturally in the area both from an algorithmic perspective – Monte Carlo methods naturally yield multiplicative approximations – and a hardness perspective – this notion of approximability leads to a rich hierarchy of computational classes of counting problems via approximation preserving reductions, see [39]. Moreover, it is worth noting that we can recover an evaluation of the partition function (even on non-real parameters) from an accurate enough additive approximation, thus making additive approximations not very interesting from a complexity point of view – in fact we will exploit this idea in Chapter 2 of this thesis in some of our reductions.

One of the most relevant FPRAS for partition functions is that of Jerrum and Sinclair on the Ising model when the edge interaction y is real, $y \geq 1$ and there is no external field [76]. This setting is known as the ferromagnetic Ising model. The case when $y \in (0,1)$ is known as the antiferromagnetic Ising model, and it is not difficult to show that approximating the partition function on these edge interactions is NP-hard [61]. Thus, the edge interaction y presents a computational phase transition at $y = 1$. A more difficult problem is that of showing #P-hardness of approximation of the partition function, and we address this problem in Chapter 2.

The algorithm of Jerrum and Sinclair is a Monte Carlo Markov Chain algorithm whose analysis is based on the analysis of the conductance of a certain Markov chain. At the time of writing this thesis we do not know other FPRAS that succeeds at approximating the partition function of the Potts model on any input graph G (other than exact evaluation algorithms at exceptional/easy points). A more promising field of research is that of finding an FPRAS for a partition function when the graphs considered have bounded degree.

1.2.1 Phase transitions on bounded-degree graphs

The study of spin systems on bounded degree graphs has lead to the discovery of several connections between statistical mechanics and approximate counting, and it has motivated the development of several novel techniques to come up with fast approximation schemes. To illustrate this connection, let us focus on the Ising model, keeping in mind that the ideas mentioned here apply to other two-spin systems such as the hard-core model. Let $\Delta \geq 3$ be an integer. We are interested in the problem of approximating $Z_{\text{Ising}}(G;\beta)$ when the input graph G has maximum degree at most Δ. In his seminal work on the independent set polynomial,

Weitz discovered a connection between the physical behaviour of 2-spin systems on the infinite $(\Delta-1)$-regular tree [120], which had already been the focus of previous studies from the point of view of statistical mechanics, and the complexity of approximating the partition function on graphs with maximum degree at most Δ. Before introducing these ideas, let us briefly describe the concept of Gibb measure. Roughly speaking, a Gibbs measure is a probability measure μ over all configurations of an infinite tree such that the marginal of μ on any finite subtree T (possibly with some vertices pinned to spins) agrees with the Gibbs distribution of the Ising model on T. The infinite $(\Delta-1)$-regular tree experiences a physical phase transition at $\beta_c = (\Delta-2)/\Delta$: for edge interactions in $(\beta_c, 1)$ there is a unique Gibbs measure μ, whereas for edges interactions in $(0, \beta_c)$ more than one Gibbs measure arise. This phase transition is related to the number of macroscopic equilibrium of the spin system, see [119] for details. Uniqueness of the Gibbs measure μ occurs when there is *decay of correlations* in $(\Delta-1)$-regular trees, which essentially means that the correlation or influence of a vertex v on the marginal of the root r, defined as $\mu(r \mapsto +|v \mapsto +) - \mu(r \mapsto +|v \mapsto -)$, where $+$ and $-$ are the two possible spins, decays exponentially on the distance from the root to the vertex v (even if some vertices of the tree are pinned to certain spins). Conversely, in the non-uniqueness regime, decay of correlations does not hold and a boundary condition on a set of vertices can continue to have an effect on the marginals of the root, even as the distance tends to infinity.

A key idea in the argument of Weitz is noting that correlations between two vertices r and v on a finite graph G corresponds to correlations in the tree of self-avoiding walk of G starting at r, thus, linking the partition function of the Ising model on graphs with maximum degree Δ to Gibbs measures of the inifite $(\Delta-1)$-regular tree. With this connection in place, one can exploit decay of correlations to approximate marginals of the Gibbs distribution of the Ising model on G, see [120] for details. As a consequence, when $\beta > (\Delta-2)/\Delta$ there is an FPRAS for $Z_{\text{Ising}}(-;\beta)$ on graphs with maximum degree at most Δ [76, 107]. On the other hand, when $0 < \beta < (\Delta-2)/\Delta$, there is no FPRAS for $Z_{\text{Ising}}(-;\beta)$ on graphs with maximum degree at most Δ unless $\mathsf{NP} = \mathsf{RP}$ [52]. Thus, $\beta_c = (\Delta-2)/\Delta$ also behaves as a computational phase transition for the Ising model. One issue with Weitz algorithm is that even if the running time is polynomial in n, the exponent of the polynomial is $O(\log \Delta)$. Very recently there has been a resurgence of the Markov Chain Monte Carlo method based on spectral independence that leads to almost linear sampling algorithms for spin systems in the uniqueness region (on bounded-degree graphs), including the Ising model [7, 28, 19, 17]. Applications of spectral independence require us to show that certain sums of correlations/influences are bounded. These applications usually rely on decay of correlations ideas to prove this bound. The obtained sampling results can be then used to obtain to approximation schemes via self-reducibility arguments [77]. In this thesis, we exploit spectral independence arguments in the context of the random k-SAT model, obtaining the first application of spectral independence that holds even when decay of correlation fails. We will come back to our spectral independence results in Section 1.5, where we give more details about the ideas behind this technique.

1.2.2 Why complex numbers?

Due to the difficulty of determining the complexity of the approximation problem, most approximate counting publications on these partition functions restrict their attention to real parameters. However, given their origin in statistical mechanics, partition functions were studied on non-real parameters since the very beginning. In fact, the framework of viewing partition functions as polynomials in the complex plane of the underlying parameters has been well-explored in statistical physics [121, 69, 85, 122, 12]. Indeed, as pointed out in [109], the possible points of physical phase transitions are precisely the real limit points of complex zeros of the partition function, and, thus, complex zeroes of partition functions have long been studied in the context of statistical mechanics [122]. This problem has recently gained traction in computer science in the context of approximate counting. On the positive side, zero-free regions in the complex plane translate into efficient algorithms for approximating the partition function [8, 101] and this scheme has lead to a broad range of new algorithms even for positive real values of the underlying parameters [88, 103, 87, 104, 10, 62, 63, 65]. On the negative side, the presence of zeros poses a barrier to this approach and, in fact, it has sometimes been demonstrated that zeros mark the onset of computational hardness for the approximability of the partition function [59, 55, 16, 15]. A key approach in all of these applications, and one that we will also develop in this thesis, is the connection to complex dynamics, we give more details in Section 1.4 of this introduction.

Coming back to approximation schemes based on zero-free regions, as noted by Barvinok [8], one can exploit the analytic properties of the partition function to obtain an analytic approximation of $\log Z$ via its Taylor series, which in turns yields a multiplicative approximation of the partition function. This tool turns out to be particularly powerful in the context of bounded-degree graphs, where we can compute the first $O(\log n)$ coefficients of the Taylor series of $\log Z$ in polynomial time for a multitude of partition functions, see [101, 102], including the Ising model among others. We exploit these promising ideas in Chapter 3, where we give a novel zero-free region for the Ising model on bounded degree graphs (see Section 1.4 of this introduction for more details). With this background and motivation in mind, we are ready to describe the main results of this thesis.

1.3 Approximating the partition function of the Potts model

In Chapter 2 we study the complexity of approximating the partition function of the Potts model and the Tutte polynomial on planar graphs as the parameter y ranges in the complex plane. In this section we describe previous work as well as our novel results on this question.

1.3.1 Overview of previous work

Traditionally, this problem has been mainly considered in the case where y is a positive real, however as explained in Section 1.2, recent developments have shown that for various models, including the Ising and Potts models, there is a close interplay between the location of zeros of

the partition function in the complex plane and the approximability of the problem, even for positive real values of y.

The only known hardness of approximation result that applies for general values y in the complex plane is by Goldberg and Guo [55], which addresses the case $q = 2$ (the Ising model) on non-real edge interactions. For general (non-planar) graphs and non-real y, Goldberg and Guo show #P-hardness on the non-real unit circle ($|y| = 1$) with $y \neq \pm i$, and establish NP-hardness elsewhere. The case $q \geq 3$ is largely open apart from the case when y is real which has been studied extensively even for planar graphs [76, 57, 56, 59, 84, 55]. We will review all these results more precisely in the next section, where we also state our main theorems.

1.3.2 Results

In this thesis, we completely classify the complexity of approximating $Z_{\text{Potts}}(G; q, y)$ for $q \geq 2$ and non-real y, even on planar graphs G; in fact, our results also classify the complexity of approximating the Tutte polynomial on planar graphs for reals $q \geq 2$ and non-real γ. Along the way, we also answer a question for the Jones polynomial raised by Bordewich, Freedman, Lovász, and Welsh [20].

To formally state our results, we define the computational problems we consider. Let K and ρ be real algebraic numbers with $K > 1$ and $\rho \in (0, \pi/2)$. We investigate the complexity of the following problems for any integer q with $q \geq 2$ and any algebraic number y.[1]

Name: Factor-K-NormPotts(q, y)
Instance: A (multi)graph G.
Output: If $Z_{\text{Potts}}(G; q, y) = 0$, the algorithm may output any rational number. Otherwise, it must output a rational number $\hat{N}$ such that $\hat{N}/K \leq |Z_{\text{Potts}}(G; q, y)| \leq K\hat{N}$.

Name: Distance-ρ-ArgPotts(q, y)
Instance: A (multi)graph G.
Output: If $Z_{\text{Potts}}(G; q, y) = 0$, the algorithm may output any rational number. Otherwise, it must output a rational $\hat{A}$ such that, for some $a \in \arg(Z_{\text{Potts}}(G; q, \gamma))$, $|\hat{A} - a| \leq \rho$.

A well-known fact is that the difficulty of the problems Factor-K-NormPotts(q, γ) and Distance-ρ-ArgPotts(q, y) does not depend on the constants $K > 1$ and $\rho \in (0, \pi/2)$. This can be proved using standard powering techniques (see [55, Lemma 11] for a proof when $q = 2$). In fact, the complexity of Factor-K-NormPotts(q, y) is the same even for $K = 2^{n^{1-\varepsilon}}$ for any constant $\varepsilon > 0$ where n is the size of the input.

In the special case that q equals 2, we omit the argument q and write Ising instead of Potts in the name of the problem. Similarly, when the input of the problems is restricted to planar graphs, we write PlanarPotts instead of Potts. We also consider the problems

[1] For $z \in \mathbb{C}\backslash\{0\}$, we denote by $|z|$ the norm of z, by $\text{Arg}(z) \in [0, 2\pi)$ the principal argument of z and by $\arg(z)$ the set $\{\text{Arg}(z) + 2\pi j : j \in \mathbb{Z}\}$ of all the arguments of z, so that for any $a \in \arg(z)$ we have $z = |z| \exp(ia)$.

FACTOR-K-NORMTUTTE(q,γ) and DISTANCE-ρ-ARGTUTTE(q,γ) for the Tutte polynomial when q,γ are algebraic numbers. Note also that, when q,γ are real, the latter problem is equivalent to finding the sign of the Tutte polynomial, and we sometimes write SIGNTUTTE(q,γ) (and SIGNPLANARTUTTE(q,γ) for the planar version of the problem).

Our first and main result of Chapter 2 is a full resolution of the complexity of approximating $Z_{\text{Potts}}(G;q,y)$ for $q \geq 3$ and non-real y. More precisely, we show the following.

Theorem 1.1. *Let $q \geq 3$ be an integer, $y \in \mathbb{C}\backslash\mathbb{R}$ be an algebraic number, and $K > 1$. Then, the problems* FACTOR-K-NORMPLANARPOTTS(q,y) *and* DISTANCE-π/3-ARGPLANARPOTTS(q,y) *are* #P*-hard, unless $q = 3$ and $y \in \{e^{2\pi i/3}, e^{4\pi i/3}\}$ when both problems can be solved exactly in polynomial time.*

We remark that, for real $y > 0$, the complexity of approximating $Z_{\text{Potts}}(G;q,y)$ on planar graphs is not fully known, though on general graphs the problem is #BIS-hard [56] and NP-hard for $y \in (0,1)$ [57], for all $q \geq 3$. For real $y < 0$, the problem is NP-hard on general graphs when $y \in (-\infty, 1-q]$ for all $q \geq 3$ ([59])[2] and #P-hard on planar graphs when $y \in (1-q, 0)$ and $q \geq 5$ ([84], see also [58]). Our techniques for proving Theorem 1.1 allow us to resolve the remaining cases $q = 3, 4$ for $y \in (1-q, 0)$ on planar graphs, as a special case of the following theorem that applies for general $q \geq 3$. This is our second main result of Chapter 2.

Theorem 1.2. *Let $q \geq 3$ be an integer, $y \in (-q+1, 0)$ be a real algebraic number, and $K > 1$. Then* FACTOR-K-NORMPLANARPOTTS(q,y) *and* DISTANCE-π/3-ARGPLANARPOTTS(q,y) *are* #P*-hard, unless $(q,y) = (4,-1)$ when both problems can be solved exactly in polynomial time.*

Our third main contribution is a full classification of the range of the parameters where approximating the partition function of the Ising model is #P-hard. On planar graphs G, $Z_{\text{Ising}}(G;y)$ can be computed in polynomial time for all y, see, for instance, [116]. For general (non-planar) graphs and non-real y, our next result shows that the NP-hardness results of [55] can be elevated to #P-hardness.

Theorem 1.3. *Let $y \in \mathbb{C}\backslash\mathbb{R}$ be an algebraic number, and $K > 1$. Then,* FACTOR-K-NORMISING(y) *and* DISTANCE-π/3-ARGISING(y) *are* #P*-hard, unless $y = \pm i$ when both problems can be solved exactly in polynomial time.*

For real y, we remark that the problems of approximating $Z_{\text{Ising}}(G;y)$ and determining its sign (when non-trivial) are well-understood:[3] the problem is FPRASable for $y > 1$, NP-hard for $y \in (0,1)$ ([76]), #P-hard for $y \in (-1,0)$ [55, 59], and equivalent to approximating

[2] Note, for $y \in (-\infty, 1-q) \cup [0,\infty)$, #P-hardness is impossible (assuming NP $\neq$ #P): finding the sign of $Z_{\text{Potts}}(G;q,y)$ is easy, even on non-planar graphs ([59]), and $Z_{\text{Potts}}(G;q,y)$ can be approximated using an NP-oracle. For $y = 1-q$, the same applies when $q \geq 6$; the cases $q \in \{3,4,5\}$ are not fully resolved though [59] shows that $q = 3, 4$ are NP-hard, whereas $q = 5$ should be easy unless Tutte's 5-flow conjecture is false [121, Section 3.5].

[3] Analogously to Footnote 2, for $y \in (-\infty,-1) \cup (0,1)$ #P-hardness is unlikely since the problem can be approximated with an NP-oracle.

#PerfectMatchings for $y < -1$ [57]. For $y = 0, \pm 1$, $Z_{\text{Ising}}(G; y)$ can be computed exactly in polynomial time.

1.3.3 Brief proof outline

In previous #P-hardness results for approximating the Tutte polynomial, the main technique was to reduce the problem of counting the number of (s,t)-cuts with minimum possible cardinality (denoted #MinimumCardinality (s,t)-Cut, see [59, 55] for a definition) to the problem of approximating $Z_{\text{Tutte}}(G; q, \gamma)$ using an elaborate binary search based on suitable oracle calls. Key to these oracle calls are gadget constructions which are mainly based on planar graphs which "implement" points (q', γ'); this means that, by pasting the gadgets appropriately onto the input graph G, the computation of $Z_{\text{Tutte}}(\cdot; q', \gamma')$ reduces to the computation of $Z_{\text{Tutte}}(\cdot; q, \gamma)$. Much of the work in [59, 55], and for us as well, is understanding what values (q', γ') can be implemented starting from (q, γ).

For planar graphs, while the binary-search technique from [55] is still useful, we have to use a different overall reduction scheme since the problem #MinimumCardinality (s,t)-Cut is not #P-hard when the input is restricted to planar graphs [106]. To obtain our #P-hardness results our plan instead is to reduce the problem of exactly evaluating the Tutte polynomial for some appropriately selected parameters q', γ' to the problem of computing its sign and the problem of approximately evaluating it at parameters q, γ; note, this gives us the freedom to use any parameters q', γ' we wish as long as the corresponding exact problem is #P-hard. Then, much of the work consists of understanding what values (q', γ') can be " approximately implemented" starting from (q, γ), with the added difficulty that γ here may be non-real. We conclude that for $q > 2$ and γ non-real we can indeed implement an arbitrarily close approximation to any $(q, \hat{\gamma})$ for any $\hat{\gamma} \in \mathbb{R}$. The gadgets constructed in our results are planar; we give more details about these constructions in an extended proof outline in Section 2.1.

1.3.3.1 Consequences of our techniques for the Tutte/Jones polynomials

While our main results are on the Ising/Potts models, in order to prove them it is convenient to work in the "Tutte world"; this simplifies the proofs and has also the benefit of allowing us to generalise our results to non-integer q. The following result generalises Theorem 1.1 to non-integer $q > 2$.

Theorem 1.4. *Let $q > 2$ be a real, $\gamma \in \mathbb{C} \backslash \mathbb{R}$ be an algebraic number, and $K > 1$. Then,* Factor-K-NormPlanarTutte(q, γ) *and* Distance-$\pi/3$-ArgPlanarTutte(q, γ) *are* #P*-hard, unless $q = 3$ and $\gamma + 1 \in \{e^{2\pi i/3}, e^{4\pi i/3}\}$ when both problems can be solved exactly in polynomial time.*

After the results in this section were made public, our Theorem 1.4 has been reproved in [14], extending the range of q and y where it applies. More precisely, [14, Corollary 13] shows that Factor-K-NormPlanarTutte$(q, y-1)$ and Distance-$\pi/3$-ArgPlanarTutte$(q, y-1)$

are #P-hard for all pairs $(q, y) \in \mathbb{C}^2 \setminus \mathbb{R}^2$ with $q \notin \{0, 1, 2\}$ such that one of the following conditions hold:

1. $|y| > 1$;
2. $|1 - q| > 1$ or $\mathrm{Re}(q) > 3/2$, except when $y = 1$ or $(q, y) \in \{(3, e^{2\pi i/3}), (3, e^{4\pi i/3})\}$.

Note that the exceptions $y = 1$ and $(q, y) \in \{(3, e^{2\pi i/3}), (3, e^{4\pi i/3})\}$ given in Item 2 are included in our Theorem 1.4 as the Tutte polynomial $Z_{\mathrm{Tutte}}(\cdot; q, y - 1)$ can be evaluated in polynomial time at these points. The proof given in [14] uses our reduction from exact evaluation of the Tutte polynomial to approximating the norm or the argument of the partition function. The main difference between both works is the techniques used to develop the implementation results; [14] exploits the connection between complex dynamics and partition functions that we present in Chapter 3 whereas our proof of Theorem 1.4 carefully constructs these implementations exploiting the fact that $q > 2$. In particular, in the proofs of [14] the authors apply the constructions of series-parallel graphs presented in Section 3.6.3 with some modifications.

Our techniques can further be used to elevate previous NP-hardness results of [59, 57] in the Tutte plane to #P-hardness for planar graphs, and answer a question on the Jones polynomial at roots of unity, raised by Bordewich et al. in [20]. Regarding the latter application, in [20] the authors show that polynomial-sized quantum circuits can be simulated by determining the sign of the real part of the Jones polynomial of a link at certain roots of unity and, thus, wondered about the hardness of the latter problem. We show that determining this sign is actually #P-hard. A more detailed discussion can be found in Section 2.6.

1.4 Approximating the partition funtion of the Ising model on bounded-degree graphs

In Chapter 3 we present further work on the complexity of approximating the partition function of the Ising model, this time in terms of the interaction between the edge interaction (denoted β in this section following standard notation for the Ising model) and a parameter $\Delta \geq 3$ which is an upper bound on the maximum degree of the input graph G. Recall that for arbitrary graphs ($\Delta = \infty$), we have shown that the approximation problem is #P-hard except at the easy points $\beta \in \{0, \pm 1, \pm i\}$ (Theorem 1.3). As we have mentioned in Section 1.2, the situation changes for bounded-degree graphs, where approximation schemes based on Barvinok method on zero-free regions can be found for edge interactions close to $\beta = 1$. Motivated by these recent techniques and the powerful connection between complex dynamics and 2-spin systems on complex parameters developed in [16], we explore the complexity picture for graphs with maximum degree Δ.

1.4.1 Overview of previous work

Before describing our results, we briefly describe existing work on the problem of approximating the partition function of the Ising model. Let $\Delta \geq 3$ be an integer. When the input graph G has maximum degree at most Δ, this problem has already been well studied in the case where β is a positive real. As described in Section 1.2, when $\beta > (\Delta-2)/\Delta$ there is an FPRAS for $Z_{\text{Ising}}(-;\beta)$ on graphs with maximum degree at most Δ [76, 107]. When $0 < \beta < (\Delta-2)/\Delta$, there is no FPRAS for $Z_{\text{Ising}}(-;\beta)$ on graphs with maximum degree at most Δ unless $\mathsf{NP} = \mathsf{RP}$ [52].

The complexity of approximation is mostly not understood when β is complex. Prior to this work, there was no inapproximability result for any non-real edge interactions. Indeed, the reductions developed in Chapter 2 do not hold when the set of graphs is restricted to those having maximum degree at most Δ; the main issue with these reductions is that most of the gadgets used blow up the degree of the vertices of the graph. Therefore, a different approach is needed. Regarding tractability results, zero-free regions of the partition function of the Ising model have been the focus of recent publications [9, 86, 89]. Nonetheless, these regions turn out to be far from optimal as we will see in Section 3.3.2 (see Figure 1.1 in this introduction for the case $\Delta = 3$).

1.4.2 Results

In this thesis we shed some light on this approximability problem by significantly extending the known zero-free regions (leading to approximation schemes) and by giving an inapproximability result that covers most of the complex plane. Our zero-free region for the Ising model is stated in Theorem 1.5.

Theorem 1.5. *Let Δ be an integer with $\Delta \geq 3$. Let $G = (V, E)$ be a graph of maximum degree at most Δ. Let $\varepsilon_\Delta = \tan(\pi/(4(\Delta-1))) \in (0,1)$. Then $Z_{\text{Ising}}(G;\beta) \neq 0$ for all $\beta \in \mathbb{C}$ with $|\beta-1|/|\beta+1| \leq \varepsilon_\Delta$.*

Theorem 1.5 can be applied in conjunction with the algorithms of Barvinok, and Patel and Regts [8, 101] to obtain an FPTAS for $Z_{\text{Ising}}(-;\beta)$, giving the following corollary. Note that our approximability results are stated for algebraic edge interactions, as we did in Section 1.3, since they allow for efficient computation, see Section 2.2 and the references therein for more details.

Corollary 1.6. *Let Δ be an integer with $\Delta \geq 3$. Let β be an algebraic number such that $|\beta-1|/|\beta+1| < \varepsilon_\Delta$, where $\varepsilon_\Delta = \tan(\pi/(4(\Delta-1)))$. Then there is an algorithm that, on inputs a graph G with maximum degree at most Δ and a rational $\varepsilon > 0$, runs in time $\text{poly}(\text{size}(G), 1/\varepsilon)$ and outputs $\hat{Z} = Z_{\text{Ising}}(G;\beta)e^z$ for some complex number z with $|z| \leq \varepsilon$.*

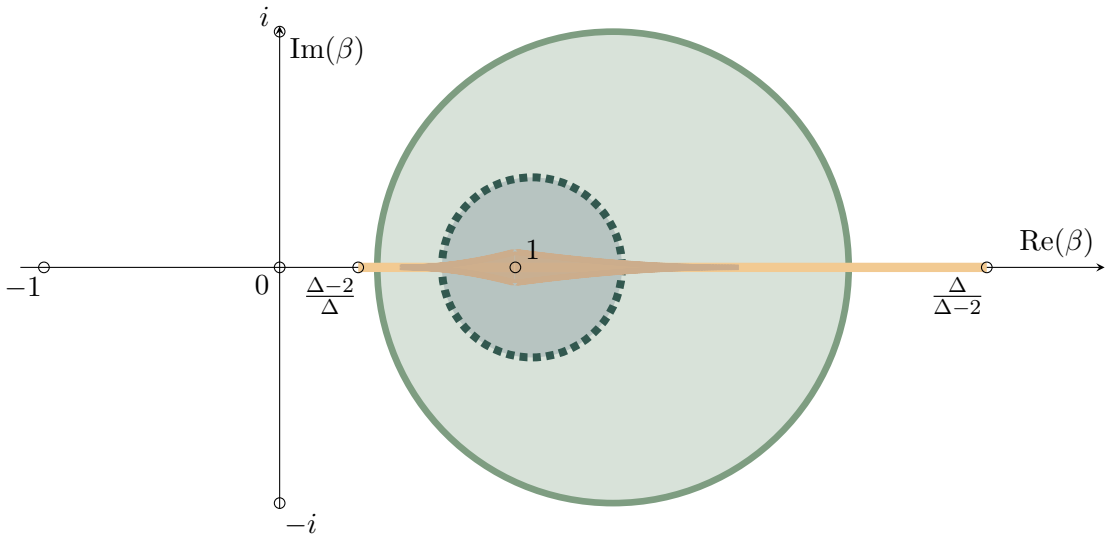


Figure 1.1: Zero-free regions for the partition function of the Ising models on graphs with maximum degree $\Delta = 3$. The following four regions have been plotted:
- The large disk corresponds to the region given in Theorem 1.5.
- The small dotted disk corresponds to $|\beta - 1|/|\beta + 1| \leq \delta_\Delta$, where δ_Δ is as in (3.8), and it contains the regions stated in Corollaries 3.20 and 3.21 due to Barvinok, Mann and Bremner [8, 89].
- The diamond-shaped region corresponds to the zero-free region given in [9] by Barvinok and Barvinok (see Theorem 3.24 for the statement).
- The segment joining $(\Delta - 2)/\Delta$ and $\Delta/(\Delta - 2)$ corresponds to the region given in [86] by Liu, Sinclair and Srivastava (see Theorem 3.23 for the statement).

Theorem 1.5 significantly extends the zero-free regions given in [9, 8, 86, 89]. The case $\Delta = 3$ is depicted in Figure 1.1. In fact, the zero-free regions of Barvinok, and Mann and Bremner [8, 89] are contained in our result for any $\Delta \geq 3$, see Section 3.3.2 for a detailed description of these zero-free regions. In [89] the authors also discuss how an FPRAS for the partition function of the Ising model on bounded-degree graphs can be used to strongly simulate certain classes of IQP circuits. We note that their quantum simulation results are also extended as a consequence of Theorem 1.5.

When it comes to hardness results on complex edge interactions, we are not aware of any hardness result in the literature that covers non-real edge interactions. Our hardness result is given in Theorem 1.7. First, let us introduce some notation. We consider the problem of multiplicatively approximating the norm of $Z_{\text{Ising}}(G;\beta)$ and the problem of additively approximating the principal argument of $Z_{\text{Ising}}(G;\beta)$ for a fixed algebraic number β. These computational problems can be formally stated as follows. Let $K > 1$ and $\rho \in (0, \pi/2)$ be real numbers.

Name: IsingNorm(β, Δ, K)
Instance: A (multi)graph G with maximum degree at most Δ.
Output: If $Z_{\text{Ising}}(G;\beta) = 0$, then the algorithm may output any rational number. Otherwise, it must output a rational number $\hat{N}$ such that $\hat{N}/K \leq |Z_{\text{Ising}}(G;\beta)| \leq K\hat{N}$.

Name: IsingArg(β, Δ, ρ)
Instance: A (multi)graph G with maximum degree Δ.
Output: If $Z_{\text{Ising}}(G;\beta) = 0$, then the algorithm may output any rational number. Otherwise, it must output a rational number $\hat{A}$ such that for some $a \in \arg(Z_{\text{Ising}}(G;\beta))$ we have $|a - \hat{A}| \leq \rho$.

It is important to note that each choice of the parameters β, Δ, K, ρ gives a different computational problem. As noted in Section 1.3, by a standard powering argument of the partition function, the choice of K and ρ does not change the hardness of the problem (as long as $K > 1$ and $\rho \in (0, \pi/2)$), see [55, Lemma 3.2].

Theorem 1.7. *Let Δ be an integer with $\Delta \geq 3$ and let $\beta \in \mathbb{C}$ be an algebraic number such that $\beta \notin \mathbb{R} \cup \{i, -i\}$ and $|\beta - 1|/|\beta + 1| > 1/\sqrt{\Delta - 1}$. Then the problems* IsingNorm$(\beta, \Delta, 1.01)$ *and* IsingArg$(\beta, \Delta, \pi/3)$ *are #P-hard.*

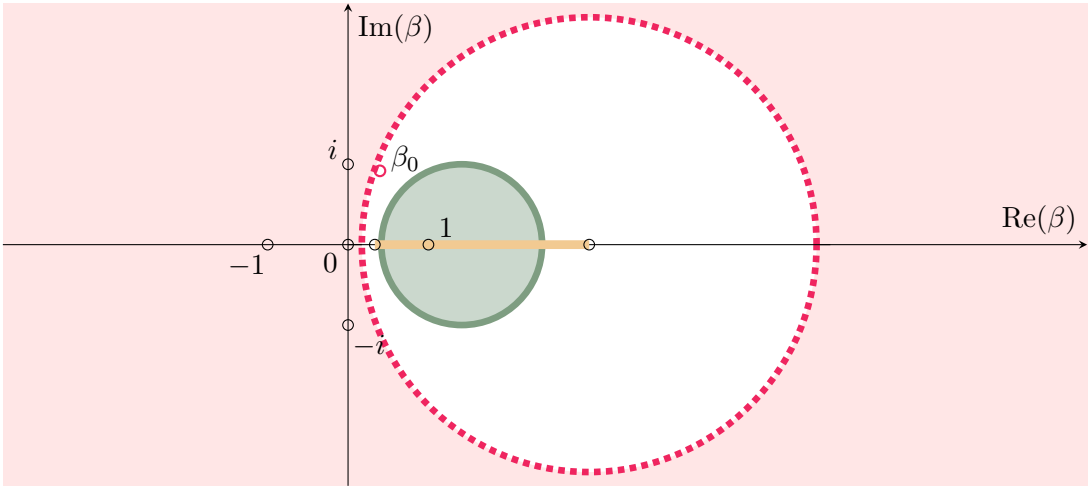


Figure 1.2: The complexity of approximating the partition function of the Ising model on graphs with maximum degree $\Delta = 3$ and $\beta \in \mathbb{C} \setminus \mathbb{R}$.
• Theorem 1.7: when $|\beta - 1|/|\beta + 1| > 1/\sqrt{\Delta - 1}$ and $\beta \notin \{i, -i\}$, IsingNorm$(\beta, \Delta, 1.01)$ and IsingArg$(\beta, \Delta, \pi/3)$ are #P-hard (region outside the large dotted red circle).
• Corollary 3.44: there are points $\beta_0 \in \mathbb{C} \setminus \mathbb{R}$ with $|\beta_0 - 1|/|\beta_0 + 1| < 1/\sqrt{\Delta - 1}$ such that $Z_{\text{Ising}}(G;\beta_0) = 0$ for some graph G with maximum degree Δ. The problems IsingNorm$(\beta_0, \Delta, 1.01)$ and IsingArg$(\beta_0, \Delta, \pi/3)$ are #P-hard.
• Theorem 1.5: there is an FPTAS for $Z_{\text{Ising}}(-;\beta)$ when $|\beta - 1|/|\beta + 1| < \tan(\pi/(4\Delta - 4))$ (region inside the small green circle).
• Theorem 3.23 by Liu, Sinclair and Srivastava [86]: the interval $((\Delta - 2)/\Delta, \Delta/(\Delta - 2))$ is contained in an open zero-free region (thick segment on the real line), so there is an FPTAS for $Z_{\text{Ising}}(-;\beta)$.
∘ The points $0, 1, -1, i$ and $-i$ are easy points of the Ising model: the partition function can be evaluated at these points in polynomial time in the size of the input graph [73].

Corollary 1.6 and Theorem 1.7 leave the complexity of the problems IsingNorm$(\beta, \Delta, 1.01)$

and IsingArg$(\beta, \Delta, \pi/3)$ unaddressed for those edge interactions $\beta \notin \mathbb{R}$ such that

$$\tan\left(\frac{\pi}{4(\Delta-1)}\right) \leq \left|\frac{\beta-1}{\beta+1}\right| \leq \frac{1}{\sqrt{\Delta-1}}. \tag{1.6}$$

It turns out that the partition function has zeros inside the region given by (1.6) (see Corollary 3.44). Moreover, we show that if there is a "nice" graph G such that $Z_{\text{Ising}}(G;\beta) = 0$, then IsingNorm$(\beta, \Delta, 1.01)$ and IsingArg$(\beta, \Delta, \pi/3)$ are #P-hard, see Lemma 3.43 and Corollary 3.45. This allows us to find points β as in (1.6) such that the approximation problems are #P-hard, as depicted in Figure 1.2.

1.4.3 Brief proof outline

In the proof of Theorem 1.5 we use the SAW tree construction of Godsil and Weitz [54, 120] to reduce the study of zero-free regions of partition functions on graphs to the study of zero-free regions of partition functions on trees (see Section 3.2.1 for details). The partition function of a two-spin system on a tree admits a recurrence expression that can be studied to find such zero-free regions. This approach has been successfully applied in the literature for the Ising model and other partition functions [86, 13, 16]. In our work we exploit the properties of the Mobius function $h_\beta(z) = (\beta z + 1)/(\beta + z)$ appearing in this recurrence for the Ising model. This Mobius function satisfies the equality

$$\frac{h_\beta(z)-1}{h_\beta(z)+1} = \frac{(\beta-1)(z-1)}{(\beta+1)(z+1)},$$

which neatly relates properties of $(\beta-1)/(\beta+1)$ to properties of the partition function of the Ising model on trees, and greatly simplifies the derivation of the zero-free region of Theorem 1.5.

In order to obtain our inapproximability results, we construct graphs H with maximum degree at most Δ and two distinguished vertices s, t with degree 1 such that substituting an edge in the host graph with (H, s, t) has the effect of altering the edge interaction β of the original edge to a new edge interaction β'. In this case, we say that H (β, Δ)-implements β', see Section 3.2.3 for a formal definition. As explained in Section 1.3, implementations have played an important role in proofs of hardness of evaluating and approximating partition functions, and they are the main tool to reduce exact computation to approximate computation via a binary search [15, 59]. Here we take advantage of our results developed in Chapter 2 to reduce approximate computation of the partition function to exact computation. Then we exploit arguments from complex dynamics to (β, Δ)-implement approximations of any complex edge interaction. The key idea is coming up with a recurrent construction, so that starting at an edge interaction z, we can implement $g(z^{\Delta-1})$ for some Mobius map g. Then we can analyse which points we can reach by iteratively applying $g(z^{\Delta-1})$. The technical details are quite convoluted as we can not afford the size of our gadgets to blow up, we refer to the full proof outline presented in Section 3.1 for more details. After we made this work public, our results on the connection between complex dynamics and implementations have been applied in the

context of the Tutte and Chromatic polynomials in [14], where the authors analyse the Mobius map $f_q(z) = 1 + q/(z-1)$ and include several improvements to our approach that allow them to conclude hardness for of approximation for planar graphs (maximum degree is not bounded) for a large family of parameters of the Tutte polynomial, improving Theorem 1.4 of this thesis significantly as a consequence (see the paragraph after Theorem 1.4 for a statement of their result).

1.5 Sampling satisfying assignments from the random k-SAT model

In Chapter 4 we study the random k-SAT model, which we have introduced in Section 1.1.3 from the point of view of statistical mechanics. We are motivated by the increasing interest in the computational problem of sampling satisfying assignments of a k-CNF formula Φ uniformly at random and the recent progress in the spectral independence framework to prove fast mixing of certain Markov chains. Sampling is closely connected to the problem of estimating the number of satisfying assignments of Φ, which corresponds to the partition function of the model, see Section 1.1.3, and we will delve into this connection in this section.

1.5.1 Overview of previous work

Regarding the algorithmic problem of sampling satisfying assignments uniformly at random, in the random k-SAT model progress has been slower relative to other well-studied models on random graphs (such as k-colourings or independent sets). One of the main reasons for this is that the usual distribution properties that are typically used to obtain fast algorithms (such as correlation decay and spatial mixing) fail to hold for densities as low as $\alpha = o_k(1)$ [95]. These issues are in fact present already in the bounded-degree k-SAT setting, where the formulae are worst-case but every variable is constrained to have a bounded number of occurrences. For random formulae, these issues are further aggravated by the fact that the degrees of a linear number of variables are unbounded. Very recently, [49] gave an approximate counting algorithm (FPTAS) for the number of satisfying assignments of Φ when k is large enough and $\alpha \lesssim 2^{k/300}$ (where $\lesssim$ hides a polynomial factor in $1/k$). This algorithm elevates Moitra's counting method for bounded-degree k-SAT [93] to the random formula setting, and is the first polynomial-time approximate-counting algorithm to achieve an exponential-in-k bound on α. However, its running time is $n^{\exp(\Theta(k))}$ because the algorithm repeatedly has to enumerate local structures (including solving LPs as a subroutine), which does not scale well with k. Hence, the problem of finding a *fast* algorithm for sampling the satisfying assignments in the random k-SAT model has remained open.

1.5.2 Results

In this work we give a fast algorithm that in time $n^{1+o_k(1)}$ approximately samples satisfying assignments of a random k-SAT formula of density $\alpha \leq 2^{0.039k}$, within arbitrarily small polynomial error. Our work also delves into the connections between the solution space geometry of k-CNF Φ and algorithms for efficiently sampling from the solutions of Φ.

To formally state our main result, we say that an event $\mathcal{E}$ regarding the choice of the random formula Φ holds *with high probability* (abbreviated w.h.p.) if $\Pr(\mathcal{E}) = 1 - o(1)$ as $n \to \infty$, see Section 1.1.3 for the definition of random formula used in this probability distribution. The total variation distance between two probability distributions μ and ν over the same space Ω is given by $\frac{1}{2}\sum_{x\in\Omega}|\mu(x) - \nu(x)|$ and is denoted by $d_{\mathrm{TV}}(\mu, \nu)$. Our main result can now be stated as follows.

Theorem 1.8. *For any real $\theta \in (0, 1)$, there is $k_0 \geq 3$ with $k_0 = O(\log(1/\theta))$ such that, for any integers $k \geq k_0$ and $\xi \geq 1$, and for any positive real $\alpha \leq 2^{0.039k}$, the following holds.*

There is an efficient algorithm to sample from the satisfying assignments of a random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$ within $n^{-\xi}$ total variation distance of the uniform distribution. The algorithm runs in time $O(n^{1+\theta})$, and succeeds w.h.p. over the choice of Φ.

Using standard techniques from the literature, this $O(n^{1+\theta})$ uniform sampling algorithm can be used to obtain a randomised approximation scheme for counting satisfying assignments of Φ in time $O(n^{2+\theta}/\varepsilon^2)$, where ε is the multiplicative error, see [43, Section 7] and Remark 4.51 for details.

1.5.3 Brief proof outline

A unifying theme of previous approaches to counting and sampling CSP solutions is a tool called *marking*, first introduced in [93], which finds a set of "marked" variables such that the set of satisfying assignments projected on these variables is connected. Marking is also an essential step in the developing of our sampling algorithm. Our algorithm first runs a Markov chain to sample assignments of a judiciously-chosen subset of marked variables of Φ (from the relevant marginal distribution), and subsequently extending this random assignment to all the variables. This has the advantage that it avoids the enumeration of local structures, and in fact achieves a nearly-linear running time. We give a high-level overview of the techniques developed in our proofs in Section 4.1. Roughly, our Markov chain is a uniform-block Glauber dynamics which, interestingly, mixes quickly despite the presence of high-degree variables in the random formula. The main point of departure from similar approaches that have been applied to the bounded-degree setting is that we completely circumvent sophisticated coupling arguments that have been used there and which are unfortunately severely constricted by the unbounded degrees in our setting (and made inapplicable). Instead, our main technical contribution is to show that the stationary distribution of our chain is $(c^k \log n)$-spectrally independent for some constant $c \in (0, 1)$, allowing us to apply recently-developed tools in the analysis of Markov chains. Unlike

most applications of spectral independence, our proof does not rely on correlation decay (which, as we mentioned, fails to hold for densities exponential in k). We show our spectral-independence bounds by relating the probabilistic properties of the solution space with the structure of the formula using coupling techniques, so that we can exploit local sparsity properties of random k-SAT. We refer to Section 4.1 for an extended proof outline.

1.5.4 The geometry of the space of satisfying assignments

Our results can be applied to analyse the solution space geometry of random k-CNF formulae for the densities under consideration. Many involved heuristics in statistical physics make predictions about the geometry of the solution space of a random k-CNF instance, often depicted in diagrams like Figure 1.3. Some phases and transitions in this diagram are precisely understood. For example, as mentioned above, the satisfiability threshold (pictured in the transition to the rightmost image in Figure 1.3) was determined by [38]. Another transition of interest is the clustering threshold, above which the solution space of a random k-CNF shatters into exponentially many linearly separated connected components, each of which contains an exponentially small fraction of the satisfying assignments of the formula, as rigorously understood in [33, 3, 90, 96].

Figure 1.3: Heuristic phase diagrams such as above [83] depict the predicted evolution of the structure of the solution space of a random k-CNF as the density α of the formula increases from left to right. We primarily study the leftmost regime.

In the lower-density regime, the solution space geometry of random k-CNFs appears poorly understood. It is widely believed that beneath a critical clause density, the solution space of a random k-CNF is "connected." However, from the literature, it is not even clear what "connected" means. Connectivity is sometimes used in the statistical physics literature as a characterization of the entropy or energy profile of the solution space of a random k-CNF formula as in [124]. In such settings, connectivity is often characterized by an absence of clustering behavior, leaving somewhat of a mystery as to the graphical properties of the solution space of a low density random k-CNF.

Conjectures about connectivity take different forms, and different notions of what connectivity might mean are articulated in [124, 83, 33]. The most common precise notion of connectivity is with respect to Hamming distance, i.e. understanding connectivity properties of the graph of solutions to a random k-CNF, where solutions are $f(n)$-connected if their Hamming distance is at most $f(n)$. At lower densities, random k-CNFs can still have isolated solutions far in

Hamming distance from other satisfying assignments. However, the prevailing belief is that below some threshold, the overwhelming majority of solutions to a random k-CNF lie in a giant component that is $o(n)$-connected.

Much more is known about related notions and local versions of connectivity, like looseness, which characterises how rigid a particular satisfying assignment is. Roughly speaking, a satisfying assignment to a formula is $f(n)$-loose if any variable can be flipped to yield a new satisfying assignment by changing at most $f(n)$ additional variable assignments. In [1], the authors showed $o(n)$-looseness holds in the connectivity regime for related, simpler random models, random q-coloring, and hypergraph 2-coloring, conjecturing that $o(n)$-looseness holds for random k-CNF instances below the clustering threshold. This conjecture was partially resolved in [33], where in an analysis of the decimation process for random k-SAT, the authors observed that with high probability over formulae and satisfying assignments, at least 99% of the variables were $O(\log n)$-loose. Looseness, however, is a local notion, not a global one. The set of elements in $\{0,1\}^n$ that have Hamming weight at least $2n/3$ or at most $n/3$ is 1-loose, but $\Omega(n)$-connected.

We will concern ourselves with the following precise notion of connectivity.

Definition 1.9 (D-Connectivity)**.** *Let $\Phi = (\mathcal{V}, \mathcal{C})$ be a k-CNF formula. For any assignment $\Lambda\colon \mathcal{V} \to \{\mathsf{F}, \mathsf{T}\}$, let $\|\Lambda\|_1$ be the number of variables Λ assigns to be T. Throughout, we implicitly consider variable assignments in $\mathbb{F}_2^n$, so $\|\cdot\|_1$ encodes Hamming weight and $\|\Lambda_1 - \Lambda_2\|_1$ encodes Hamming distance.*

We say a sequence of satisfying assignments $\zeta_0 \leftrightarrow \zeta_1 \leftrightarrow \cdots \leftrightarrow \zeta_\ell$ of Φ is a D-path *if $\|\zeta_i - \zeta_{i-1}\|_1 \leq D$ for each $i \in [t]$. We say two satisfying assignments of Φ, $\Lambda, \Lambda' \in \Omega$, are D-connected if there exists a D-path connecting Λ and Λ' (that is, $\zeta_0 = \Lambda$ and $\zeta_\ell = \Lambda'$).*

Marking-based deterministic and MCMC algorithms are mysterious at first glance, as they enable counting and sampling of k-CNF solutions even in regimes where the solution space is disconnected (i.e. not 1-connected). In this work, we leverage the idea of marking in a novel way to construct paths that certify global connectivity properties of the solution space of k-CNFs at densities close to where counting algorithms are known.

Theorem 1.10. *There is $k_0 \geq 3$ and a polynomial $p(k)$ with non-negative integer coefficients such that, for any integer $k \geq k_0$, and for any positive real $\alpha \leq 2^{0.227k}$, the following claim holds with high probability over the choice of a random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. Two satisfying assignments chosen uniformly at random are $p(k)\log(n)$-connected with probability at least $1 - 1/n$.*

In fact, we show it suffices to take $p(k) = 2k^5$. Our new applications of marking also have implications for other, more local, structural properties of the k-CNF solution space, like looseness.

Definition 1.11. *Given a k-CNF formula $\Phi = (\mathcal{V}, \mathcal{C})$ and a satisfying assignment Λ, a variable $v \in \mathcal{V}$ is* $f(n)$-loose *with respect to Λ if there exists a satisfying assignment to Φ, $\tau \in \Omega$, with $\tau(v) \neq \Lambda(v)$ and $\|\Lambda - \tau\|_1 \leq f(n)$.*

For a random k-CNF formula $\Phi = \Phi(k, n, m)$ and a satisfying assignment Λ chosen uniformly at random, we say that Φ is $f(n)$-loose *if with high probability over (Φ, Λ), all variables $v \in V$ are $f(n)$-loose with respect to Λ.*

We observed earlier that looseness does not imply connectivity; in fact, the other direction of implication is also false as looseness is an incomparable goal to connectivity. Looseness requires that *locally*, we are able to flip *any* variable and get to a nearby solution rather than merely the existence of a path away from a solution. Nonetheless, we are able to deduce some nontrivial results about the looseness of the solution space of random k-CNFs.

Theorem 1.12. *There is $k_0 \geq 3$ such that, for any integer $k \geq k_0$, and for any positive real $\alpha \leq 2^{0.227k}$, the random k-CNF formula $\Phi(k, n, \lfloor \alpha n \rfloor)$ is $\mathrm{poly}(k)\log(n)$-loose.*

We note here that, independently of this work, He, Wu, and Wang [68] also obtained sampling algorithms for random k-CNF formulae. The approach of [68] is based on bounding chains following the recursive sampler method developed in [6, 67, 66]. Their algorithm works up to densities roughly equal to $2^{k/3}$ and samples satisfying assignments within ε total variation distance of the uniform distribution in time $(n/\varepsilon)^{1+O(k^{-5})}$.

1.6 Organisation of this thesis

The rest of this thesis is organised as follows. In Chapter 2 we prove our results on the hardness of approximating the partition function of the Potts model on complex edge interactions, which we have stated in Section 1.3. In Chapter 3 we study the complexity map of approximating the partition function of the Ising model on bounded-degree graphs, proving the results plotted in Figure 1.2, see Section 1.4. In Chapter 4 we describe our almost linear sampler for satisfying assignments of random k-CNF formulas and prove its correctness, yielding an almost quadratic algorithm to approximate the partition function of the random k-SAT model. Finally, in Chapter 5 we present the conclusions and open questions derived from this thesis. Chapters 2, 3 and 4 follow the following organisation. Each chapter starts with a detailed outline of the proof approach, which extends the proof outline given in this introduction and should help the reader to follow the chapter. This proof outline contains informal definitions of the concepts used, that will be formalised later in the chapter, and states the main technical lemmas of each chapter. Then, the second section of each chapter contains the preliminary material needed to follow our proofs. Chapter 4 is an exception to this rule, as the preliminaries and the proof outline are presented at the same time for ease of reading, as our technical lemmas require us to provide a lot of notation in order to state them. After the proof outline and preliminary material, each chapter delves into the technical aspects of our work following the proof outline, the main results of each chapter have been stated in this introduction.

Chapter 2

The complexity of approximating the complex-valued Potts model

◦ This chapter is based on the following publication:
Andreas Galanis, Leslie Ann Goldberg, and Andrés Herrera-Poyatos. The complexity of approximating the complex-valued potts model. *Comput. Complexity*, 31(1):Paper No. 2, 2022. doi:10.1007/s00037-021-00218-x.
◦ A preliminary version of this work appeared in MFCS:
Andreas Galanis, Leslie Ann Goldberg, and Andrés Herrera-Poyatos. The complexity of approximating the complex-valued potts model. In *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 36:1–36:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2020.36.

Organisation of this chapter

This chapter contains the proofs of the results presented in Section 1.3 on the hardness of approximating the partition function of the Potts model on complex edge interactions. The organisation of this chapter is as follows. First, in Section 2.1 we provide a full outline of our proof so as to make it easier to the reader to follow this chapter. Secondly, in Section 2.2 we present some of the preliminary material that we need in our proofs. In Section 2.3 we formalise the concept of approximate shifts, which will play a key role in our reductions. In Section 2.4 we study approximate shifts in the complex plane, proving our main technical results. In Section 2.5 we prove our hardness results via a binary search approach that builds on the approximate shifts developed in the previous sections. Finally, in Section 2.6 we gather some of the relevant consequences of our results, extending hardness results of approximating the Tutte polynomial on planar graphas in the real plane, and answering a question raised by Bordewich, Freedman, Lovász and Welsh in [20] on the connection between quantum computation and approximating the Jones polynomial at roots of unity.

2.1 Proof outline

In this section we provide some insight on the proofs of Theorems 1.1, 1.2, 1.3 and 1.4, introduced in Section 1.3. The proofs presented here are performed in the context of the Tutte polynomial.

In previous #P-hardness results [59, 55] for approximating the Tutte polynomial, the main technique was to reduce the exact counting #MinimumCardinality (s,t)-Cut problem to the problem of approximating $Z_{\text{Tutte}}(G;q,\gamma)$ using an elaborate binary search based on suitable oracle calls. Key to these oracle calls are gadget constructions which are mainly based on planar graphs which "implement" points (q',γ'); this means that, by pasting the gadgets appropriately onto a graph G, the computation of $Z_{\text{Tutte}}(G;q',\gamma')$ reduces to the computation of $Z_{\text{Tutte}}(\cdot;q,\gamma)$. Much of the work in [59, 55], and for us as well, is understanding what values (q',γ') can be implemented starting from (q,γ).

For planar graphs, while the binary-search technique from [55] is still useful, we have to use a different overall reduction scheme since the problem #MinimumCardinality (s,t)-Cut is not #P-hard when the input is restricted to planar graphs [106]. To obtain our #P-hardness results our plan instead is to reduce the problem of exactly evaluating the Tutte polynomial for some appropriately selected parameters q',γ' to the problem of computing its sign and the problem of approximately evaluating it at parameters q,γ; note, this gives us the freedom to use any parameters q',γ' we wish as long as the corresponding exact problem is #P-hard. Then, much of the work consists of understanding what values (q',γ') can be implemented starting from (q,γ), so we focus on that component first.

We first review previous constructions in the literature, known as shifts, and then introduce our refinement of these constructions, which we call polynomial-time approximate shifts, and state our main result about them.

2.1.1 Shifts in the Tutte plane

We say that that there is a *shift* from (q,γ_1) to (q,γ_2) if there is a graph $H=(V,E)$ and vertices s,t such that

$$\gamma_2 = q\frac{Z_{st}(H;q,\gamma_1)}{Z_{s|t}(H;q,\gamma_1)},$$

where $Z_{st}(H;q,\gamma_1)$ is the contribution to $Z_{\text{Tutte}}(H;q,\gamma_1)$ from configurations $A\subseteq E$ in which s,t belong to the same connected component in (V,A), while $Z_{s|t}(H;q,\gamma_1)$ is the contribution from all other configurations A. In the following, we will usually encounter shifts in the (x,y)-parametrisation of the Tutte plane, rather than the (q,γ)-parameterisation which was used for convenience here. To translate between these, set $y=\gamma+1$ and $(x-1)(y-1)=q$, see [121, Chapter 3]. We denote by $\mathcal{H}_q$ the hyperbola $\{(x,y)\in\mathbb{C}^2 : (x-1)(y-1)=q\}$, and we will use both parametrisations as convenient. Section 2.2.2 has a more detailed description of shifts that apply to the multivariate Tutte polynomial.

As described earlier, shifts can be used to "move around" the complex plane. If one knows hardness for some $(x_2,y_2)\in\mathcal{H}_q$, and there is a shift from $(x_1,y_1)\in\mathcal{H}_q$ to (x_2,y_2), then one also obtains hardness for (x_1,y_1). This approach has been very effective when attention is restricted to real parameters [57, 58, 59], however, when it comes to non-real parameters, the success of this approach has been limited. To illustrate this, in [55], the authors established #P-hardness of the Ising model when $y_2\in(-1,0)$, and used this to obtain #P-hardness for y_1 on the unit

circle by constructing appropriate shifts. However, their shift construction does not extend to general complex numbers, and this kind of result seems unreachable with those techniques.

2.1.2 Polynomial-time approximate shifts

To obtain our main theorems, we instead need to consider what we call polynomial-time approximate shifts. First, we need the following definitions.

Definition 2.1 (theta graph, series-parallel graph)**.** *A theta graph consists of two terminals s and t joined by internally disjoint paths [24]. A series-parallel graph with terminals s and t can be obtained from the single-edge graph with edge (s,t) by repeatedly subdividing edges or adding parallel edges [22, Chapter 11].*

An *polynomial-time approximate shift* from $(x_1, y_1) \in \mathcal{H}_q$ to $(x_2, y_2) \in \mathcal{H}_q$ is an algorithm that, for any positive integer n, computes in time polynomial in n a graph G_n that (x_1, y_1)-implements $(\hat{x}_2, \hat{y}_2)$ with $|y_2 - \hat{y}_2| \leq 2^{-n}$. In fact, our constructions need to maintain planarity, and we will typically ensure this by either making every G_n a series-parallel graph, in which case we call the algorithm a *polynomial-time approximate series-parallel shift*, or by making every G_n a theta graph, in which case we call the algorithm a *polynomial-time approximate theta shift*.

These generalised shifts allow us to overcome the challenges mentioned above and are key ingredients in our reduction. Our main technical theorem about them is the following.

Theorem 2.2. *Let $q \geq 2$ be a real algebraic number. Let x and y be algebraic numbers such that $(x,y) \in \mathcal{H}_q$, $y \in (-1,0) \cup (\mathbb{C}\backslash\mathbb{R})$ and $(x,y) \notin \{(i,-i),(-i,i),(\omega_3,\omega_3^2),(\omega_3^2,\omega_3)\}$, where $\omega_3 = \exp(2\pi i/3)$. Then, for any pair of real algebraic numbers $(x',y') \in \mathcal{H}_q$ there is a polynomial-time approximate series-parallel shift from (x,y) to (x',y').*

The exceptions $\{(i,-i),(-i,i),(\omega_3,\omega_3^2),(\omega_3^2,\omega_3)\}$ are precisely the non-real points of the (x,y) plane where the Tutte polynomial of a graph can be evaluated in polynomial time (see Section 2.5.3). As we will see, being able to (x,y)-implement approximations of any number in $(-1,0)$ is essentially the property that makes the approximation problem #P-hard at (x,y).

We remark that the idea of implementing approximations of a given weight or edge interaction has been explored in the literature, though only when all the edge interactions involved are real. We review these results in Section 2.3.

We study the properties of polynomial-time approximate shifts in Section 2.3 and prove Theorem 2.2 in Section 2.4. In the next section, we describe some of the techniques used.

2.1.2.1 Proof Outline of Theorem 2.2

Shifts, as defined in Section 2.1.1, have a transitivity property: if there is a shift from (x_1, y_1) to (x_2, y_2) and from (x_2, y_2) to (x_3, y_3), then there is a shift from (x_1, y_1) to (x_3, y_3), see Section 2.2.2 for more details.

The polynomial-time approximate shift given in Theorem 2.2 is constructed in a similar way. First, we construct a polynomial-time approximate shift from (x, y) to some (x_2, y_2) such that $y_2 \in (-1, 0)$, where x_2 and y_2 depend on x, y. Then, we construct a polynomial-time approximate shift from (x_2, y_2) to (x', y'). Finally, we combine both polynomial-time approximate shifts using an analogue of the transitivity property.

However, when this approach is put into practice, there is a difficulty that causes various technical complications: we only have mild control in our constructions over the intermediate shift (x_2, y_2). In particular, even if the numbers x and y are algebraic, we cannot guarantee that x_2 and y_2 are algebraic, and this causes problems with obtaining the required transitivity property. Instead, we have to work with a wider class of numbers, the set $\mathsf{P}_{\mathbb{C}}$ of *polynomial-time computable numbers*. These are numbers that can be approximated efficiently, i.e., for $y \in \mathsf{P}_{\mathbb{C}}$ there is an algorithm that computes $\hat{y}_n \in \mathbb{Q}[i]$ with $|y - \hat{y}_n| \leq 2^{-n}$ in time polynomial in n [80, Chapter 2]. We denote by $\mathsf{P}_{\mathbb{R}} = \mathbb{R} \cap \mathsf{P}_{\mathbb{C}}$ the set of polynomial-time computable real numbers.

Our polynomial-time approximate shifts are constructed in Section 2.4. The first of these polynomial-time approximate shifts is provided by Lemma 2.3.

Lemma 2.3. *Let q be a real algebraic number with $q \geq 2$. Let x and y be algebraic numbers such that $(x, y) \in \mathcal{H}_q$, $y \in (-1, 0) \cup (\mathbb{C} \backslash \mathbb{R})$ and $(x, y) \notin \{(i, -i), (-i, i), (\omega_3, \omega_3^2), (\omega_3^2, \omega_3)\}$, where $\omega_3 = \exp(2\pi i/3)$. Then there is a polynomial-time approximate series-parallel shift from (x, y) to (x', y') for some $(x', y') \in \mathcal{H}_q$ with $x', y' \in \mathsf{P}_{\mathbb{R}}$ and $y' \in (0, 1)$.*

The construction in Lemma 2.3 is obtained using a theta graph and trying to get a shift that is very close to the real line. However, we cannot control the point (x', y') that we are approximating, and as mentioned, x', y' might not be algebraic. The proof of Lemma 2.3 requires the most technical work in this chapter and is given in Section 2.4.4.

Using Lemma 2.3, we have a series-parallel polynomial-time approximate shift from (x, y) to some $(x', y') \in \mathcal{H}_q$ with $x', y' \in \mathsf{P}_{\mathbb{R}}$ and $y' \in (0, 1)$. Next, we have to construct a polynomial-time approximate shift from (x', y') to $(\hat{x}, \hat{y})$, where $(\hat{x}, \hat{y})$ is the point that we want to shift to in Theorem 2.2. In fact, we actually use a theta shift, which also facilitates establishing the required transitivity property later on. Note that since y' is not necessarily algebraic, we can not directly apply the results that have already appeared in the literature on implementing approximations of edge interactions. In the next lemma, we generalise these results to the setting of polynomial-time computable numbers, where we need to address some further complications that arise from computing with polynomial-time computable numbers instead of algebraic numbers. The proof of the lemma is given in Section 2.4.5.

Lemma 2.4. *Let $q, x, y \in \mathsf{P}_{\mathbb{R}}$ such that $q > 0$, $(x, y) \in \mathcal{H}_q$, y is positive and $1 - q/2 < y < 1$. There is a polynomial-time algorithm that takes as an input:*

- *two positive integers k and n, in unary;*
- *a real algebraic number $w \in [y^k, 1]$.*

The algorithm produces a theta graph J that (x, y)-implements $(\hat{x}, \hat{y})$ such that $\left|\hat{y} - w\right| \leq 2^{-n}$. The size of J is at most a polynomial in k and n, independently of w.

Then, we are able to combine the shifts in Lemmas 2.3 and 2.4 via a transitivity property for polynomial-time approximate shifts (see Lemma 2.16 in Section 2.3), and therefore prove Theorem 2.2, see Section 2.4 for the details.

2.1.3 The reductions

In Section 2.5.6 we show how to use a polynomial-time approximate shift from (x_1, y_1) to (x_2, y_2) to reduce the problem of approximating the Tutte polynomial at (x_2, y_2) to the same problem at (x_1, y_1). The following lemma gives such a reduction for the problem of approximating the norm, we also give an analogous result for approximating the argument.

Lemma 2.5. *Let $q \neq 0$, γ_1 and $\gamma_2 \neq 0$ be algebraic numbers, and $K > 1$. For $j \in \{1, 2\}$, let $y_j = \gamma_j + 1$ and $x_j = 1 + q/\gamma_j$. If there is a polynomial-time series-parallel approximate shift from (x_1, y_1) to (x_2, y_2), then we have a reduction from* Factor-K-NormTutte(q, γ_2) *to* Factor-K-NormTutte(q, γ_1). *This reduction also holds for the planar version of the problem.*

In order to prove Lemma 2.5, we need some lower bounds on the norm of the partition function $Z_{\text{Tutte}}(G; q, \gamma)$. This kind of lower bound plays an important role in several hardness results on the complexity of approximating partition functions [55, 15]. Here, we have to work a bit harder than usual since we have two (algebraic) underlying parameters (in the case of Tutte), and we need to use results in algebraic number theory, see Section 2.5.1 for details.

By combining Theorem 2.2 and Lemma 2.5 with existing hardness results, we obtain our hardness results for non-real edge interactions in Section 2.5.8. On the way, we collect some hardness on real parameters as well that strengthen previous results in the literature, and part of Section 2.5 is devoted to this. The main reason behind these improvements is that previous work on real parameters used reductions from approximately counting minimum cardinality (s, t)-cuts [59, 55], the minimum 3-way cut problem [57], or maximum independent set for planar cubic graphs [58], which are either easy on planar graphs or the parameter regions they cover are considerably smaller or cannot be used to conclude #P-hardness. We instead reduce the exact computation of $Z_{\text{Tutte}}(G; q, \gamma)$ to its approximation, which has the advantage that the problem that we are reducing from is #P-hard for planar graphs [116]. Interestingly, our reduction requires us to apply an algorithm of Kannan, Lenstra and Lovász [78] to reconstruct the minimal polynomial of an algebraic number from an additive approximation of the number. The lower bounds on the partition function $Z_{\text{Tutte}}(G; q, \gamma)$ that are gathered in Section 2.5.1 also play a role in this reduction, the details will be given in Section 2.5.5.

2.2 Preliminaries

2.2.1 The multivariate Tutte polynomial

The random cluster formulation of the multivariate Tutte polynomial is particularly convenient when working with implementations (as we will see in Section 2.2.2), and is defined as follows. Let $G = (V, E)$ be a graph. For any weight function $\gamma\colon E \to \mathbb{C}$ and $q \in \mathbb{C}$, the *multivariate Tutte polynomial of* G is

$$Z_{\text{Tutte}}(G; q, \gamma) = \sum_{A \subseteq E} q^{k(A)} \prod_{e \in A} \gamma_e. \tag{2.1}$$

We will make use of the following notation. Let s and t be two distinct vertices of G. We define

$$Z_{st}(G; q, \gamma) = \sum_{\substack{A \subseteq E: \\ s \text{ and } t \text{ in the same component}}} q^{k(A)} \prod_{e \in A} \gamma_e.$$

Analogously, let $Z_{s|t}$ be the contribution to $Z_{\text{Tutte}}(G; q, \gamma)$ from the configurations $A \subseteq E$ such that s and t are in different connected components in (V, A). That is, $Z_{s|t}(G; q, \gamma) = Z_{\text{Tutte}}(G; q, \gamma) - Z_{st}(G; q, \gamma)$.

2.2.2 Implementing weights, series compositions and parallel compositions

In this section, we define implementations, shifts, series compositions and parallel compositions. The definitions and results that we give are standard and can also be found, for instance, in [73, Section 4], [58, Section 2.1] or [109, Section 4].

Let $q \in \mathbb{C}$ with $q \neq 0$. The value of q is fixed across all this section. Let H be a weighted graph with weight function $\hat{\gamma}$. Let s and t be two distinct vertices of H, which are usually referred to as terminals. We say that the graph H *$\hat{\gamma}$-implements* the weight w with respect to the terminals s and t if

$$w = q \frac{Z_{st}(H; q, \hat{\gamma})}{Z_{s|t}(H; q, \hat{\gamma})}.$$

We say that H $\hat{\gamma}$-implements the weight w if there are terminals s and t such that H $\hat{\gamma}$-implements the weight w with respect to s and t. These definitions are motivated by Lemma 2.6, whose proof is a straightforward computation involving the definitions of implementations and the multivariate Tutte polynomial.

Lemma 2.6 ([58, Equation 2.2])**.** *Let G and H be two graphs with weight functions γ and $\hat{\gamma}$ respectively. Let f be an edge of G with weight γ_f such that H $\hat{\gamma}$-implements γ_f with respect to terminals s and t. Let G_f be the graph constructed by considering the union of G and H, identifying the terminals s and t with the endpoints of f in G and removing f. Let γ' be the weight function on G_f that inherits the weights from γ and $\hat{\gamma}$. Then*

$$Z_{st}\left(G_f; q, \gamma'\right) = \frac{Z_{s|t}\left(H; q, \hat{\gamma}\right)}{q^2} Z_{st}\left(G; q, \gamma\right), \quad Z_{s|t}\left(G_f; q, \gamma'\right) = \frac{Z_{s|t}\left(H; q, \hat{\gamma}\right)}{q^2} Z_{s|t}\left(G; q, \gamma\right).$$

In particular, we have $Z_{Tutte}(G_f; q, \gamma') = \frac{Z_{s|t}(H;q,\hat{\gamma})}{q^2} Z_{Tutte}(G; q, \gamma)$. *Moreover, if* G γ*-implements a weight* w*, then* G_f *also* γ'*-implements* w.

Therefore, if we can compute $Z_{s|t}(H; q, \hat{\gamma})$ efficiently and $Z_{s|t}(H; q, \hat{\gamma}) \neq 0$, then computing $Z_{\text{Tutte}}(G; q, \gamma)$ is as hard as computing $Z_{\text{Tutte}}(G_f; q, \gamma')$. This observation leads to some of the reductions that appear in this chapter.

In the remaining sections we usually assume that the weights are constant, that is, each edge of the graph has the same weight, and we will make it clear when this is not the case. In the constant weight function case Lemma 2.6 can be applied to each edge of the graph constructed by copying G and substituting each edge f in G by a copy of H (identifying the endpoints of f with s and t). Let $\alpha_1, \alpha_2 \in \mathbb{C}$. We say that there is a *shift* from (q, α_1) to (q, α_2) if there is a graph H that α_1-implements α_2. An important property of shifts is transitivity; if there are shifts from (q, α_1) to (q, α_2) and from (q, α_2) to (q, α_3), then there is a shift from (q, α_1) to (q, α_3). This is a consequence of Lemma 2.6. Let $y_1 = \alpha_1 + 1$ and $y_2 = \alpha_2 + 1$. We define x_1 and x_2 by $q = (x_1 - 1)(y_1 - 1) = (x_2 - 1)(y_2 - 1)$, which is the change of variables that relates the Tutte polynomial and Z_{Tutte}. We equivalently refer to the shift from (q, α_1) to (q, α_2) as a shift from (x_1, y_1) to (x_2, y_2), and we also say that H (x_1, y_1)-implements (x_2, y_2). This notation is convenient to express many of the shifts considered in this chapter.

To conclude this section we introduce two tools that will provide us with many examples of implementations and shifts: parallel compositions and series compositions. For each $j \in \{1, 2\}$, let G_j be a graph, let s_j and t_j be two terminals of G_j, and let γ_j be a weight function such that G_j γ_j-implements a weight w_j with respect to s_j and t_j.

Parallel compositions. The parallel composition of (G_1, s_1, t_1) and (G_2, s_2, t_2) is the graph G constructed by considering the union of G_1 and G_2 and identifying s_1 with s_2 and t_1 with t_2. Let $\hat{\gamma}$ be the weight function on G inherited from γ_1 and γ_2. It is well-known and easy to check that G $\hat{\gamma}$-implements the weight

$$w = (1 + w_1)(1 + w_2) - 1 \tag{2.2}$$

with respect to the terminals s_1 and t_1. Let (x_1, y_1) and (x_2, y_2) be the Tutte coordinates of (q, w_1) and (q, w_2) respectively (so $y_j = w_j + 1$ and $(x_j - 1)(y_j - 1) = q$). Then the Tutte coordinates of (q, w) are (x, y) with $y = y_1 y_2$ and $(x - 1)(y - 1) = q$. Let Υ be a graph with two vertices s, t and one edge joining them, and let Υ^n be the parallel composition of n copies of (Υ, s, t) (so Υ^n has two vertices and n edges joining them). Then Υ^n (x, y)-implements (x', y') with $y' = y^n$ and $(x' - 1)(y' - 1) = q$. This is known as an *n-thickening* of (x, y) and it yields a shift from (x, y) to (x', y^n).

Series compositions. The series composition of (G_1, s_1, t_1) and (G_2, s_2, t_2) is the graph G constructed by considering the union of G_1 and G_2 and identifying t_1 with s_2. Let $\hat{\gamma}$ be the weight function on G inherited from γ_1 and γ_2. It is well-known and easy to check that G

$\hat{\gamma}$-implements the weight

$$w = \frac{w_1 w_2}{w_1 + w_2 + q}$$

with respect to the terminals s_1 and t_2. Note that w satisfies

$$\left(1 + \frac{q}{w}\right) = \left(1 + \frac{q}{w_1}\right)\left(1 + \frac{q}{w_2}\right). \tag{2.3}$$

Let (x_1, y_1) and (x_2, y_2) be the Tutte coordinates of (q, w_1) and (q, w_2) respectively (so $y_j = w_j + 1$ and $(x_j - 1)(y_j - 1) = q$). Then, in view of (2.3), the Tutte coordinates of (q, w) are (x, y) with $x = x_1 x_2$ and $(x - 1)(y - 1) = q$. Let Υ be a graph with two vertices s, t and one edge joining them, and let Υ_n be the series composition of n copies of (Υ, s, t) (so Υ_n is a path graph with n edges). Then Υ_n (x, y)-implements (x', y') with $x' = x^n$ and $(x' - 1)(y' - 1) = q$. This is known as an *n-stretching* of (x, y) and it yields a shift from (x, y) to (x^n, y').

For series-parallel and theta graphs (see Definition 2.1), these constructions give that either $Z_{s|t}(G; q, \gamma) = 0$, or the series-parallel graph G (with terminals s and t) γ-implements a weight $w(G, s, t; q, \gamma)$ that can be computed from the recursive definition of series-parallel graphs in polynomial time. In particular, let $\Theta_{(l_1,\ldots,l_m)}$ be the theta graph with m internal paths of lengths $l_1, \ldots, l_m$. In this case,[1] we have that

$$w\left(\Theta_{(l_1,\ldots,l_m)}, s, t; q, \gamma\right) = \prod_{j=1}^{m}\left(1 + \frac{q}{x^{l_j} - 1}\right) - 1, \tag{2.4}$$

where $x = 1 + q/\gamma$. Series-parallel graphs can be built using series and parallel compositions. The following definition is equivalent to the one in Definition 2.1. A graph G is *series-parallel* (with terminals s and t) if either G is the graph with two vertices s and t and one edge joining them, or G is the parallel or series composition of (G_1, s_1, t_1) and (G_2, s_2, t_2), where $s = s_1$, $t = t_2$ and G_j is a series-parallel graph with terminals s_j and t_j [22, Chapter 11].

Finally, accross all this thesis the *size of a graph* $G = (V, E)$ is the integer $\text{size}(G) = |V| + |E|$. Note that the size of $\Theta_{(l_1,\ldots,l_m)}$ is $2\sum_{j=1}^{m} l_j - m + 2$.

2.2.3 Computing with algebraic numbers

Our reductions will work when the partition functions under consideration are evaluated on algebraic numbers. Here we overview how we algorithmically perform computations in the field of algebraic numbers. We represent an algebraic number z as its minimal polynomial p and a rectangle R of the complex plane such that z is the only root of p in R. We can compute the addition, subtraction, multiplication, division and conjugation of algebraic numbers in polynomial time in the length of their representations, see [113] for details. As a consequence, we can also compute the real and imaginary parts of z and the norm of z, which are algebraic

[1] We should mention that we will make use of the Θ asymptotic notation in this chapter and this notation should not be confused with that of theta graphs.

numbers themselves, in polynomial time. Note that an algebraic number is 0 if and only if its minimal polynomial is x, which can be easily checked in this representation. Hence, we can also determine in polynomial time whether two algebraic numbers z_1 and z_2 are equal by checking if $z_1 - z_2$ is 0.

When z is a real algebraic number, we can simply represent it as its minimal polynomial p and an interval I with rational endpoints such that z is the only root of p in I. If we are given a real algebraic number z with this representation, then we can approximate it as closely as we want by applying Sturm sequences and binary search [41]. In fact, for z_1 and z_2 real algebraic numbers, Sturm sequences also allow us to check whether $z_1 \geq z_2$ in time polynomial in the length of the representations of z_1 and z_2. See [41] for more details and complexity analysis.

A *root of unity* is a complex number z such that $z^k = 1$ for some positive integer k. The smallest positive integer n such that $z^n = 1$ is the *order of* z. Note that roots of unity are algebraic numbers. The roots of unity of order n share the same minimal polynomial, known as the n-th cyclotomic polynomial, whose degree is $\varphi(n)$, the Euler phi function. We can determine whether an algebraic number z is a root of unity by checking whether its minimal polynomial is cyclotomic, see [21] for a polynomial-time algorithm. If z is a root of unity, then we can easily compute its order from its representation; we compute the smallest n such that the minimal polynomial of z divides $z^n - 1$. This computation runs in polynomial time in the length of the representation of z as a consequence of the elementary bound $\varphi(n) \geq \sqrt{n/2}$.

2.3 Polynomial-time approximate shifts

Implementing a specific weight cannot always be achieved. Nonetheless, sometimes we can implement an approximation of the desired weight with as much precision as we need. These implementations have been exploited several times in the literature on Tutte polynomials and the Ising model; see [57, 58, 59, 60]. Here we collect some of these results appearing in [59], which in turn are based on arguments in [58]; here, we follow the presentation in [60] (that was stated for $q = 2$).

Lemma 2.7 ([60, Lemma 22], [59, Lemma 5])**.** *Let x and y be real algebraic numbers such that $y \notin [-1, 1]$ and $(x-1)(y-1) = q > 0$. There is a polynomial-time algorithm that takes as an input:*

- *two positive integers n and k, in unary;*
- *a real algebraic number $y' \in [1, |y|^k]$.*

This algorithm produces a theta graph G that (x, y)-implements $(\hat{x}, \hat{y})$ such that $\left|y' - \hat{y}\right| \leq 2^{-n}$. The size of G is at most a polynomial in n and k, independently of y'.

In Lemma 2.4 (Section 2.4), we give a similar result to Lemma 2.7 where the numbers x and y may not be algebraic. The fact that the graph G computed in Lemma 2.7 is a theta graph

is not directly stated in the statement of [59, Lemma 5] but it can easily be inferred from the proof. This also applies to Lemma 2.8.

Lemma 2.8 ([60, Lemma 22], [59, Lemma 7])**.** *Let x_1, y_1, x_2, y_2 be real algebraic numbers such that $y_1 \in (-1, 1)$, $y_2 \notin [-1, 1]$, and $(x_1 - 1)(y_1 - 1) = (x_2 - 1)(y_2 - 1) = q < 0$. There is a polynomial-time algorithm that takes as an input:*

- *two positive integers n and k, in unary;*
- *a real algebraic number $y' \in [1, |y_1|^{-k}]$.*

This algorithm produces a theta graph $G = (V, E)$ and a weight function $\hat{\gamma} \colon E \to \{y_1 - 1, y_2 - 1\}$ such that G $\hat{\gamma}$-implements $(\hat{x}, \hat{y})$ with $\left|y' - \hat{y}\right| \leq 2^{-n}$. The size of G is at most a polynomial in n and k, independently of y'.

Corollary 2.9. *Let x_1, y_1, x_2, y_2 be real algebraic numbers such that $y_1 \in (-1, 0) \cup (0, 1)$, $y_2 \notin [-1, 1]$, $(x_1 - 1)(y_1 - 1) = (x_2 - 1)(y_2 - 1) = q$, $q \neq 0$. There is a polynomial-time algorithm that takes as an input:*

- *two positive integers n and k, in unary;*
- *a positive real algebraic number y' such that $|y'| \in [|y_1|^k, |y_1|^{-k}]$.*

This algorithm produces a theta graph $G = (V, E)$ and a weight function $\hat{\gamma} \colon E \to \{y_1 - 1, y_2 - 1\}$ such that G $\hat{\gamma}$-implements $(\hat{x}, \hat{y})$ with $\left|y' - \hat{y}\right| \leq 2^{-n}$. The size of G is at most a polynomial in n and k, independently of y'. Moreover, if either $y_1 < 0$ or $y_2 < 0$, then the restriction that y' is positive can be replaced with a restriction that y' is non-zero.

Proof. This result easily follows from Lemmas 2.7 and 2.8 by an argument of Goldberg and Jerrum (see the proof of [59, Lemma 2]). We include here their argument for completeness. The case when $y' \geq 1$ has been covered in Lemmas 2.7 and 2.8. First, let us assume that $y' \in (0, 1)$. We have $1 \leq y' \cdot y_1^{-2k} \leq |y_1|^{-2k}$ and using Lemmas 2.7 and 2.8 we can implement $\tilde{y}$ with $|\tilde{y} - y' \cdot y_1^{-2k}| \leq 2^{-n}$. We have $|y_1^{2k}\tilde{y} - y'| \leq 2^{-n}$, so we set $\hat{y} = y_1^{2k}\tilde{y}$. The graph G is the parallel composition of the graph used to implement $\tilde{y}$ and $2k$ edges with weight y_1. Finally, let us assume that there is $i \in \{1, 2\}$ such that $y_i < 0$, and let us consider the case where y' is negative. We implement an approximation $\hat{y}'$ of $y'/y_i > 0$, and return $\hat{y} = \hat{y}' y_i$. □

The graphs G produced by the algorithms given in Lemma 2.7, Lemma 2.8 and Corollary 2.9 are theta graphs. One may wonder which weights can be approximated as in these results. This leads to the following definition, which was informally introduced in the proof outline (Section 2.1); we state it formally here for ease of reading.

Definition 2.10 (polynomial-time approximate shift)**.** *Let $(x_1, y_1), (x_2, y_2) \in \mathcal{H}_q$. Let $\gamma_1 = y_1 - 1$ and $\gamma_2 = y_2 - 1$. We say that there is a* polynomial-time approximate shift *from (q, γ_1) to (q, γ_2) or, equivalently, from (x_1, y_1) to (x_2, y_2), if there is an algorithm that, for any positive*

integer n, computes in polynomial time in n a graph G_n that (x_1, y_1)-implements $(\hat{x}_2, \hat{y}_2)$ with $|y_2 - \hat{y}_2| \leq 2^{-n}$. If the graph G_n computed by this algorithm is always a theta graph (resp. a series-parallel graph), then we say that this is a polynomial-time approximate theta shift *(resp. polynomial-time approximate series-parallel shift).*

Lemma 2.7 gives polynomial-time approximate theta shifts from (x_1, y_1) to (x_2, y_2) when the considered numbers are real algebraic, $y_1 \notin [-1, 1]$, $y_2 \in [1, \infty)$ and $q > 0$. Note that shifts are a particular case of polynomial-time approximate shifts. Moreover, due to the transitivity property of shifts, if there is a shift from (x_1, y_1) to (x_2, y_2) and there is a polynomial-time approximate shift from (x_2, y_2) to (x_3, y_3), then there is a polynomial-time approximate shift from (x_1, y_1) to (x_3, y_3). In fact, polynomial-time approximate shifts exhibit some of the properties of shifts; in Lemma 2.11 we show that they behave well with respect to parallel and series compositions and in Lemma 2.16 we show that they are transitive under certain conditions. In Section 2.4 we give more examples of polynomial-time approximate shifts, some of which will be constructed by transitivity. These approximate shifts play an important role in our hardness proofs.

Lemma 2.11. *Let $q \in \mathbb{C} \setminus \{0\}$ and let $(x_j, y_j) \in \mathcal{H}_q$ for each $j \in \{1, 2, 3\}$. Let us assume that there are polynomial-time approximate shifts from (x_1, y_1) to (x_2, y_2), and from (x_1, y_1) to (x_3, y_3). Let $(x_4, y_4), (x_5, y_5) \in \mathcal{H}_q$ with $y_4 = y_2 y_3$ and $x_5 = x_2 x_3$. Then:*

1. *there is a polynomial-time approximate shift from (x_1, y_1) to (x_4, y_4);*
2. *there is a polynomial-time approximate shift from (x_1, y_1) to (x_5, y_5).*

Moreover, if the polynomial-time approximate shifts from (x_1, y_1) to (x_2, y_2) and (x_3, y_3) are series-parallel, then the obtained polynomial-time approximate shifts are also series-parallel.

Proof. For $j \in \{2, 3\}$, let $G_{n,j}$ be the graph computed by the polynomial-time approximate shift from (x_1, y_1) to (x_j, y_j), so $G_{n,j}$ (x_1, y_1)-implements $(\hat{x}_j, \hat{y}_j)$ with $|y_j - \hat{y}_j| \leq 2^{-n}$, for certain terminals t_j and s_j.

For Item 1, let P_n be the parallel composition of $(G_{n,2}, s_2, t_2)$ and $(G_{n,3}, s_3, t_3)$. The graph P_n gives a shift from (x_1, y_1) to $(\hat{x}_4, \hat{y}_2\hat{y}_3) \in \mathcal{H}_q$. Since $|y_3 - \hat{y}_3| \leq 2^{-n}$, we have $|\hat{y}_3| \leq |y_3| + 1$ and

$$|y_2 y_3 - \hat{y}_2\hat{y}_3| \leq |y_2 - \hat{y}_2|\,|\hat{y}_3| + |y_3 - \hat{y}_3|\,|y_2| \leq 2^{-n}\left(|y_3| + 1 + |y_2|\right).$$

Therefore, for k large enough, the graphs P_{n+k} give a polynomial-time approximate shift from (x_1, y_1) to (x_4, y_4) with $y_4 = y_2 y_3$.

For Item 2, the proof is analogous but now we define the graph S_n as the series composition of $(G_{n,2}, s_2, t_2)$ and $(G_{n,3}, s_3, t_3)$, which gives a shift from (x_1, y_1) to $(\hat{x}_2\hat{x}_3, \hat{y}_4) \in \mathcal{H}_q$.

Note that if the original polynomial-time approximate shifts are series-parallel, then the obtained ones are also series-parallel by the definition of series-parallel graphs. □

When it comes to hardness results, we are only interested in algebraic numbers. However, we will have to consider polynomial-time approximate shifts from (x_1, y_1) to (x_2, y_2) such that the numbers involved are not algebraic. This is due to the fact that, even if x_1 and y_1 are algebraic, x_2 and y_2 might not be. Nonetheless, in that case we can ensure that x_2 and y_2 are polynomial-time computable.

Definition 2.12 (polynomial-time computable number)**.** *A real number x is* polynomial-time computable *if there is a function $\phi\colon \mathbb{N} \to \mathbb{Q}$ that is computable in polynomial time (with the input written in unary notation, i.e., 0^n) such that $|x - \phi(n)| \leq 2^{-n}$ for all $n \in \mathbb{N}$, see [80, Chapter 2] for a treatment on these numbers.*

The definition of polynomial-time computable number given in [80, Chapter 2] uses dyadic rational numbers instead of rational numbers, but these two definitions are easily seen to be equivalent. We denote the set of polynomial-time computable real numbers by $\mathsf{P}_{\mathbb{R}}$. One can easily show that the set $\mathsf{P}_{\mathbb{R}}$ is a field. Real algebraic numbers are in $\mathsf{P}_{\mathbb{R}}$ because we can approximate them as closely as we want by applying Sturm sequences and binary search [41]. We say that a complex number z is *polynomial-time computable* if $z = x + iy$ for some $x, y \in \mathsf{P}_{\mathbb{R}}$. We denote the set of polynomial-time computable complex numbers by $\mathsf{P}_{\mathbb{C}}$. Algebraic numbers are in $\mathsf{P}_{\mathbb{C}}$ (their real and imaginary parts are real algebraic numbers). It turns out that $\mathsf{P}_{\mathbb{C}}$ is an algebraically-closed field [80, Chapter 2]. In particular, for $z \in \mathsf{P}_{\mathbb{C}}$, we have $|z| \in \mathsf{P}_{\mathbb{R}}$.

If there is a polynomial-time approximate theta shift from (x_1, y_1) to (x_2, y_2) and x_1 and y_1 are algebraic, then we can compute in polynomial time in n an algebraic number that additively approximates y_2 up to an additive error 2^{-n}. Since we can approximate algebraic numbers by rational numbers efficiently, it follows that x_2 and y_2 are polynomial-time computable. However, if we only know that x_1 and y_1 are polynomial-time computable, then it is not clear if x_2 and y_2 are polynomial-time computable or not. Lemma 2.15 gives a partial answer to this question and plays a key role in our transitivity result for polynomial-time approximate shifts (Lemma 2.16). First, we need to prove some lemmas on polynomial-time computable numbers.

Lemma 2.13. *Let $z \in \mathsf{P}_{\mathbb{C}}$. There is an algorithm that computes $b_1 \in \mathbb{Q}$ with $|z| \leq b_1$. Moreover, if $z \neq 0$, then there is an algorithm that computes $b_2 \in \mathbb{Q}$ with $0 < b_2 \leq |z|$.*

Proof. Let $x = |z|$. From $x \in \mathsf{P}_{\mathbb{R}}$, it follows that we can compute a sequence $\hat{x}_n \in \mathbb{Q}$ such that $|x - \hat{x}_n| \leq 2^{-n}$, that is, we have $x \in [\hat{x}_n - 2^{-n}, \hat{x}_n + 2^{-n}]$. This computation for $n = 1$ gives the upper bound $\hat{x}_1 + 1/2$. Note that the sequences $\hat{x}_n - 2^{-n}$ and $\hat{x}_n + 2^{-n}$ converge to x. Hence, if $x \neq 0$, then there must be n such that $0 < \hat{x}_n - 2^{-n} \leq x$. We compute $\hat{x}_n$ until this inequality happens, obtaining the desired lower bound. □

Lemma 2.14. *Let $z \in \mathsf{P}_{\mathbb{C}}$ with $|z| \neq 1$. There is a polynomial-time algorithm that takes as inputs two positive integers n and k and computes a positive integer $r(n, k)$ such that*

1. *$r(n, k)$ is increasing in k;*
2. *$r(n, k) = n + \Theta(k)$;*

3. *if* $|z-\hat{z}| \leq 2^{-r(n,k)}$, *then* $\left|\frac{1}{z^k-1} - \frac{1}{\hat{z}^k-1}\right| \leq 2^{-n}$.

Proof. By Lemma 2.13, we can compute an integer $t \geq 0$ such that $2^{-t} \leq ||z|-1|$ and $|z| \leq 2^t$. Note that for every integer $k \geq 1$ we have the bound $\left|z^k-1\right| \geq 2^{-t}$. Indeed, if $|z|<1$, then

$$2^{-t} \leq 1-|z| \leq 1-|z|^k \leq \left|z^k-1\right|$$

and when $|z|>1$, we analogously find that $2^{-t} \leq |z|-1 \leq |z|^k-1 \leq \left|z^k-1\right|$.

Let n and k be the inputs of our algorithm. Let $r(n,k) = n+(t+1)(k+1)$, and note that r is increasing in k and $r(n,k) = n+\Theta(k)$, establishing Items 1 and 2.

For Item 3, consider $\hat{z}$ such that $|z-\hat{z}| \leq 2^{-r(n,k)}$. Since $|\hat{z}| \leq |z|+2^{-r(n,k)} \leq 2^{t+1}$, for every $j \in \{0,\dots,k-1\}$ we have $|\hat{z}|^j |z|^{k-1-j} \leq 2^{t(k-1)+j}$ and hence

$$\begin{aligned}\left|z^k-\hat{z}^k\right| &= \left|(z-\hat{z})\sum_{j=0}^{k-1}\hat{z}^j z^{k-1-j}\right| \leq |z-\hat{z}|\sum_{j=0}^{k-1}|\hat{z}|^j|z|^{k-1-j} \\ &\leq |z-\hat{z}|\sum_{j=0}^{k-1}2^{t(k-1)+j} < |z-\hat{z}|\,2^{t(k-1)+k} \leq 2^{-(n+2t+1)}.\end{aligned}$$

Moreover, we have that $\left||z^k-1|-\left|\hat{z}^k-1\right|\right| \leq \left|z^k-\hat{z}^k\right| < 2^{-(t+1)}$ and, thus,

$$\left|\hat{z}^k-1\right| \geq \left|z^k-1\right| - 2^{-(t+1)} \geq 2^{-(t+1)},$$

where we used that $\left|z^k-1\right| \geq 2^{-t}$. Therefore, we find that

$$\left|\frac{1}{z^k-1}-\frac{1}{\hat{z}^k-1}\right| = \left|\frac{z^k-\hat{z}^k}{(z^k-1)(\hat{z}^k-1)}\right| \leq 2^{2t+1}\left|z^k-\hat{z}^k\right| \leq 2^{-n}. \qquad \square$$

Lemma 2.15. *Let* $q \in \mathsf{P}_{\mathbb{C}}$ *with* $q \neq 0$ *and let* $\gamma \in \mathsf{P}_{\mathbb{C}}$ *with* $\gamma \notin \{0\} \cup -q/2 + iq\mathbb{R}$. *There is a polynomial-time algorithm that takes as an input:*

- *a positive integer* n;
- *a theta graph* $G = \Theta_{(l_1,\dots,l_m)}$ *with terminals* s *and* t.

This algorithm computes $f(n,G)$ *such that*

1. $f(n,G) = n+\Theta(\mathrm{size}(G))$;
2. *for any* $\hat{\gamma}$ *with* $|\gamma-\hat{\gamma}| \leq 2^{-f(n,G)}$, *we have* $|w(G,s,t;q,\gamma) - w(G,s,t;q,\hat{\gamma})| \leq 2^{-n}$.

Proof. Let $y = \gamma+1$ and $x = 1+q/\gamma$. Note that $|x|=1$ if and only if $|\gamma+q| = |\gamma|$. By basic geometry, the latter statement is equivalent to $\gamma \in -q/2+iq\mathbb{R}$. Hence, by hypothesis, $|x| \neq 1$. There are two cases:

- $|x|<1$. Then for any positive integer k we have

$$\left|1+\frac{q}{x^k-1}\right| \leq 1+\frac{|q|}{1-|x|^k} \leq 1+\frac{|q|}{1-|x|} = 1+\frac{|q|}{|1-|x||}.$$

- $|x| > 1$. Then for any positive integer k we have

$$\left|1 + \frac{q}{x^k - 1}\right| \leq 1 + \frac{|q|}{|x|^k - 1} \leq 1 + \frac{|q|}{|x| - 1} = 1 + \frac{|q|}{|1 - |x||}.$$

Since $q, x \in \mathsf{P}_{\mathbb{C}}$, we can apply Lemma 2.13 along with the above bounds to compute a non-negative integer t_x such that $\left|1 + q/(x^k - 1)\right| \leq 2^{t_x}$ for every positive integer k. Lemma 2.13 also allows us to compute non-negative integers t_q and t_γ such that $|q| \leq 2^{t_q}$ and $2^{-t_\gamma} \leq |\gamma|$.

Let n and $G = \Theta_{(l_1,\ldots,l_m)}$ be the inputs of our algorithm. Let $k = \max\{l_1, \ldots, l_m\}$. Since $|x| \neq 1$, we can compute $g(n, G) = r(n + (t_x + 1)(m + 1) + t_q, k)$, where r is as in Lemma 2.14 for the polynomial-time computable number x. We compute $f(n, G) = g(n, G) + t_q + 2t_\gamma + 1$. We claim that f satisfies the statement. In view of the properties of r, we have

$$f(n, G) = g(n, G) + \Theta(1) = n + \Theta(\text{size}(G)).$$

We define $y_j = 1 + q/\left(x^{l_j} - 1\right)$ for every $j \in \{1, \ldots, m\}$. Recall that in (2.4) we argued that

$$w(G, s, t; q, \gamma) = \prod_{j=1}^{m} y_j - 1.$$

Let $\hat{\gamma}$ with $|\gamma - \hat{\gamma}| \leq 2^{-f(n,G)}$. Let $\hat{y} = \hat{\gamma} + 1$ and $\hat{x} = 1 + q/(\hat{y} - 1)$. Then

$$w(G, s, t; q, \hat{\gamma}) = \prod_{j=1}^{m} \hat{y}_j - 1,$$

where $\hat{y}_j = 1 + q/\left(\hat{x}^{l_j} - 1\right)$. Since $|\gamma - \hat{\gamma}| \leq 2^{-f(n,G)} \leq 2^{-t_\gamma - 1}$, we have $|\hat{\gamma}| \geq |\gamma| - 2^{-t_\gamma - 1} \geq 2^{-t_\gamma - 1}$ and

$$|x - \hat{x}| = \left|\frac{q}{\gamma} - \frac{q}{\hat{\gamma}}\right| = \left|q\frac{\hat{\gamma} - \gamma}{\gamma\hat{\gamma}}\right| \leq |q|\,|\hat{\gamma} - \gamma|\,2^{2t_\gamma + 1} \leq 2^{t_q + 2t_\gamma + 1 - f(n,G)} = 2^{-g(n,G)}.$$

In light of the properties of r (Lemma 2.14) and the fact that $l_j \leq k$, it follows that

$$|y_j - \hat{y}_j| = \left|\frac{q}{x^{l_j} - 1} - \frac{q}{\hat{x}^{l_j} - 1}\right| \leq |q|\,2^{-n-(t_x+1)(m+1)-t_q} \leq 2^{-n-(t_x+1)(m+1)}$$

for every $j \in \{1, \ldots, m\}$. Thus, we have $|\hat{y}_j| \leq |y_j| + 1 \leq 2^{t_x+1}$. We obtain

$$\begin{aligned}\left|\prod_{j=1}^{m} y_j - \prod_{j=1}^{m} \hat{y}_j\right| &= \left|\sum_{j=1}^{m} (y_j - \hat{y}_j) \prod_{s=1}^{j-1} \hat{y}_s \prod_{s=j+1}^{m} y_s\right| < \sum_{j=1}^{m} |y_j - \hat{y}_j|\,2^{t_x(m-1)+j-1} \\ &\leq 2^{t_x(m-1)} \sum_{j=1}^{m} 2^{-n-(t_x+1)(m+1)+j-1} \leq 2^{-n-m-2} \sum_{j=1}^{m} 2^j < 2^{-n}.\end{aligned}$$

Equivalently, $|w(G, s, t; q, \gamma) - w(G, s, t; q, \hat{\gamma})| < 2^{-n}$ as we wanted to prove. □

We now prove the main transitivity property of polynomial-time approximate shifts that we will use in our constructions.

Lemma 2.16. *Let $q \in \mathsf{P}_{\mathbb{C}}$ with $q \neq 0$ and let $(x_j, y_j) \in \mathcal{H}_q$ for each $j \in \{1,2,3\}$. Let us assume that the following hypotheses hold:*

1. *x_2 and y_2 are polynomial-time computable;*
2. *$y_2 \notin \{1\} \cup (1 - q/2 + iq\mathbb{R})$;*
3. *there is a polynomial-time approximate shift from (x_1, y_1) to (x_2, y_2);*
4. *there is a polynomial-time approximate theta shift from (x_2, y_2) to (x_3, y_3).*

Then there is a polynomial-time approximate shift from (x_1, y_1) to (x_3, y_3). Moreover, if the polynomial-time approximate shift from (x_1, y_1) to (x_2, y_2) is series-parallel, then the polynomial-time approximate shift from (x_1, y_1) to (x_3, y_3) is also series-parallel.

Proof. Let $\gamma_j = y_j - 1$ for every $j \in \{1,2,3\}$. Let n be a positive integer. We give an algorithm that constructs a graph J_n, in polynomial time in n, such that J_n γ_1-implements $\hat{\gamma}_3$ with $|\gamma_3 - \hat{\gamma}_3| \leq 2^{-n}$. This algorithm is as follows. First, we use the approximate theta shift from (x_2, y_2) to (x_3, y_3) to compute a theta graph G_2 with terminals s_2 and t_2 such that

$$|\gamma_3 - w(G_2, s_2, t_2; q, \gamma_2)| \leq 2^{-n-1}. \tag{2.5}$$

The size of G_2 is at most polynomial in n. In light of Lemma 2.15, we can compute, in polynomial time in n, a positive integer $f(n+1, G_2)$ such that for any $\hat{\gamma}_2$ with $|\gamma_2 - \hat{\gamma}_2| \leq 2^{-f(n+1,G_2)}$, we have

$$|w(G_2, s_2, t_2; q, \gamma_2) - w(G_2, s_2, t_2; q, \hat{\gamma}_2)| \leq 2^{-n-1}. \tag{2.6}$$

We also have $f(n+1, G_2) = n + \Theta(\text{size}(G_2))$, so $f(n+1, G_2)$ is bounded by a polynomial in n. Now we use the approximate shift from (x_1, y_1) to (x_2, y_2) to compute, in polynomial time in n, a graph G_1 such that G_1 γ_1-implements $\hat{\gamma}_2$ with $|\gamma_2 - \hat{\gamma}_2| \leq 2^{-f(n+1,G_2)}$. Combining (2.5) and (2.6) with the triangle inequality, we obtain $\left|\gamma_3 - w(G_2, s_2, t_2; q, \hat{\gamma}_2)\right| \leq 2^{-n}$.

Finally, we construct a graph J_n as a copy of G_2 where every edge is substituted by a copy of G_1 as in Lemma 2.6. Since the sizes of G_1 and G_2 are polynomial in n, the size of J_n also is polynomial in n. Recall that G_2 $\hat{\gamma}_2$-implements $\hat{\gamma}_3 = w(G_2, s_2, t_2; q, \hat{\gamma}_2)$ and G_1 γ_1-implements $\hat{\gamma}_2$. Therefore, the graph J_n γ_1-implements $\hat{\gamma}_3$, and $|\gamma_3 - \hat{\gamma}_3| \leq 2^{-n}$, as we wanted to obtain. Finally, if the polynomial-time approximate shift from (x_1, y_1) to (x_2, y_2) is series-parallel, then the graphs J_n are easily seen to be series-parallel, and the result follows. □

2.4 Polynomial-time approximate shifts with complex weights

In this section we show how to implement approximations of real weights when the original weight is a non-real algebraic number. As a consequence of our results, for any real algebraic number q with $q \geq 2$ and any pair of algebraic numbers $(x, y) \in \mathcal{H}_q$ with $y \notin \mathbb{R}$ and $(x, y) \notin \{(-i, i), (i-i), (\omega_3^2, \omega_3), (\omega_3, \omega_3^2)\}$, where $\omega_3 = \exp(2\pi i/3)$, there is a polynomial-time approximate

shift from (x, y) to any pair of real algebraic numbers $(x', y') \in \mathcal{H}_q$ (see Theorem 2.2). Our approach to prove Theorem 2.2 is as follows. First, we show that there is $(x', y') \in \mathcal{H}_q$ with $y' \in (0, 1)$ such that there is a polynomial-time approximate theta shift from (x, y) to (x', y') (see Lemma 2.32). Since x and y are algebraic, it follows that x' and y' are polynomial-time computable. Secondly, we extend part of Lemma 2.7 to the case where the numbers involved are only known to be polynomial-time computable (see Lemma 2.4). Finally, we use the transitivity property given in Lemma 2.16 to combine both results in the proof of Theorem 2.2.

2.4.1 Some algorithms for algebraic numbers

In our proofs we use and develop some specific algorithms on algebraic numbers. We gather these algorithms in this section. The first non-trivial operation that we can perform (other than the standard ones described in the preliminaries - Section 2.2.3) is checking if the argument of an algebraic number is in a fixed interval.

Lemma 2.17. *Let $a, b \in \mathbb{Q} \cap [0, 1]$ with $a \leq b$. Then for any algebraic number z we can check whether $\operatorname{Arg}(z) \in [2\pi a, 2\pi b]$ in time polynomial in the length of the representation of z.*

Proof. We can split the interval $[2\pi a, 2\pi b]$ into intervals of length at most $\pi/2$ and check if $\operatorname{Arg}(z)$ belongs to any of those intervals. Hence, let us assume for the sake of simplicity that $[2\pi a, 2\pi b] \subseteq [0, \pi/2]$. The other cases are analogous. Note that $e^{2\pi i a}$ and $e^{2\pi i b}$ are roots of unity and, in particular, algebraic. Thus, we can compute $z_a = ze^{-2\pi i a}$ and $z_b = ze^{2\pi i(1/4-b)}$. We have $\operatorname{Arg}(z_a) \in [0, \pi/2]$ if and only if $\operatorname{Arg}(z) \in [2\pi a, \pi/2 + 2\pi a]$, and $\operatorname{Arg}(z_b) \in [0, \pi/2]$ if and only if $\operatorname{Arg}(z) \in [-\pi/2 + 2\pi b, 2\pi b]$. We conclude that $\operatorname{Arg}(z) \in [2\pi a, 2\pi b]$ if and only if $\operatorname{Arg}(z_a) \in [0, \pi/2]$ and $\operatorname{Arg}(z_b) \in [0, \pi/2]$. Finally, note that, for any algebraic number y, since $\operatorname{Re}(y)$ and $\operatorname{Im}(y)$ are algebraic, we can determine if $\operatorname{Arg}(y) \in [0, \pi/2]$ or not by checking the inequalities $\operatorname{Re}(y) \geq 0$ and $\operatorname{Im}(y) \geq 0$. □

In the rest of this section we show how to efficiently compute a sequence $\sigma(n)$ such that $\operatorname{Arg}(z^{\sigma(n)}) \in [2\pi a, 2\pi b]$ for every n. We will use the following well-known result, see, e.g., [23, Section 1.2]: if $z \in \mathbb{C}$ is not a root of unity and $|z| = 1$, then $\{z^j : j \in \mathbb{N}\}$ is dense in the unit circle.

Lemma 2.18. *Let $a, b \in \mathbb{Q} \cap [0, 1]$ with $a < b$. Let z be an algebraic number such that $|z| = 1$ and z is not a root of unity. Then there exists a sequence of positive integers $\{\sigma(n)\}$ and a positive integer k such that such that:*

1. *k can be computed from z;*
2. *$\sigma(n)$ can be computed in polynomial time in n;*
3. *$n \leq \sigma(n) \leq n + k - 1$ for every positive integer n;*
4. *$\operatorname{Arg}(z^{\sigma(n)}) \in [2a\pi, 2b\pi] + 2\pi\mathbb{Z}$ for every positive integer n.*

Proof. Our algorithm to compute $\sigma(n)$ is as follows. Set $\sigma(0) = 0$. We compute $\sigma(n)$ as the smallest integer such that $n \leq \sigma(n)$ and $\mathrm{Arg}(z^{\sigma(n)}) \in [2a\pi, 2b\pi]$. We can check whether $\mathrm{Arg}(z^{\sigma(n)}) \in [2a\pi, 2b\pi]$ or not by applying the procedure given in Lemma 2.17.

We show that $\sigma(n)$ is well-defined. Let $\theta = \mathrm{Arg}(z)$. Since z is not a root of unity, $\{z^j : j \in \mathbb{N}\}$ is dense in the unit circle, as we have discussed in the previous paragraph. Therefore, there is $q \in \mathbb{N}$ such that $\mathrm{Arg}(z^q) \in [0, 2(b-a)\pi]$. Note that we can compute q in constant time with the help of Lemma 2.17. Let $\tau = \mathrm{Arg}(z^q)$. Since z is not a root of unity, we find that $\tau \neq 0$. Let $t = \lceil 2\pi/\tau \rceil$. Since t is the smallest positive integer such that $t\tau \geq 2\pi$, t can be computed by sequentially determining which of the following intervals contains the argument of z^{qj}: $(0, \pi/2)$, $(\pi/2, \pi)$, $(\pi, 3\pi/2)$ or $(3\pi/2, 2\pi)$. Hence, we can compute $k = tq$. For each positive integer n, since $t\tau \geq 2\pi$ and $\tau < 2(b-a)\pi$, there is $p_n \in \{0, \ldots, t-1\}$ such that $n\theta + p_n\tau \in [2a\pi, 2b\pi] + 2\pi\mathbb{Z}$. The integer $m_n = n + p_n q$ satisfies $n \leq m_n \leq n + k - 1$ and

$$m_n\theta \in n\theta + p_n\tau + 2\pi\mathbb{Z} \subseteq [2a\pi, 2b\pi] + 2\pi\mathbb{Z}.$$

We conclude that $\sigma(n)$ is well-defined and $n \leq \sigma(n) \leq m_n \leq n+k-1$, so our algorithm computes $\sigma(n)$ in polynomial time in n. □

Lemma 2.19. *Let z be a root of unity of order k with $k \notin \{1, 2, 4\}$. Then there exists a sequence of positive integers $\{\sigma(n)\}$ and an integer l such that:*

1. *$\sigma(n)$ can be computed in polynomial time in n;*
2. *$n \leq \sigma(n) \leq n + k - 1$ for every positive integer n;*
3. *$z^{\sigma(n)} = e^{2\pi i l/k}$ for every positive integer n;*
4. *$\pi < 2\pi l/k < 3\pi/2$.*

Proof. Let $\theta = \mathrm{Arg}(z)$. Since $\theta \neq 0$, we can write $\theta = 2\pi j/k$ for some integer j coprime with k. We consider two cases.

Case I: $\mathbf{k = 3}$. Then either we have $\theta = 2\pi/3$ and we compute $\sigma(n) \in \{n, n+1, n+2\}$ with $\sigma(n) \equiv 2 \pmod 3$, or we have $\theta = 4\pi/3$ and we compute $\sigma(n) \in \{n, n+1, n+2\}$ with $\sigma(n) \equiv 1 \pmod 3$. In any case, we have $\sigma(n)\theta \in 4\pi/3 + 2\pi\mathbb{Z}$, that is, $z^{\sigma(n)} = e^{4\pi i/3}$ for any positive integer n.

Case II: $\mathbf{k \geq 5}$. Then there is an integer l such that $k/2 < l < 3k/4$, that is, $2\pi l/k \in (\pi, 3\pi/2)$. The Euclidean algorithm gives two integers t_1, t_2 such that $t_1 j + t_2 k = 1$. We compute $\sigma(n) \in \{n, \ldots, n+k-1\}$ such that $\sigma(n) \equiv t_1 l \pmod k$. We can write $\sigma(n) = t_1 l + q_n k$ for some integer q_n. We have

$$\sigma(n)\theta = t_1 l \frac{2\pi j}{k} + q_n 2\pi j = l(1 - t_2 k)\frac{2\pi}{k} + q_n 2\pi j = \frac{2\pi l}{k} + (q_n j - l t_2)2\pi$$

and, equivalently, $z^{\sigma(n)} = e^{2\pi i l/k}$ for every positive integer n. □

Corollary 2.20. *Let z be an algebraic number such that $z \notin \mathbb{R} \cup i\mathbb{R}$. Let $\theta = \mathrm{Arg}(z)$. Then there exists a sequence of positive integers $\{\sigma(n)\}$, a positive integer k and a positive rational number C such that such that:*

1. *k and C can be computed from z;*
2. *$\sigma(n)$ can be computed in polynomial time in n;*
3. *$n \leq \sigma(n) \leq n + k - 1$ for every positive integer n;*
4. *$\cos(\sigma(n)\theta) \leq -C$ and $\sin(\sigma(n)\theta) \leq -C$ for every positive integer n.*

Proof. We may assume that $|z| = 1$ since, otherwise, we can compute the algebraic number $z/|z|$ and apply the following algorithm to this quantity. We invoke either Lemma 2.18 for $a = 7/12$ and $b = 8/12$ or Lemma 2.19, depending on whether z is a root of unity or not, which can be checked as explained at the beginning of this section. In any case, we find a sequence σ and a positive integer k that satisfy the first three assertions announced in the statement. In the non-root of unity case, we have $\cos(\sigma(n)\theta) \leq \cos(2\pi b) < 0$ and $\sin(\sigma(n)\theta) \leq \sin(2\pi a) < 0$ for every positive integer n. In the root of unity case, the sequences $\cos(\sigma(n)\theta)$ and $\sin(\sigma(n)\theta)$ are negative constants. In any case, we can compute a positive rational number C such that $\cos(\sigma(n)\theta) \leq -C$ and $\sin(\sigma(n)\theta) \leq -C$ for every positive integer n. □

Corollary 2.21. *Let z be an algebraic number with $|z| > 1$. Then for any $x \in \mathbb{Q}$ with $x > 0$, we can compute n such that $\mathrm{Re}\,(z^n) \geq x$. Moreover, if $z \notin [0, \infty)$, then we can compute m such that $\mathrm{Re}\,(z^m) \leq -x$.*

Proof. Let $z = Re^{i\theta}$ for some $\theta \in [0, 2\pi)$ and $R > 1$. We determine if $z/|z| = e^{i\theta}$ is a root of unity or not, and compute its order as explained before. If $e^{i\theta}$ is a root of unity of order k, then $z^k \in (1, \infty)$, so computing n is straightforward. If $e^{i\theta}$ is not a root of unity, then, in view of Lemma 2.19 for $a = 1/12$ and $b = 1/6$, we can compute a sequence σ such that $\sigma(j) \geq j$ and $\sigma(j)\theta \in [\pi/6, \pi/3] + 2\pi\mathbb{Z}$ for every positive integer j. We find that $\mathrm{Re}(z^{\sigma(j)}) \geq R^{\sigma(j)} \cos(\pi/3) \geq R^j/2$. Hence, we can compute j large enough such that $\mathrm{Re}(z^{\sigma(j)}) \geq x$ and we choose $n = \sigma(j)$.

Now let us assume that $z \notin [0, \infty)$. Note that $e^{i\theta} \neq 1$. If $e^{i\theta}$ is a root of unity of order 2 or 4, then the result is trivial. If $\theta \notin \{0, \pi/2, \pi, 3\pi/2\}$, then, by invoking Corollary 2.20, we compute σ and a positive rational number C such that $\sigma(j) \geq j$ and $\cos(\sigma(j)\theta) \leq -C$ for every positive integer j. We find that $\mathrm{Re}(z^{\sigma(j)}) \leq -CR^{\sigma(j)} \leq -CR^j$. Hence, we can compute j large enough such that $\mathrm{Re}(z^{\sigma(j)}) \leq -x$ and we choose $m = \sigma(j)$. □

2.4.2 Some shifts for non-real algebraic numbers

In this section we gather some of the shifts that we use in our proofs. Let q be a real algebraic number with $q \geq 2$ and let $(x, y) \in \mathcal{H}_q$ be a pair of algebraic numbers. We are interested in computing a shift from (x, y) to $(x_1, y_1) \in \mathcal{H}_q$ with $x_1 \notin \mathbb{R}$ and $|x_1| > 1$ whenever possible.

The existence of this shift turns out to be closely related to the hardness of approximating $|Z_{\text{Tutte}}(G;q,\gamma)|$ with $\gamma = y-1$; when we can compute such a shift the approximation problem is #P-hard, as we will see in Section 2.5. Recall that one can evaluate the Tutte polynomial of a graph in polynomial time at any of the points in $\{(-i,i),(i,-i),(\omega_3^2,\omega_3),(\omega_3,\omega_3^2)\}$, where $\omega_3 = \exp(2\pi i/3)$ (see Section 2.5.3). These are the points for which our results fail to construct the desired shift.

The results of this section involve computations that might not run in polynomial time in the length of the representation of the algebraic numbers q, x and y involved. However, when applying these results, the numbers q, x and y are constants and, hence, this will not affect the complexity of our algorithms.

Remark 2.22. *Let q be a positive real number and let $(x,y) \in \mathcal{H}_q$. From $(x-1)(y-1) = q$ it follows that x is real if and only if y is real. Note that $x = 1 + q/(y-1) = (y+q-1)/(y-1)$. As noted in the proof of Lemma 2.15, we find that $|x| = 1$ if and only if $|y+q-1| = |y-1|$, that is, y is on the line $1 - q/2 + i\mathbb{R}$. Moreover, $|x| > 1$ if and only if $\operatorname{Re}(y) > 1 - q/2$. Note that when $q \geq 2$ and $\operatorname{Re}(y) > 0$, we have $\operatorname{Re}(y) > 1 - q/2$ and, thus, $|x| > 1$. These observations will be applied several times in this section.*

Lemma 2.23. *Let q be a real algebraic number with $q \geq 2$. Let x and y be algebraic numbers such that $(x,y) \in \mathcal{H}_q$ and $\operatorname{Arg}(y) \notin \{0, \pi/2, 2\pi/3, \pi, 4\pi/3, 3\pi/2\}$. Then we can compute a theta graph J that (x,y)-implements (x_1,y_1) with $|x_1| > 1$ and $x_1 \notin \mathbb{R}$.*

Proof. We show how to compute n such that $\operatorname{Re}(y^n) > 0$ and $\operatorname{Im}(y^n) > 0$. For such a n, we let $y_1 = y^n$ and $x_1 = 1 + q/(y_1 - 1)$, so Remark 2.22 ensures that $|x_1| > 1$ and $x_1 \notin \mathbb{R}$. Hence, we can return J as the graph with two vertices and n edges joining them. Since y and $|y|$ are algebraic numbers, we can compute the algebraic number $y/|y|$. We can detect if $y/|y|$ is a root of unity or not as explained in Section 2.4.1. There are two cases:

(i) $y/|y|$ is not a root of unity. Then we can apply Lemma 2.18 with $a = 1/12, b = 1/6$ and $z = y^n$ to compute the smallest positive integer n such that $\operatorname{Arg}(y^n) \in [\pi/6, \pi/3]$. Recall that such an integer exists because $\{(y/|y|)^j : j \in \mathbb{N}\}$ is dense in the unit circle. Finally, since $\operatorname{Arg}(y^n) \in [\pi/6, \pi/3]$, we have $\operatorname{Re}(y^n) > 0$ and $\operatorname{Im}(y^n) > 0$.

(ii) $y/|y|$ is a root of unity of order r with $r \geq 5$. Recall that we can compute r by sequentially computing the powers of $y/|y|$ until we obtain 1. Then we have $(y/|y|)^{r+1} = e^{i2\pi/r}$. Note that the real and imaginary parts of $e^{i2\pi/r} = \cos(2\pi/r) + i\sin(2\pi/r)$ are positive. □

Note that the argument given in Lemma 2.23 strongly uses the fact that $q \geq 2$, that is, $1 - q/2 \leq 0$. A proof of a version of Lemma 2.23 with $q \in (0,2)$ is unknown to us. Now we deal with the cases $\operatorname{Arg}(y) \in \{\pi/2, 2\pi/3, 4\pi/3, 3\pi/2\}$, where the exemptions $(-i,i),(i,-i),(\omega_3^2,\omega_3),(\omega_3,\omega_3^2)$ arise. Note that $(-i,i),(i,-i) \in \mathcal{H}_2$ and $(\omega_3^2,\omega_3),(\omega_3,\omega_3^2) \in \mathcal{H}_3$. In fact, one can easily check that these are the only pairs (x,y) such that $|y| = 1$ and $q \in \{2,3\}$.

Lemma 2.24. *Let q be a real algebraic number with $q \geq 2$. Let x and y be algebraic numbers such that $(x, y) \in \mathcal{H}_q$, $y \neq 0$ and $\operatorname{Arg}(y) \in \{2\pi/3, 4\pi/3\}$. If $q \neq 3$ or $|y| \neq 1$, then we can compute a series-parallel graph J that (x, y)-implements (x_1, y_1) with $|x_1| > 1$ and $x_1 \notin \mathbb{R}$.*

Proof. Note that $y/|y|$ is a root of unity of order 3. We have $\operatorname{Re}(y) = |y| \cos(2\pi/3) = -|y|/2 < 0$. Let $x = 1 + q/(y-1)$. We consider three cases.

Case I: $\operatorname{Re}(y) > 1 - q/2$. Then, by Remark 2.22, $|x| > 1$. We return J as the graph with 2 vertices and one edge joining them.

Case II: $\operatorname{Re}(y) < 1 - q/2$. Then $|x| < 1$. Let $y_n = 1 + q/(x^n - 1)$. An n-stretch gives a shift from (x, y) to (x^n, y_n). Since $x \notin \mathbb{R}$, there are infinitely many values of n such that $y_n \notin \mathbb{R}$. Note that y_n converges to $1 - q \in (-\infty, -1]$, and the distance between $1 - q$ and the set of complex points $\{z \in \mathbb{C} : \operatorname{Arg}(z) \in \{\pi/2, 2\pi/3, 4\pi/3, 3\pi/2\}\}$ is larger than 0. Hence, we can compute n such that $\operatorname{Arg}(y_n) \notin \{0, \pi/2, 2\pi/3, \pi, 4\pi/3, 3\pi/2\}$. Since $(x^n, y_n) \in \mathcal{H}_q$, the result follows from applying Lemma 2.23 to (x^n, y_n), the transitivity property of shifts and noticing that the obtained graph is series-parallel.

Case III: $\operatorname{Re}(y) = 1 - q/2$. Note that $q > 2$ because for $q = 2$ we would obtain $\operatorname{Re}(y) = 0$. We distinguish three subcases:

- $|y| > 1$. We compute the smallest positive integer n such that $\operatorname{Arg}(y^n) = 2\pi/3$ and $\operatorname{Re}(y^n) = -|y|^n/2 < 1 - q/2$. The proof is concluded by applying Case II to (x_n, y^n), where $x_n = 1 + q/(y^n - 1)$, the transitivity property of shifts and noticing that the obtained graph is series-parallel.

- $|y| < 1$. We compute the smallest positive integer n such that $|y|^n < q - 2$ and $\operatorname{Arg}(y^n) = 2\pi/3$. We have $\operatorname{Re}(y^n) > 1 - q/2$ (otherwise by applying $\operatorname{Re}(y^n) = -|y|^n/2$ we would find that $|y|^n \geq q - 2$), so $|x_n| > 1$ for $x_n = 1 + q/(x^n - 1)$. We return J as the graph with two vertices and n edges joining them.

- $|y| = 1$. Then $1 - q/2 = \operatorname{Re}(y) = -|y|/2 = -1/2$. It follows that $q = 3$, but this case ($|y| = 1$ and $q \neq 3$) was excluded in the hypothesis.

This finishes the proof. □

Lemma 2.25. *Let q be a real algebraic number with $q \geq 2$. Let y be an algebraic number such that $y \neq 0$ and $\operatorname{Arg}(y) \in \{\pi/2, 3\pi/2\}$.*

1. *If $q > 2$, then we can compute a theta graph J that (x, y)-implements (x_1, y_1) with $|x_1| > 1$ and $x_1 \notin \mathbb{R}$.*

2. *If $q = 2$ and $|y| \neq 1$, then we can compute a series-parallel graph J that (x, y)-implements (x_2, y_2) with $y_2 \in (-1, 0)$.*

Proof. The hypotheses $y \neq 0$ and $\operatorname{Arg}(y) \in \{\pi/2, 3\pi/2\}$ are equivalent to $y \neq 0$ and $\operatorname{Re}(y) = 0$. Let $x = 1 + q/(y-1)$. If $q > 2$, then $1 - q/2 < 0 = \operatorname{Re}(y)$ and $|x| > 1$ as a consequence of

Remark 2.22, so we return the graph with two vertices and one edge joining them as J. The second claim (case $q = 2$) has been studied in [55, Lemma 3.15], where the graph constructed is a 2-thickening of a k-stretching. □

Corollary 2.26. *Let q be a real algebraic number with $q \geq 2$. Let x and y be algebraic numbers such that $(x, y) \in \mathcal{H}_q$, $y \notin (-\infty, -1] \cup [0, \infty)$ and $(x, y) \notin \{(i, -i), (-i, i), (\omega_3, \omega_3^2), (\omega_3^2, \omega_3)\}$, where $\omega_3 = \exp(2\pi i/3)$. Then we can compute $(x_2, y_2) \in \mathcal{H}_q$ and a series-parallel graph J such that $|x_2| < 1$ and J (x, y)-implements (x_2, y_2).*

Proof. First, we assume that $y \notin \mathbb{R}$. The case $q = 2$ and $y \in \mathbb{R}i$ is covered in Lemma 2.25, so we assume that $q \neq 2$ or $y \notin \mathbb{R}i$. By applying Lemmas 2.23, 2.24 or 2.25 (depending on the argument of y), we can compute a graph J that (x, y)-implements (x_1, y_1) with $|x_1| > 1$ and $x_1 \notin \mathbb{R}$. We apply Corollary 2.21 with $z = x_1$ in order to compute n such that $\mathrm{Re}(x_1^n) > 1$. A n-stretching of (x_1, y_1) gives a shift from (x_1, y_1) to $(\hat{x}, \hat{y})$, where $\hat{x} = x_1^n$ and $\hat{y} = 1 + q/(\hat{x} - 1)$. We have $\mathrm{Re}(\hat{y}) = 1 + q(\mathrm{Re}(\hat{x}) - 1)/|\hat{x} - 1|^2 > 1$, so $|\hat{y}| > 1$. There are two cases:

- $\hat{y} \notin \mathbb{R}$. We apply Corollary 2.21 with $z = \hat{y}$ to compute t such that $\mathrm{Re}(\hat{y}^t) < 1 - q/2 < 0$. We set $y_2 = \hat{y}^t$ and $x_2 = 1 + q/(y_2 - 1)$. By the transitivity property of shifts, we have a shift from (x, y) to (x_2, y_2). Since $\mathrm{Re}(y_2) < 1 - q/2$, we conclude that $|x_2| < 1$ (Remark 2.22).
- $\hat{y} \in \mathbb{R}$. Hence, we have $\hat{y} \in (1, \infty)$. We can compute a positive integer l such that the norm of $y' = \hat{y}^l y$ is larger than 1. Note that $y' = \hat{y}^l y \notin \mathbb{R}$. A parallel composition yields a shift from (x, y) to (x', y'), where $x' = 1 + q/(y' - 1)$. We compute the graph J by applying the previous case to (x', y').

Now we deal with the case $y \in (-1, 0)$. A 2-thickening gives us a shift from (x, y) to (a_1, b_1), where $b_1 = y^2 \in (0, 1)$ and $a_1 = 1 + q/(b_1 - 1) < 1 - q \leq -1$. A 2-stretching gives us a shift from (a_1, b_1) to (a_2, b_2), where $a_2 = a_1^2 > 1$ and $b_2 = 1 + q/(a_2 - 1) > 1$. We compute a positive integer j such that $b_2^j y < -q$ and, with the help of a j-thickening, construct a shift from (a_2, b_2) to (a_3, b_3) with $b_3 = b_2^j$. The transitivity property of shifts allows us to construct a shift from (x, y) to (a_3, b_3). To conclude the proof, we apply a parallel composition between the latter shift and the identity shift from (x, y) to (x, y), obtaining a shift from (x, y) to (x_2, y_2) with $y_2 = b_3 y$. Recall that $b_3 y = b_2^j y < -q$, so $q/(y_2 - 1) \in (-1, 0)$ and $x_2 = 1 + q/(y_2 - 1) \in (0, 1)$.

Finally, note that the graphs considered in this proof are series-parallel. □

2.4.3 An approximate shift to $(0, 1 - q)$

In Lemma 2.27 and Corollary 2.28 we give a polynomial-time approximate series-parallel shift from (x, y) to $(0, 1 - q)$ under certain conditions.

Lemma 2.27. *Let $q \in \mathsf{P}_{\mathbb{R}}$ with $q > 0$. Let $(x, y) \in \mathcal{H}_q$ such that $x, y \in \mathsf{P}_{\mathbb{C}}$ and $\mathrm{Re}(y) < 1 - q/2$. Then there is a polynomial-time approximate theta shift from (x, y) to $(0, 1 - q)$.*

Proof. Let $x = 1 + q/(y-1)$. In light of Remark 2.22, we have $|x| < 1$. Therefore, the weight $y_j = 1 + q/(x^j - 1)$ implemented by an j-stretch converges to $1 - q$ as $j \to \infty$. We have

$$|q - 1 + y_j| = \left| \frac{qx^j}{x^j - 1} \right| \leq \frac{q\,|x|^j}{1 - |x|^j} \leq \frac{q\,|x|^j}{1 - |x|}. \tag{2.7}$$

We use (2.7) to give a a polynomial-time approximate theta shift from (x, y) to $(0, 1-q)$. Let n be a positive integer, so the desired accuracy of the quantity in (2.7) is 2^{-n}. We are going to return a path graph with j edges for j large enough. It remains to show how to compute j from n. Since $q, |x| \in \mathsf{P}_{\mathbb{R}}$, we can compute $b, c \in \mathbb{Q}$ such that $q \leq c$ and $0 < b \leq 1 - |x|$ (Lemma 2.13). Hence, $|x| \leq 1 - b < 1$, and it suffices to compute j with $j \geq \log_{1-b}(2^{-n}b/c)$. □

Corollary 2.28. *Let q be a real algebraic number with $q \geq 2$. Let x and y be algebraic numbers such that $(x, y) \in \mathcal{H}_q$, $y \notin (-\infty, -1] \cup [0, \infty)$ and $(x, y) \notin \{(-i, i), (i, -i), (\omega_3^2, \omega_3), (\omega_3, \omega_3^2)\}$, where $\omega_3 = \exp(2\pi i/3)$. Then there is a polynomial-time approximate series-parallel shift from (x, y) to $(0, 1-q)$.*

Proof. From Corollary 2.26 we obtain a shift from (x, y) to (x_2, y_2) with $|x_2| < 1$ or, equivalently, $\mathrm{Re}(y_2) < 1 - q/2$. The result follows from applying Lemma 2.27 to (x_2, y_2) and the transitivity property of shifts. □

2.4.4 An approximate shift to (x', y') with $y' \in (0, 1)$

In Lemma 2.31 we show that if a sequence z_n of complex numbers has certain properties, then there is $w \in (0, 1) \cap \mathsf{P}_{\mathbb{R}}$ that is the limit of $\prod_{j=1}^{n} z_j^{e_j}$ for some non-negative integers $e_1, e_2, \ldots$ that we can compute. Then we apply this result to a subsequence of $\{y_n\}$, where (x^n, y_n) is the pair implemented by an n-stretch of (x, y), obtaining a polynomial-time approximate theta shift from (x, y) to some (x', y') with $y' \in (0, 1)$ (Lemma 2.32). First, we need the following elementary results.

Lemma 2.29. *We have $\sin(x) \leq x \leq \pi \sin(x)/2$ for every $x \in [0, \pi/2]$.*

Proof. First, we prove that $\sin(x) \leq x$ for every $x \in [0, \pi/2]$. Let $f(x) = x - \sin(x)$. We have $f'(x) = 1 - \cos(x) > 0$ for every $x \in [0, \pi/2]$. Hence, f is strictly increasing in $[0, \pi/2]$. Since $f(0) = 0$, we obtain $x - \sin(x) \geq 0$ for every $x \in [0, \pi/2]$.

Now we prove that $x \leq \pi \sin(x)/2$ for every $x \in [0, \pi/2]$. Let $g(x) = \pi \sin(x)/2 - x$ for every $x \in [0, \pi/2]$. We have $g'(x) = \pi \cos(x)/2 - 1$. Let $y \in [0, \pi/2]$ such that $\cos(y) = 2/\pi$. Note that $g'(x) > 0$ in $[0, y)$, $g(y) = 0$ and $g'(x) < 0$ in $(y, \pi/2]$. Hence, g only reaches a minimum at $x \in \{0, \pi/2\}$. Since $g(0) = g(\pi/2) = 0$, we conclude that $0 \leq \pi \sin(x)/2 - x$ for every $x \in [0, \pi/2]$. □

Lemma 2.30. *Let $z \in \mathbb{C}$. Let $\{z_n\}$ be a sequence of algebraic complex numbers such that:*

1. *we can compute two rational numbers C and R such that $C > 0$, $R \in (0, 1)$ and $|z - z_n| \leq CR^n$ for every positive integer n;*

2. we can compute the representation of the algebraic number z_n in polynomial time in n.

Then $z \in \mathsf{P}_{\mathbb{C}}$, i.e., z is polynomial-time computable.

Proof. Let n be an arbitrary positive integer. For $j = \lceil \log_R \left(2^{-n-1}/C\right) \rceil$ we have $|z - z_j| \leq 2^{-n-1}$. Note that $j = \Theta(n)$ and hence z_j is an algebraic number whose representation we can compute in time polynomial in n. So, we can also compute $\hat{z}_j \in \mathbb{Q}[i]$ such that $|z_j - \hat{z}_j| \leq 2^{-n-1}$ in time polynomial in n. Then, we have that

$$|z - \hat{z}_n| \leq |z_n - z| + |z_n - \hat{z}_n| \leq 2^{-n}.$$

Since n was arbitrary, we have that z is polynomial-time computable. □

Lemma 2.31. *Let $r, c \in (0,1) \cap \mathbb{Q}$. Let $\{z_n\}$ be a sequence of algebraic complex numbers with:*

1. *$|z_n| < 1$ for every positive integer n;*
2. *$z_n = 1 - f(n) + ig(n)$ with $f, g \colon \mathbb{Z}^+ \to (0,1)$;*
3. *$cr^n \leq f(n) \leq r^n/2$ and $cr^n \leq g(n) \leq r^n/2$ for every positive integer n.*

Then there is $w \in (0,1)$ and a bounded sequence of positive integers $\{e_n\}$ such that

$$\left|\prod_{j=1}^{n} z_j^{e_j} - w\right| \leq \left(\frac{\pi}{2} + \frac{\pi}{c(1-r)}\right) r^n$$

for every positive integer n. Moreover, if the representation of the algebraic number z_n can be computed in polynomial time in n, then $w \in \mathsf{P}_{\mathbb{R}}$ and e_n can be computed in polynomial time in n.

Proof. We can write $z_n = \rho_n e^{i\theta_n}$ for some $\rho_n \in (0,1)$ and $\theta_n \in (0, \pi/2)$. Note that $1 - f(n) < \rho_n$. Let $h(n) = 1 - \rho_n$. We obtain

$$0 < h(n) < f(n) \leq r^n/2 \tag{2.8}$$

for every positive integer n. We have

$$\sin(\theta_n) = \frac{\mathrm{Im}(z_n)}{\rho_n} = \frac{g(n)}{1 - h(n)}.$$

In view of Lemma 2.29, we obtain

$$\frac{g(n)}{1-h(n)} \leq \theta_n \leq \frac{\pi g(n)}{2(1-h(n))}.$$

Since $0 < h(n) \leq 1/2$ (see (2.8)), it follows that

$$g(n) \leq \theta_n \leq \pi g(n). \tag{2.9}$$

As a consequence, we find that, for any integer n with $n \geq 2$,

$$\frac{\theta_{n-1}}{\theta_n} \leq \pi \frac{g(n-1)}{g(n)} \leq \frac{\pi}{2cr}, \tag{2.10}$$

where we used the fact that $cr^n \leq g(n) \leq r^n/2$. The bounds (2.8), (2.9) and (2.10) will be used several times in this proof.

Let $\tau_0 = 0$. We define τ_n and e_n by induction on n. Let e_n be the largest integer such that $\tau_{n-1} + e_n\theta_n \leq 2\pi$ and let $\tau_n = \tau_{n-1} + e_n\theta_n$. By definition, $\{\tau_n\}$ is an increasing sequence that is bounded above by 2π. Moreover, we have $2\pi - \theta_n < \tau_n$, since $\tau_n + \theta_n \leq 2\pi$ contradicts the definition of e_n. That is, we have $0 \leq 2\pi - \tau_n < \theta_n$. We show that e_n is bounded. Note that $e_1 \leq 2\pi/\theta_1 \leq 2\pi/(cr)$, where we used that $cr \leq g(1) \leq \theta_1$ (recall (2.9)). For $n \geq 2$ we have

$$0 \leq e_n = \frac{\tau_n - \tau_{n-1}}{\theta_n} \leq \frac{2\pi - \tau_{n-1}}{\theta_n} < \frac{\theta_{n-1}}{\theta_n} \leq \frac{\pi}{2cr},$$

where we applied (2.10). By combining the latter inequality with the case $n = 1$ we conclude that

$$0 \leq e_n \leq \frac{2\pi}{cr} \tag{2.11}$$

for every positive integer n.

The sequence $\{e^{i\tau_n}\}$ converges to 1. In fact, we show that it does so exponentially fast. Note that the derivative of e^{it} has constant norm 1. Therefore, e^{it} is a Lipschitz function with constant 1, that is, $\left|e^{it} - e^{is}\right| \leq |s - t|$ for every $s, t \in \mathbb{R}$. It follows that

$$\left|1 - e^{i\tau_n}\right| = \left|e^{i2\pi} - e^{i\tau_n}\right| \leq |2\pi - \tau_n| < \theta_n \leq \pi g(n) \leq \frac{\pi}{2} r^n \tag{2.12}$$

for every positive integer n, where we applied (2.9).

Now we study the sequence $\{x_n\}$ for $x_n = \prod_{j=1}^{n} \rho_j^{e_j}$. Since $\rho_j \in (0, 1)$, $\{x_n\}$ is decreasing and has a limit $w \in [0, 1)$. We claim that this is the real number in $(0, 1)$ announced in the statement. First, we prove that $w > 0$. Let $b = \lceil 2\pi/(cr) \rceil$. In view of (2.11), we have

$$x_n \geq \prod_{j=1}^{n} \rho_j^b = \Big(\prod_{j=1}^{n} (1 - h(j)) \Big)^b.$$

Recall that a product of the form $\prod_{j=1}^{n} (1 - a_n)$ with $a_n \in [0, 1)$ converges to a positive number if and only if $\sum_{j=1}^{n} a_n$ converges [111, Proposition 3.1]. From (2.8) we obtain

$$\sum_{n=1}^{\infty} h(n) \leq \frac{1}{2} \sum_{n=1}^{\infty} r^n = \frac{r}{2(1-r)}$$

and, thus, $\prod_{j=1}^{n} (1 - h(j))$ converges to a real number L with $L > 0$. We conclude that $w \geq L^b > 0$, as we wanted to prove. Now we show that $\{x_n\}$ converges exponentially fast to w. Note that $x_n = (1 - h(n))^{e_n} x_{n-1}$ and, thus, for $n \geq 2$, we have

$$\begin{aligned} 0 \leq x_{n-1} - x_n &= x_{n-1} \left(1 - (1 - h(n))^{e_n}\right) \\ &\leq 1 - (1 - h(n))^{e_n} \leq h(n) e_n \leq \frac{\pi}{cr} r^n, \end{aligned}$$

where we used the fact that $(1 - x)^k \geq 1 - kx$ for every $x \in (0, 1)$ and $k \in \mathbb{Z}^+$, and the bounds on $h(n)$ and e_n (see (2.8) and (2.11)). We obtain

$$|x_{n+q} - x_n| \leq \sum_{j=1}^{q} |x_{n+j} - x_{n+j-1}| \leq \frac{\pi}{cr} \sum_{j=1}^{q} r^{n+j} = \frac{\pi\,(1 - r^q)}{c(1-r)} r^n$$

for any positive integers n and q. Hence, by making q tend to ∞ we conclude that

$$|x_n - w| \leq \frac{\pi}{c(1-r)} r^n \tag{2.13}$$

for every positive integer n.

In light of (2.12) and (2.13), we obtain for every positive integer n that

$$\begin{aligned}\left|\prod_{j=1}^{n} z_j^{e_j} - w\right| &\leq \left|\prod_{j=1}^{n} z_j^{e_j} - x_n\right| + |x_n - w| = |x_n| \left|\prod_{j=1}^{n} e^{ie_j\theta_j} - 1\right| + |x_n - w| \\ &\leq \left|\prod_{j=1}^{n} e^{ie_j\theta_j} - 1\right| + |x_n - w| = \left|e^{i\tau_n} - 1\right| + |x_n - w| \leq \frac{\pi}{2} r^n + \frac{\pi}{c(1-r)} r^n.\end{aligned}$$

Finally, we argue that if the representation of z_n can be computed in polynomial time in n, then e_n can be computed in polynomial time in n and we have $w \in \mathsf{P}_{\mathbb{R}}$. Note that e_1 is the smallest positive integer such that $\operatorname{Arg}(z_1^{e_1}) \in [3\pi/2, 2\pi) \cup \{0\}$ and $\operatorname{Arg}(z_1^{e_1+1}) \in (0, \pi/2]$ and, thus, e_1 can be computed by sequentially applying Lemma 2.17 with intervals $[3\pi/2, 2\pi]$ and $[0, \pi/2]$, with the z of Lemma 2.17 equal to z^k for every positive integer $k \leq e_1 + 1$. This takes constant time since the quantities and objects involved are constant. For $n \geq 2$, let us assume that we have computed $e_1, \ldots, e_{n-1}$, and let $y_{n-1} = \prod_{j=1}^{n-1} z_j^{e_j}$ (so $\tau_{n-1} = \operatorname{Arg}(y_{n-1})$). Since the sequence $\{e_n\}$ is bounded and the length of the representation of z_n is bounded by a polynomial in n, the computation of y_{n-1} takes polynomial time in n. Then e_n is the smallest non-negative integer such that $\operatorname{Arg}(y_{n-1} z_n^{e_n}) \in [3\pi/2, 2\pi) \cup \{0\}$ and $\operatorname{Arg}\left(y_{n-1} z_n^{e_n+1}\right) \in (0, \pi/2]$, and we can compute e_n again by sequentially applying Lemma 2.17 with intervals $[3\pi/2, 2\pi]$ and $[0, \pi/2]$, with the z of Lemma 2.17 equal to z^k for every positive integer $k \leq e_n + 1$. There is a bounded number of applications of Lemma 2.17 because e_n is bounded, and each application takes polynomial time in n because the length of the representation of $y_{n-1} z_n^k$ is polynomial in n for any $k \in \{1, 2, \ldots, e_n\}$. We conclude that w is the limit of a sequence of algebraic numbers that converges exponentially fast and the representation of its n-th element can be computed in polynomial time in n. As a consequence, we have $w \in \mathsf{P}_{\mathbb{R}}$ by Lemma 2.30. □

Lemma 2.32. *Let q be a real algebraic number with $q > 0$. Let x and y be algebraic numbers such that $(x, y) \in \mathcal{H}_q$, $y \notin \mathbb{R}$ and $|x| > 1$. Then there is a polynomial-time approximate theta shift from (x, y) to (x', y') for some $(x', y') \in \mathcal{H}_q$ with $y' \in (0, 1) \cap \mathsf{P}_{\mathbb{R}}$.*

Proof. Since $y \notin \mathbb{R}$, we have $x \notin \mathbb{R}$ (Remark 2.22). Let us write $x = Re^{i\theta}$ for some $R > 1$ and $\theta \in (0, 2\pi)$. An m-stretch gives a shift from (x, y) to (x^m, y_m) with $y_m = (x^m + q - 1)/(x^m - 1)$. By plugging $x = Re^{i\theta}$ in the definition of y_m and multiplying by $R^m e^{-im\theta} - 1$ in the numerator and denominator, we obtain

$$y_m = \frac{R^{2m} - q + 1 + (q-2)R^m \cos(m\theta) - iqR^m \sin(m\theta)}{1 + R^{2m} - 2R^m \cos(m\theta)}. \tag{2.14}$$

If $\theta \in \{\pi/2, 3\pi/2\}$, that is, $x \in i\mathbb{R}$, then for $m \equiv 2 \pmod 4$ we have $\cos(m\theta) = -1$, $\sin(m\theta) = 0$ and

$$y_m = \frac{(1 + R^m)^2 - q(1 + R^m)}{(1 + R^m)^2} = \frac{1 + R^m - q}{1 + R^m}.$$

Hence, for $m \equiv 2 \pmod 4$ such that $1 + R^m > q$, we have $y_m \in (0, 1)$, so we can choose $y' = y_m$ and we are done.

In the rest of the proof we assume that $\theta \notin \{\pi/2, 3\pi/2\}$. We are going to apply Lemma 2.31 to a subsequence of y_m. First, we invoke Corollary 2.20 with $z = x$ in order to find a sequence $\sigma(m)$, a positive integer k and a positive rational C that satisfies:

- $\sigma(m)$ can be computed in polynomial time in m;
- k and C can be computed in constant time from x;
- $m \leq \sigma(m) \leq m + k - 1$ for every positive integer m;
- $\sin(\sigma(m)\theta) \leq -C$ and $\cos(\sigma(m)\theta) \leq -C$ for every positive integer m.

It follows that

$$\operatorname{Re}\left(x^{\sigma(m)}\right) = \operatorname{Re}\left(R^{\sigma(m)} e^{i\sigma(m)\theta}\right) \leq -CR^{\sigma(m)} \leq -CR^m.$$

Since $R > 1$, we can compute a positive integer m_1 such that for $m \geq m_1$ we have $\operatorname{Re}(x^{\sigma(m)}) < 1 - q/2$ and, thus, $\left|y_{\sigma(m)}\right| < 1$ (recall that $y_m = (x^m + q - 1) / (x^m - 1)$ and Remark 2.22). Let

$$a_m = 1 - \operatorname{Re}(y_m) = \frac{q - qR^m \cos(m\theta)}{1 + R^{2m} - 2R^m \cos(m\theta)};$$
$$b_m = \operatorname{Im}(y_m) = \frac{-qR^m \sin(m\theta)}{1 + R^{2m} - 2R^m \cos(m\theta)};$$

that is, $y_m = 1 - a_m + ib_m$. We have

$$R^{2\sigma(m)} \leq 1 + R^{2\sigma(m)} - 2R^{\sigma(m)} \cos(\sigma(m)\theta) \leq 4R^{2\sigma(m)}.$$

Therefore, we obtain

$$\frac{qC}{4} R^{-\sigma(m)} \leq a_{\sigma(m)} \leq 2qR^{-\sigma(m)}, \qquad \frac{qC}{4} R^{-\sigma(m)} \leq b_{\sigma(m)} \leq qR^{-\sigma(m)}. \tag{2.15}$$

We compute a positive integer m_2 such that $m_2 \geq \log_R(4q)$ and $m_2 \geq m_1$. We also compute a rational number c with $c \in (0, qCR^{-m_2-k-1}/4)$. Note that computing these quantities takes constant time. Let $f(m) = a_{\sigma(m+m_2)}$ and $g(m) = b_{\sigma(m+m_2)}$. In view of (2.15) and the inequalities $R^{-m-k+1} \leq R^{-\sigma(m)} \leq R^{-m}$, we find that

$$cR^{-m} \leq f(m) \leq \frac{1}{2} R^{-m}, \qquad cR^{-m} \leq g(m) \leq \frac{1}{2} R^{-m}, \tag{2.16}$$

for any positive integer m. The sequence $\{z_m\} = \{y_{\sigma(m+m_2)}\}$ satisfies

- $\left|z_m\right| < 1$ for every positive integer m;
- $z_m = 1 - f(m) + ig(m)$ with $f, g \colon \mathbb{Z}^+ \to (0, 1)$;
- f and g are bounded as in (2.16).

- z_m is an algebraic number whose representation can be computed in polynomial time in m. This is due to the facts that $z_m = (x^{\sigma(m+m_2)} + q - 1)/(x^{\sigma(m+m_2)} - 1)$, $\sigma(m)$ can be computed in polynomial time in m, and $\sigma(m) = O(m)$.

Therefore, we can apply Lemma 2.31 to the sequence $\{z_m\}$ for $r = R^{-1}$. There are $y' \in (0,1) \cap \mathsf{P}_{\mathbb{R}}$ and a bounded sequence of positive integers $\{e_m\}$ such that

$$\left|\prod_{j=1}^{m} z_j^{e_j} - y'\right| \leq \left(\frac{\pi}{2} + \frac{\pi}{c(1-1/R)}\right) R^{-m}$$

for every positive integer m. Moreover, we can compute e_m in polynomial time in m. Let $M = \pi/2 + \pi/(c(1-1/R))$. For any positive integer n, we can compute an integer m with $m \geq \log_{1/R}(2^{-n}/M)$ and $m = \Theta(n)$ in polynomial time in n. We obtain

$$\left|\prod_{j=1}^{m} z_j^{e_j} - y'\right| \leq 2^{-n}.$$

This gives the following polynomial-time approximate theta shift from (x, y) to (x', y'), where $x' = 1 + q/(y' - 1)$. For each positive integer n we return a graph J_n that is the parallel composition of the path graphs that are used to implement the weights $y_{\sigma(j+m_2)}$, each one repeated e_j times, for $j \in \{1, \ldots, m\}$. The graph J_n (x, y)-implements $(\hat{x}, \hat{y}) \in \mathcal{H}_q$ for $\hat{y} = \prod_{j=1}^{m} z_j^{e_j} = \prod_{j=1}^{m} y_{\sigma(j+m_2)}^{e_j}$. □

Lemma 2.3. *Let q be a real algebraic number with $q \geq 2$. Let x and y be algebraic numbers such that $(x, y) \in \mathcal{H}_q$, $y \in (-1, 0) \cup (\mathbb{C} \backslash \mathbb{R})$ and $(x, y) \notin \{(i, -i), (-i, i), (\omega_3, \omega_3^2), (\omega_3^2, \omega_3)\}$, where $\omega_3 = \exp(2\pi i/3)$. Then there is a polynomial-time approximate series-parallel shift from (x, y) to (x', y') for some $(x', y') \in \mathcal{H}_q$ with $x', y' \in \mathsf{P}_{\mathbb{R}}$ and $y' \in (0, 1)$.*

Proof. If $y \in (-1, 0)$, then a 2-thickening of (x, y) gives the result. Hence, let us assume that $y \notin (-1, 0)$ in the rest of the proof. There are two cases:

- $q \neq 2$ or $y \notin i\mathbb{R}$. We apply either Lemma 2.23, Lemma 2.24 or Lemma 2.25, depending on $\mathrm{Arg}(y)$, to find a shift from (x, y) to $(x_1, y_1) \in \mathcal{H}_q$ with $y_1 \notin \mathbb{R}$ and $|x_1| > 1$. The graph of this shift is series-parallel. Then we apply Lemma 2.32 to obtain a polynomial-time approximate theta shift from (x_1, y_1) to some $(x', y') \in \mathcal{H}_q$ with $y' \in (0, 1) \cap \mathsf{P}_{\mathbb{R}}$. The result follows from the transitivity property of shifts.

- $q = 2$ and $y \in i\mathbb{R}$. Since $y \neq \pm i$, Lemma 2.25 gives a shift from (x, y) to (x', y') for some $(x', y') \in \mathcal{H}_q$ with $y' \in (-1, 0)$. A 2-thickening of (x', y') gives the result.

The fact that $x' \in \mathsf{P}_{\mathbb{R}}$ follows from $x' = 1 + q/(y' - 1)$ and $y' \in \mathrm{P}_{\mathbb{R}}$. □

2.4.5 Approximate shifts for polynomial-time computable real numbers

In this subsection we show how we can obtain a polynomial-time approximate shift from (x, y) to (x', y') for any $(x, y), (x', y') \in \mathcal{H}_q$ when $q \geq 2$, $y \in (0, 1) \cap \mathsf{P}_{\mathbb{R}}$ and y' is a positive real

algebraic number (Lemma 2.4). This extends a particular case of Lemma 2.7 to polynomial-time computable numbers. Our proof follows the same approach as that of [60, Lemma 22] but we have to overcome some difficulties that arise when working with the class of numbers $\mathsf{P}_{\mathbb{R}}$. These difficulties will become apparent in the proof, but the reader that is familiar with the literature might want to skip the proof. Then we combine this result and Lemma 2.3 to prove Theorem 2.2, the main result of Section 2.4.

Lemma 2.4. *Let $q, x, y \in \mathsf{P}_{\mathbb{R}}$ such that $q > 0$, $(x, y) \in \mathcal{H}_q$, y is positive and $1 - q/2 < y < 1$. There is a polynomial-time algorithm that takes as an input:*

- *two positive integers k and n, in unary;*
- *a real algebraic number $w \in [y^k, 1]$.*

The algorithm produces a theta graph J that (x, y)-implements $(\hat{x}, \hat{y})$ such that $\left|\hat{y} - w\right| \leq 2^{-n}$. The size of J is at most a polynomial in k and n, independently of w.

Proof. If $w = 1$, then J is the graph with vertices s and t and no edges. In the rest of the proof we focus on the case $w \in (0, 1)$.

Recall that $x = 1 + q/(y - 1)$. Since $q > 0$ and $y \in (1 - q/2, 1)$, we find that $x \in (-\infty, -1)$. Let $y_j = 1 + q/(x^j - 1)$. A j-stretch gives a shift from (x, y) to (x^j, y_j). If j is even, then $x^j > 1$ and $y_j > 1$. Otherwise, $x^j < -1$ and $y_j \in (1 - q/2, 1)$. Moreover, the sequences $\{y_{2j+1}\}$ and $\{y_{2j}\}$ are increasing and decreasing, respectively, and $|y_j - 1|$ can be made exponentially small as a function of j. We use these properties of y_j to show that we can compute $y_{(e_1,\dots,e_m)} = \prod_{j=1}^{m} y_j^{e_j}$ such that $\left|y_{(e_1,\dots,e_m)} - w\right| \leq 2^{-n}$. Let J be the parallel composition of the path graphs that (x, y)-implement (x^j, y_j), each one repeated e_j times, for $j \in \{1, \dots, m\}$. Then J is a theta graph and, in view of (2.4), we have $w(G; q, y - 1) = y_{(e_1,\dots,e_m)} - 1$, that is, J (x, y)-implements $(\hat{x}, \hat{y}) \in \mathcal{H}_q$ with $\hat{y} = y_{(e_1,\dots,e_m)}$. The graph J is the theta graph output by our algorithm.

First, we define a sequence $\{d_j\}$ that will be related to the exponents $e_1, \dots, e_m$. Since $q, x \in \mathsf{P}_{\mathbb{R}}$, we can compute rational upper bounds of q and x (Lemma 2.13) and, with the help of these bounds, a positive integer j_0 such that $j_0 > \log_{|x|} q$. Let $d_j = 0$ for every positive integer j with $j < j_0$ and let $d_j = 0$ for every even positive integer j. For j odd with $j \geq j_0$ we define d_j recursively as the largest non-negative integer such that $y_{(d_1,\dots,d_j)} \geq w$. The integer d_j is well-defined because $0 < y_j < 1$ when j is odd and $j \geq j_0$. An equivalent definition is that $\{d_j\}$ satisfies

$$y_j < w/y_{(d_1,\dots,d_j)} \leq 1 \tag{2.17}$$

for every odd integer j with $j \geq j_0$. A similar sequence $\{d_j\}$ is used in the proofs of [60, Lemma 22] and [58, Lemma 3.28]. For any odd integer m with $m \geq \log_{|x|}(q2^n - 1)$ we have $0 \leq 1 - y_m \leq 2^{-n}$ and, in light of (2.17),

$$0 \leq 1 - w/y_{(d_1,\dots,d_m)} \leq 1 - y_m \leq 2^{-n}.$$

Since $1 \geq y_{(d_1,\dots,d_m)} \geq w$, it follows that

$$\left|w - y_{(d_1,\dots,d_m)}\right| \leq y_{(d_1,\dots,d_m)} 2^{-n} \leq 2^{-n}. \tag{2.18}$$

Now we study the size of the integers $d_1, \dots, d_m$. We bound d_j using an argument given in [58, Lemma 3.28]. First, we show that d_{j_0} is $O(k)$. We have $y_{j_0}^{d_{j_0}} \geq w$. We obtain

$$d_{j_0} \leq \log_{y_{j_0}}(w) = \log_y(w) \log_{y_{j_0}}(y).$$

Since $w \in [y^k, 1)$ and $\log_{y_{j_0}}(y) > 0$, it follows that $0 < \log_y(w) \leq k$ and $d_{j_0} \in O(k)$. Now we show that d_j is bounded for any $j > j_0$. By applying (2.17) twice, we find that

$$y_{j-2} < w / y_{(d_1,\dots,d_{j-2})} = w y_j^{d_j} / y_{(d_1,\dots,d_j)} \leq y_j^{d_j}$$

for every odd integer j with $j > j_0$. It follows that $d_j \leq \log(y_{j-2})/\log(y_j)$ (here and in the rest of this chapter log is taken in base e). For every $x \in (1, 5/4)$, we have $3(x-1)/4 \leq \log(x) \leq x - 1$. Hence, we obtain

$$\begin{aligned} d_j \leq \frac{\log(y_{j-2})}{\log(y_j)} = \frac{\log(1/y_{j-2})}{\log(1/y_j)} &\leq \frac{4}{3} \frac{1/y_{j-2} - 1}{1/y_j - 1} \\ &= \frac{4y_j}{3y_{j-2}} \frac{1 - y_{j-2}}{1 - y_j} = \frac{4y_j}{3y_{j-2}} \frac{|x|^j + 1}{|x|^{j-2} + 1} \leq \frac{4y_j}{3y_{j-2}} |x|^2, \end{aligned}$$

where the last inequality is a consequence of $|x|^2 (|x|^{j-2} + 1) \geq |x|^j + 1$. Since y_j / y_{j-2} converges to 1 and, thus, is bounded, it follows that d_j is bounded. We conclude that $\sum_{j=1}^m d_j = O(k+m)$.

Let us assume that we can compute $d_1, \dots, d_m$ for $m = \lceil 1 + \log_{|x|}(q 2^n - 1) \rceil$. In light of (2.18), we can return J as the theta graph that implements the weight $w(J; q, y-1) = y_{(d_1,\dots,d_m)} - 1$. Since $\sum_{j=1}^m d_j = O(k+m)$ and $m = \Theta(n)$, the size of J is at most a polynomial in k and n.

If y were algebraic, computing $d_1, \dots, d_m$ in polynomial time would be straightforward from their definition because we can efficiently check inequalities between real algebraic numbers as explained in Section 2.4.1. This is the approach followed in [60, Lemma 22]. However, we only know that $y \in \mathsf{P}_{\mathbb{R}}$ and, thus, it is not clear how to efficiently determine whether $y_{(d_1,\dots,d_{m-1},d)} \geq w$ or not for any given d. In the rest of this proof, we show how to overcome this difficulty.

Let n be a positive integer, so 2^{-n} is the desired accuracy for our algorithm. Let us assume that we have computed the integers $d_1, \dots, d_{j-1}$ and we want to compute d_j for an odd positive integer j with $j \geq j_0$. We are going to sequentially try all the values $d = 0, 1, \dots$ until we have

$$y_j < \frac{w}{y_{(d_1,\dots,d_{j-1},d)}} \leq 1,$$

in which case we have found the value d_j (see (2.17)). Recall that $y_{(d_1,\dots,d_{j-1},d)} - 1$ is the weight implemented by a theta graph J_d whose size is bounded by a polynomial in k and j. Therefore, by applying Lemma 2.15 with $G = J_d$ and $\gamma = y - 1$, we can compute in polynomial time in n and the size of J_d, a positive integer $f(n+2, J_d)$ with $f(n+2, J_d) = n + \Theta(\text{size}(J_d))$ such that if $|\gamma - \hat{\gamma}| \leq 2^{-f(n+2,J_d)}$, then $|w(G; q, \gamma) - w(G; q, \hat{\gamma})| \leq 2^{-n-2}$. Since $y \in \mathrm{P}_{\mathbb{R}}$, we can compute

a rational number $\hat{\gamma}$ such that $|\gamma - \hat{\gamma}| \leq 2^{-f(n+2,J_d)}$ in polynomial time in n and the size of J_d. Let $\hat{y}_{(d_1,\dots,d_{j-1},d)} = w(G;q,\hat{\gamma}) + 1$. Then we have computed in polynomial time in k, j and n a rational number $\hat{y}_{(d_1,\dots,d_{j-1},d)}$ such that

$$\left|\hat{y}_{(d_1,\dots,d_{j-1},d)} - y_{(d_1,\dots,d_{j-1},d)}\right| \leq 2^{-n-2}.$$

Because $\left|\hat{y}_{(d_1,\dots,d_{j-1},d)} - w\right|$ is a real algebraic number, we can check if the following inequality holds in polynomial time,

$$\left|\hat{y}_{(d_1,\dots,d_{j-1},d)} - w\right| \leq 2^{-n-1}. \tag{2.19}$$

If that is the case, then

$$\left|y_{(d_1,\dots,d_{j-1},d)} - w\right| \leq \left|y_{(d_1,\dots,d_{j-1},d)} - \hat{y}_{(d_1,\dots,d_{j-1},d)}\right| + \left|\hat{y}_{(d_1,\dots,d_{j-1},d)} - w\right| \leq 3 \cdot 2^{-n}/4 < 2^{-n},$$

so $y_{(d_1,\dots,d_{j-1},d)}$ is a good enough approximation of w and we can stop the algorithm (even though we have not computed d_j). Otherwise, we claim that $\hat{y}_{(d_1,\dots,d_{j-1},d)} \geq w$ if and only if $y_{(d_1,\dots,d_{j-1},d)} \geq w$. If $\hat{y}_{(d_1,\dots,d_{j-1},d)} \geq w$ and $w > y_{(d_1,\dots,d_{j-1},d)}$, then

$$\left|\hat{y}_{(d_1,\dots,d_{j-1},d)} - w\right| \leq \left|\hat{y}_{(d_1,\dots,d_{j-1},d)} - y_{(d_1,\dots,d_{j-1},d)}\right| \leq 2^{-n-2}$$

and (2.19) holds, a contradiction. The same reasoning applies when $\hat{y}_{(d_1,\dots,d_{j-1},d)} < w$ and $w \leq y_{(d_1,\dots,d_{j-1},d)}$. Hence, we can check whether $y_{(d_1,\dots,d_{j-1},d)} \geq w$ or not by checking $\hat{y}_{(d_1,\dots,d_{j-1},d)} \geq w$, provided that (2.19) does not hold. This gives a procedure to compute d_j for odd j with $j \geq j_0$:

1. Set $d = 0$.
2. If (2.19) holds, then return d. We have failed to compute d_j, but we have succeeded in finding an approximation of w.
3. If $\hat{y}_{(d_1,\dots,d_{j-1},d+1)} \geq w$, then increase d by 1 and go to step 2. Else, we have $d_j = d$.

We repeat this procedure to compute d_j sequentially until (2.19) holds, in which case we stop and return the graph J associated to $y_{(d_1,\dots,d_{j-1},d)}$.

It remains to show that this procedure always halts and runs in polynomial time. In light of (2.18), we find that, for odd $m \geq \log_{|x|}(q2^{n+2} - 1)$,

$$\left|\hat{y}_{(d_1,\dots,d_m)} - w\right| \leq \left|\hat{y}_{(d_1,\dots,d_m)} - y_{(d_1,\dots,d_m)}\right| + \left|y_{(d_1,\dots,d_m)} - w\right| \leq 2^{-n-1},$$

that is, (2.19) holds. Therefore, our procedure that computes non-negative integers $d_1, \dots, d_{m-1}, d$ with $\left|y_{(d_1,\dots,d_{m-1},d)} - w\right| \leq 2^{-n}$ halts for $m = O(n)$. As a consequence, the whole procedure runs in polynomial time in k and n. □

The proof of Lemma 2.4 can be adapted to to the case $w \in (1, y^{-k}]$. The main difference is that this time we work with the decreasing sequence $\{y_{2j}\}$. We set $d_j = 0$ for odd j and, for even j, we define d_j recursively as the largest non-negative integer such that $y_{(d_1,\dots,d_j)} \leq w$. The details of the proof are left to the reader. When studying the hardness of approximating $Z_{\text{Tutte}}(G;q,\gamma)$ we only need the version stated in Lemma 2.4.

Theorem 2.2. *Let $q \geq 2$ be a real algebraic number. Let x and y be algebraic numbers such that $(x, y) \in \mathcal{H}_q$, $y \in (-1, 0) \cup (\mathbb{C} \backslash \mathbb{R})$ and $(x, y) \notin \{(i, -i), (-i, i), (\omega_3, \omega_3^2), (\omega_3^2, \omega_3)\}$, where $\omega_3 = \exp(2\pi i/3)$. Then, for any pair of real algebraic numbers $(x', y') \in \mathcal{H}_q$ there is a polynomial-time approximate series-parallel shift from (x, y) to (x', y').*

Proof. First, let us assume that $y' \in (0, 1]$. By Lemma 2.3, there is a polynomial-time approximate series-parallel shift from (x, y) to $(\tilde{x}, \tilde{y})$ for some $(\tilde{x}, \tilde{y}) \in \mathcal{H}_q$ with $\tilde{x}, \tilde{y} \in \mathsf{P}_{\mathbb{R}}$ and $\tilde{y} \in (0, 1)$. Since $q \geq 2$, we have $1 - q/2 \leq 0$ and $\tilde{y} \in (1 - q/2, 1)$. Hence, Lemma 2.4 gives us a polynomial-time approximate theta shift from $(\tilde{x}, \tilde{y})$ to (x', y'). Since $\tilde{y} \notin 1 - q/2 + i\mathbb{R} = 1 - q/2 + iq\mathbb{R}$ and $\tilde{x}, \tilde{y} \in \mathrm{P}_{\mathbb{R}}$, the transitivity property of polynomial-time approximate shifts, Lemma 2.16, for $(x_1, y_1) = (x, y)$, $(x_2, y_2) = (\tilde{x}, \tilde{y})$ and $(x_3, y_3) = (x', y')$ gives us a polynomial-time approximate series-parallel shift from (x, y) to (x', y').

Now we treat the case $y' = 0$. As a consequence of what we have just shown in the paragraph above, there is a polynomial-time approximate series-parallel shift from (x, y) to $(1-2q, 1/2) \in \mathcal{H}_q$. An n-thickening gives a shift from $(1 - 2q, 1/2)$ to $(x_n, 2^{-n})$, where $x_n = 1 + q/(2^{-n} - 1)$, so there is also a polynomial-time approximate theta shift from $(1 - 2q, 1/2)$ to $(1 - q, 0)$. We conclude that there is a polynomial-time approximate series-parallel shift from (x, y) to $(1 - q, 0)$ by applying Lemma 2.16 with $(x_1, y_1) = (x, y)$, $(x_2, y_2) = (1 - 2q, 1/2)$ and $(x_3, y_3) = (1 - q, 0)$. Note that we can indeed apply Lemma 2.16 because $1 - 2q, 1/2 \in \mathrm{P}_{\mathbb{R}}$ and $1/2 \notin 1 - q/2 + iq\mathbb{R}$.

Now we deal with the case $y' > 1$. We use again the polynomial-time approximate series-parallel shift from (x, y) to $(x_1, y_1) = (1 - 2q, 1/2) \in \mathcal{H}_q$. We use a 2-stretch to $(1 - 2q, 1/2)$-implement (x_2, y_2) with $x_2 = (1 - 2q)^2 \geq 9$ and $y_2 = 1 + q/(x_2 - 1) > 1$. Hence, there is a polynomial-time approximate series-parallel shift from (x, y) to (x_2, y_2). Since x_2 and y_2 are real algebraic numbers with $y_2 > 1$ and $(x_2 - 1)(y_2 - 1) = q > 0$, in view of Lemma 2.7, we have a polynomial-time approximate theta shift from (x_2, y_2) to (x', y'). Note that $y_2 \notin \{1\} \cup (1 - q/2 + iq\mathbb{R})$. Hence, we can apply the transitivity property shown in Lemma 2.16 with $(x_1, y_1) = (x, y)$, $(x_2, y_2) = (x_2, y_2)$ and $(x_3, y_3) = (x', y')$ and find a polynomial-time approximate series-parallel shift from (x, y) to (x', y').

Finally, we study the case $y' < 0$. In light of Corollary 2.28, there is a polynomial-time approximate series-parallel shift from (x, y) to $(0, 1 - q)$. Note that $1 - q \leq -1$. In this proof we have already shown that there is a polynomial-time approximate series-parallel shift from (x, y) to $(x_3, y_3) \in \mathcal{H}_q$ for $y_3 = y'/(1 - q) > 0$. Since $y' = y_3(1 - q)$, by Lemma 2.11 with parameters $(x_1, y_1) = (x, y)$, $(x_2, y_2) = (0, 1 - q)$ and $(x_3, y_3) = (x_3, y_3)$, we conclude that there is a polynomial-time approximate series-parallel shift from (x, y) to (x', y'). □

2.5 Hardness results

We begin with obtaining lower bounds on $Z_{\text{Tutte}}(G; q, \gamma)$ for algebraic numbers q and γ. In Section 2.5.2, we review the algorithm of [78] for computing algebraic representations, and in Section 2.5.3 the exact #P-hardness results that we will use. The rest of the section gives various

ingredients that are needed in the reduction, which are put together in Section 2.5.8 where we prove all of our main theorems of this chapter.

2.5.1 Properties of $Z_{\text{Tutte}}(G;q,\gamma)$ for algebraic numbers q and γ

In this section we give a lower bound on $Z_{\text{Tutte}}(G;q,\gamma)$ and study the degree and height of $Z_{\text{Tutte}}(G;q,\gamma)$ when q and γ are algebraic numbers. First, we have to introduce some concepts and results from algebraic number theory. The *degree* of an algebraic number γ is the degree of its minimal polynomial p, and we denote it by $d(\gamma)$. Recall that the *degree of a field extension* F/K is the dimension of F as a K-vector space, and it is denoted by $[F:K]$. It is well-known that if γ is algebraic, then $[K(\gamma):K]$ is the degree of the minimal polynomial of γ over K [112, Chapter 5]. In particular, we have $[\mathbb{Q}(\gamma):\mathbb{Q}] = d(\gamma)$. The *usual height* of a polynomial $f \in \mathbb{Z}[x_1,\ldots,x_m]$ is the the largest value among the absolute values of its coefficients and it is denoted by $H(f)$. The *usual height* of γ is $H(\gamma) = H(p)$. One can find several (non-equivalent) definitions of the height of an algebraic number in the literature. Another one of these definitions is the absolute logarithmic height. First, we have to introduce the *Mahler's measure* of a polynomial $f \in \mathbb{Z}[x]$, which is given by

$$M(f) = |a_d| \prod_{i=1}^{d} \max\{1, |\alpha_i|\},$$

where $f(x) = \sum_{j=0}^{d} a_j x^j$, $a_d \neq 0$, and $\alpha_1,\ldots,\alpha_d$ are the roots of f. It is well-known that

$$2^{-d(f)} H(f) \leq M(f) \leq H(f)\sqrt{d(f)+1}, \tag{2.20}$$

see [118, Lemma 3.11]. The *Mahler's measure* of an algebraic number γ with minimal polynomial p is $M(\gamma) = M(p)$. The *absolute logarithmic height* of γ is $h(\gamma) = d(\gamma)^{-1} \log M(\gamma)$. Note that $h(\gamma) \geq 0$ because $M(\gamma) \geq 1$. Now we can state a lower bound for the evaluation of a polynomial at algebraic numbers.

Lemma 2.33 ([118, Section 3.5.4])**.** *Let $f \in \mathbb{Z}[x_1,\ldots,x_m]$ be a polynomial in m variables and let $\gamma_1,\ldots,\gamma_m$ be algebraic numbers. If $f(\gamma_1,\ldots,\gamma_m) \neq 0$, then we have*

$$|f(\gamma_1,\ldots,\gamma_m)| \geq e^{-cT},$$

where $T = \deg f + \log H(f)$, $c = D(2 + h(\gamma_1) + \cdots + h(\gamma_m))$ and $D = [\mathbb{Q}(\gamma_1,\ldots,\gamma_m):\mathbb{Q}]$.

Corollary 2.34. *Let q and γ be algebraic numbers. We can compute a rational number $C_{q,\gamma}$ with $C_{q,\gamma} > 1$ such that, for any graph G, either $Z_{Tutte}(G;q,\gamma) = 0$ or $|Z_{Tutte}(G;q,\gamma)| \geq C_{q,\gamma}^{-\text{size}(G)}$.*

Proof. Recall that we represent an algebraic number γ as its minimal polynomial p and a rectangle of the complex plane where γ is the only root of p. Let $G = (V,E)$ be a graph. Let $n = |V|$ and $m = |E|$. Let us assume that $Z_{\text{Tutte}}(G;q,\gamma) \neq 0$. We can apply Lemma 2.33 for $f(q,\gamma) = Z_{\text{Tutte}}(G;q,\gamma)$ to find that $|Z_{\text{Tutte}}(G;q,\gamma)| \geq e^{-cT}$, where c and T are as in Lemma 2.33. We have $c = D(2 + h(q) + h(\gamma))$ and $D = [\mathbb{Q}(q,\gamma):\mathbb{Q}]$, so $c \geq 2$. Note

that, by definition of Z_{Tutte}, we have $H(f) \leq 2^m$ and $\deg f \leq n+m$. Hence, we find that $|Z_{\text{Tutte}}(G;q,\gamma)| \geq e^{-2c\,\text{size}(G)}$. It remains to compute a rational number $C_{q,\gamma}$ in (e^{2c},∞) to conclude the result. From $D = [\mathbb{Q}(q,\gamma):\mathbb{Q}]$, we can compute D exactly. Moreover, we can apply (2.20) to upper bound $h(q)$ and $h(\gamma)$ in terms of the usual heights and degrees of q and γ, and compute an appropriate rational number $C_{q,\gamma}$ with the help of these upper bounds. □

The case $q = 2$ (Ising model) of Corollary 2.34 has previously been shown in [55, Lemma 6.4]. Note that the approach followed in this section can be applied to obtain lower bounds for other partition functions.

In the rest of this section we upper bound the degree and the usual height of the algebraic number $Z_{\text{Tutte}}(G;q,\gamma)$ in terms of the usual heights and degrees of q and γ. We will make use of these bounds in the proof of Lemma 2.41.

Let q and γ be two algebraic numbers. By the tower law, we have $[\mathbb{Q}(q,\gamma):\mathbb{Q}] = [\mathbb{Q}(q,\gamma):\mathbb{Q}(q)][\mathbb{Q}(q):\mathbb{Q}] \leq d(q)d(\gamma)$, where we used that the degree of the minimal polynomial of γ over $\mathbb{Q}(q)$ is bounded by $d(\gamma)$. Since $Z_{\text{Tutte}}(G;q,\gamma)$ is in $\mathbb{Q}(q,\gamma)$, it follows that its degree is bounded by $d(q)d(\gamma)$.

Now we argue how we can bound the usual height of $Z_{\text{Tutte}}(G;q,\gamma)$. A well-known property of the absolute logarithmic height is that $h(\alpha\beta) \leq h(\alpha)+h(\beta)$, $h(\alpha+\beta) \leq \log 2 + h(\alpha) + h(\beta)$ and $h(1/\alpha) = h(\alpha)$ [118, Property 3.3]. Moreover, if n is an integer, then $h(n) = \log|n|$. A more general property is the following one.

Lemma 2.35 ([118, Lemma 3.7])**.** *Let $f \in \mathbb{Z}[x_1,\ldots,x_t]$ be a non-zero polynomial in t variables with integer coefficients. Let $\gamma_1,\ldots,\gamma_t$ be algebraic numbers. Then*

$$h\left(f\left(\gamma_1,\ldots,\gamma_t\right)\right) \leq \log L(f) + \sum_{j=1}^{t} \deg_{x_j}(f) h(\gamma_j),$$

where $L(f)$ is the sum of the absolute values of the coefficients of f and $\deg_{x_j}(f)$ is the degree of f with respect to the j-th variable.

Corollary 2.36. *Let q and γ be algebraic numbers. Then, for any graph $G = (V,E)$ with $n = |V|$ and $m = |E|$, we have*

$$d\left(\frac{Z_{s|t}(G;q,\gamma)}{Z_{st}(G;q,\gamma)}\right) \leq d\left(q\right) d\left(\gamma\right) \quad \text{and} \quad H\left(\frac{Z_{s|t}\left(G;q,\gamma\right)}{Z_{st}\left(G;q,\gamma\right)}\right) \leq \left(2^{m+1/2} e^{nh(q)+mh(\gamma)}\right)^{2d(q)d(\gamma)}.$$

Proof. The degree bound on $Z_{s|t}(G;q,\gamma)/Z_{st}(G;q,\gamma)$ follows from the fact that that it is in $\mathbb{Q}(q,\gamma)$. For its absolute logarithmic height, we have

$$h\left(\frac{Z_{s|t}(G;q,\gamma)}{Z_{st}(G;q,\gamma)}\right) \leq h\left(Z_{st}(G;q,\gamma)\right) + h\left(Z_{s|t}(G;q,\gamma)\right).$$

Note that $L(Z_{st}(G;q,\gamma)) + L(Z_{s|t}(G;q,\gamma)) = 2^m$. As a consequence of Lemma 2.35, we find that

$$h\left(Z_{st}(G;q,\gamma)\right) + h\left(Z_{s|t}(G;q,\gamma)\right) \leq 2\left(m\log 2 + nh(q) + mh(\gamma)\right).$$

Recall that $M(\alpha) = \exp(d(\alpha)h(\alpha))$. Thus, the bounds on the Mahler's measure (2.20) yield the inequality $H(\alpha) \leq (2\exp(h(\alpha)))^{d(\alpha)}$. We conclude that

$$H\left(\frac{Z_{s|t}(G;q,\gamma)}{Z_{st}(G;q,\gamma)}\right) \leq \left(2e^{2(m\log 2+nh(q)+mh(\gamma))}\right)^{d(q)d(\gamma)} = \left(2^{m+1/2}e^{nh(q)+mh(\gamma)}\right)^{2d(q)d(\gamma)}. \quad \square$$

One could derive analogous bounds to those of Corollary 2.36 for the algebraic number $Z_{\text{Tutte}}(G;q,\gamma)$ by applying the same argument.

2.5.2 Computing representations of algebraic numbers via approximations

Kannan, Lenstra and Lovász [78] showed how to reconstruct the minimal polynomial of an algebraic number from a certain number of digits of its binary expansion, and we will use their algorithm as a black-box in our reduction of Section 2.5.5, in the following form.

Lemma 2.37 ([78, Theorem 1.19])**.** *Let α be an algebraic number and let d and U be upper bounds on the degree and usual height, respectively, of α. Suppose that we are given a rational approximation $\overline{\alpha}$ to α such that $|\alpha - \overline{\alpha}| \leq 2^{-b}/(12d)$, where b is the smallest positive integer such that*

$$2^b \geq 2^{d^2/2}(d+1)^{(3d+4)/2}U^{2d}.$$

Then the minimal polynomial of α can be determined in $O(d^5(d+\log U))$ arithmetic operations on integers having $O(d^2(d+\log U))$ binary bits.

The algorithm in Lemma 2.37 is based on the Lenstra–Lenstra–Lovász lattice basis reduction algorithm, we refer the reader to [123] for more details.

2.5.3 Exact Hardness results

We will use the following hardness results from [73] regarding the problem of exactly evaluating $Z_{\text{Tutte}}(G;q,\gamma)$, given a graph G. We refer to this problem as $\textsc{Tutte}(q,\gamma)$. Jaeger et al. [73] identify the following 9 "special" points of the Tutte plane: $(1,-1)$, $(0,0)$, $(4,-2)$, $(2,-2)$, $(2,-1)$, $(2,-i-1)$, $(2,i-1)$, $(3,\omega_3^2-1)$, and $(3,\omega_3-1)$, where $i=\sqrt{-1}$ and $\omega_3 = \exp(2\pi i/3)$.[2] With these special points in mind, their main result on the complexity of $\textsc{Tutte}(q,\gamma)$ can be stated as follows.

Theorem 2.38 ([73, Proposition 1])**.** *Let q and γ be algebraic numbers. Then $\textsc{Tutte}(q,\gamma)$ is $\#\mathsf{P}$-hard unless $q=1$ or (q,γ) is a special point, in which case $\textsc{Tutte}(q,\gamma)$ is in FP.*

In [116], Vertigan studied the complexity of the problem $\textsc{PlanarTutte}(q,\gamma)$, which also turns out to be hard for most parameters q and γ.

Theorem 2.39 ([116, Theorem 5.1])**.** *Let q and γ be algebraic numbers. Then $\textsc{PlanarTutte}(q,\gamma)$ is $\#\mathsf{P}$-hard unless $q \in \{1,2\}$ or (q,γ) is a special point, in which case $\textsc{PlanarTutte}(q,\gamma)$ is in FP.*

[2] In the (x,y)-parametrisation, the special points are $(0,0)$, $(1,1)$, $(-1,-1)$, $(0,-1)$, $(-1,0)$, $(i,-i)$, $(-i,i)$, (ω_3,ω_3^2), and (ω_3^2,ω_3).

2.5.4 Computational problems

In this section, we define a few computational problems that will be useful in our reductions; these were also considered in [55]. Let q be a real algebraic number, $\gamma_1, \ldots, \gamma_k$ be algebraic numbers, and K, ρ be real numbers with $K > 1$, $\rho > 0$.

Name: Sign-Tutte$(q, \gamma_1, \ldots, \gamma_k)$ – here $\gamma_1, \ldots, \gamma_k$ are real.
Instance: A (multi)graph G and a weight function $\hat{\gamma}\colon E \to \{\gamma_1, \ldots, \gamma_k\}$.
Output: A correct statement of the form $Z_{\text{Tutte}}(G; q, \hat{\gamma}) \geq 0$ or $Z_{\text{Tutte}}(G; q, \hat{\gamma}) \leq 0$.

Name: Factor-K-NormTutte$(q, \gamma_1, \ldots, \gamma_k)$.
Instance: A (multi)graph G and a weight function $\hat{\gamma}\colon E \to \{\gamma_1, \ldots, \gamma_k\}$.
Output: If $Z_{\text{Tutte}}(G; q, \hat{\gamma}) = 0$, the algorithm may output any rational number. Otherwise, it must output $\hat{N} \in \mathbb{Q}$ such that $\hat{N}/K \leq |Z_{\text{Tutte}}(G; q, \hat{\gamma})| \leq K\hat{N}$.

Name: Distance-ρ-ArgTutte(q, γ).
Instance: A (multi)graph G.
Output: If $Z_{\text{Tutte}}(G; q, \gamma) = 0$, the algorithm may output any rational number. Otherwise, it must output $\hat{A} \in \mathbb{Q}$ such that, for some $a \in \arg(Z_{\text{Tutte}}(G; q, \gamma))$, we have $|\hat{A} - a| \leq \rho$.

We also consider these problems for the Potts model (with parameters q and $y = \gamma + 1$), and we write Potts instead of Tutte in the name of these problems when we refer to the Potts ones. We also consider all these problems restricted to planar graphs, in which case we write PlanarTutte instead of Tutte in the name of the problem. It is a trivial observation that the planar case reduces to the general case.

2.5.5 Reducing exact computation to sign and approximate computation

In this section, we first review the binary search technique of [55], which we will refer to as "interval-shrinking". Then, we use this to obtain several of our inapproximability theorems.

Let $f(\varepsilon) = -\varepsilon A + B$ be a linear function, where A and B are real algebraic numbers with $A \neq 0$. Let $\varepsilon^* = B/A$ be the zero of f. Let $(\varepsilon', \varepsilon'')$ be an open interval with length $l > 0$ such that ε^* is in $(\varepsilon', \varepsilon'')$ or, equivalently, $f(\varepsilon')f(\varepsilon'') < 0$. We want to find a small open subinterval of $(\varepsilon', \varepsilon'')$ that contains ε^*.

First, assume that we have an oracle that, on input ε, outputs the sign of $f(\varepsilon)$, unless when $f(\varepsilon) = 0$, in which case the output of the oracle is unreliable. Let $\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_4$ be a partition of the interval $(\varepsilon', \varepsilon'')$ such that $\varepsilon_0 = \varepsilon'$, $\varepsilon_4 = \varepsilon''$ and $\varepsilon_{i+1} - \varepsilon_i \geq l/10$ for every $i \in \{0, \ldots, 3\}$. We invoke the oracle with input ε_i to determine the sign of $f(\varepsilon_i)$ for every $i \in \{0, \ldots, 4\}$; let s_i be the answer of the oracle. Then, we have a monotone sequence $s_0, \ldots, s_4$ of positive and negative signs with $s_0 \neq s_4$. Hence, there are two possibilities: either $s_0 = s_1 = s_2$, in which case $\varepsilon_1 < \varepsilon^*$ and we can recurse on $(\varepsilon_1, \varepsilon_4)$, or $s_2 = s_3 = s_4$, in which case $\varepsilon^* < \varepsilon_3$ and we can recurse on $(\varepsilon_0, \varepsilon_3)$. In any of these two cases, we can shrink the interval $(\varepsilon', \varepsilon'')$ to at most $9/10$

of its original length. Then, recursively, we can find an open subinterval of arbitrarily small length containing the zero of f.

Next, assume that we have an oracle that returns a multiplicative approximation to the norm of f. More accurately, let $\eta = 1/41$ and suppose that we have an oracle that, on input ε, returns a value $\hat{f}(\varepsilon)$ satisfying

$$(1-\eta)\,|f(\varepsilon)| < \frac{1}{1+\eta}\,|f(\varepsilon)| \leq \hat{f}(\varepsilon) \leq (1+\eta)\,|f(\varepsilon)|$$

when $f(\varepsilon) \neq 0$ (otherwise the value $\hat{f}(\varepsilon)$ is unreliable). The approach given in [55] by Goldberg and Guo to shrink $(\varepsilon', \varepsilon'')$ is as follows. First, let us assume that $A > 0$, so f is strictly decreasing. Let $\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{10}$ be a partition of the interval $(\varepsilon', \varepsilon'')$ such that $\varepsilon_0 = \varepsilon'$, $\varepsilon_{10} = \varepsilon''$ and $\varepsilon_{i+1} - \varepsilon_i \geq l/20$ for every $i \in \{0, \ldots, 9\}$. These numbers are not chosen to be optimal but they suffice. We invoke our oracle to compute $\hat{f}(\varepsilon_i)$ for $i \in \{0, \ldots, 10\}$. Let s_i be the sign (positive, negative, or zero) of $\hat{f}(\varepsilon_i) - \hat{f}(\varepsilon_{i+1})$ for each $i \in \{0, \ldots, 9\}$. We analyse the signs s_i for $i \in \{0, \ldots, 9\}$. First, we consider the case $\varepsilon_i < \varepsilon_{i+1} < \varepsilon^*$. Note that we have $f(\varepsilon_i) > f(\varepsilon_{i+1}) > 0$. Moreover,

$$\begin{aligned}\hat{f}\,(\varepsilon_i) - \hat{f}\,(\varepsilon_{i+1}) &\geq (1-\eta)\,f\,(\varepsilon_i) - (1+\eta)\,f\,(\varepsilon_{i+1}) \\ &= A\,(\varepsilon_{i+1} - \varepsilon_i - \eta\,(2\varepsilon^* - \varepsilon_i - \varepsilon_{i+1}))\,.\end{aligned}$$

Note that $\varepsilon^* - \varepsilon_i$ and $\varepsilon^* - \varepsilon_{i+1}$ are both at most l and, thus, we obtain $2\varepsilon^* - \varepsilon_i - \varepsilon_{i+1} \leq 2l$. So since $\eta = 1/41$ and $\varepsilon_{i+1} - \varepsilon_i \geq l/20$, we conclude that s_i is positive. Now we consider the case $\varepsilon^* < \varepsilon_i < \varepsilon_{i+1}$. This time we have $f(\varepsilon_{i+1}) < f(\varepsilon_i) < 0$,

$$\begin{aligned}\hat{f}\,(\varepsilon_i) - \hat{f}\,(\varepsilon_{i+1}) &\leq (1+\eta)\,(-f\,(\varepsilon_i)) - (1-\eta)\,(-f\,(\varepsilon_{i+1})) \\ &= -A\,(\varepsilon_{i+1} - \varepsilon_i - \eta\,(\varepsilon_i + \varepsilon_{i+1} - 2\varepsilon^*))\,,\end{aligned}$$

and $0 < \varepsilon_i + \varepsilon_{i+1} - 2\varepsilon^* < 2l$. We conclude that s_i is negative. If $\varepsilon_i \leq \varepsilon^*$ and $\varepsilon^* \leq \varepsilon_{i+1}$, then we do not know what the value of s_i will be. However, this is true for at most two consecutive values of i. With these properties of the signs s_i in mind, let us study the sequence $s_0, \ldots, s_9$. There are two possibilities. The first one is that s_0, s_1, s_2, s_3 are all positive, in which case $\varepsilon_2 < \varepsilon^*$ and we can recurse on $(\varepsilon_2, \varepsilon_{10})$. The second possibility is that s_6, s_7, s_8, s_9 are all negative, in which case $\varepsilon^* < \varepsilon_8$ and we can recurse on $(\varepsilon_0, \varepsilon_8)$. In any of these two cases, we can shrink the interval $(\varepsilon', \varepsilon'')$ to at most $9/10$ of its original length. Again using binary search it is possible to find a small open subinterval containing the zero of f. Let us now assume that $A < 0$. In this case, one can analogously prove that the sign s_i is positive when $\varepsilon_i < \varepsilon_{i+1} < \varepsilon^*$ and negative when $\varepsilon^* < \varepsilon_i < \varepsilon_{i+1}$, so the same procedure allows us to shrink $(\varepsilon', \varepsilon'')$.

Let q and γ be real algebraic numbers with $q \notin \{0, 1\}$ and $\gamma > 0$. Let H be a graph and let s and t be two distinct connected vertices of H. We are going to apply these interval stretching techniques to the linear function

$$f(\varepsilon; H, \gamma) = Z_{s|t}(H; q, \gamma)\left(1 - \frac{1}{q}\right) + \varepsilon\left(Z_{st}(H; q, \gamma) + \frac{1}{q} Z_{s|t}(H; q, \gamma)\right). \tag{2.21}$$

Let us write this function as $f(\varepsilon; H, \gamma) = B(H, \gamma) - \varepsilon A(H, \gamma)$, where $B(H, \gamma) = Z_{s|t}(H; q, \gamma)(1 - 1/q)$ and $A(H, \gamma) = -Z_{st}(H; q, \gamma) - q^{-1} Z_{s|t}(H; q, \gamma)$. We have

$$\begin{aligned} f(0; H, \gamma) &= Z_{s|t}(H; q, \gamma) \left(1 - \frac{1}{q}\right); \\ f(1 - q; H, \gamma) &= (1 - q) Z_{st}(H; q, \gamma). \end{aligned} \tag{2.22}$$

Under certain hypotheses, we are going to prove that $f(0; H, \gamma) f(1-q; H, \gamma) < 0$, so $A(H, \gamma) \neq 0$ and $f(-; H, \gamma)$ has a zero between 0 and $1 - q$. This allows us to find a suitable interval where we can perform interval-shrinking. For this purpose we will also need Lemma 2.40, that tells us that the zero of $f(-; H, \gamma)$ is not close to either 0 or $1 - q$.

Lemma 2.40. *Let q and γ be real algebraic numbers with $q \notin \{0, 1\}$ and $\gamma > 0$. Let $H = (V, E)$ be a graph and let s and t be two distinct connected vertices of H. Let $n = |V|$, $m = |E|$, $r = \max\{n, m\}$ and $c = 2 \max\{|q|, 1/|q|\} \max\{\gamma, 1/\gamma\}$. Let ε^* be the zero of the function $f(\varepsilon; H, \gamma) = B(H, \gamma) - A(H, \gamma)$, defined as in (2.21). Let us assume that $|Z_{st}(H; q, \gamma)| \geq c^{-r}$, $|Z_{s|t}(H; q, \gamma)| \geq c^{-r}$ and $A(H, \gamma) \neq 0$. Then we have $|1 - q - \varepsilon^*| \geq |1 - q|\, c^{-2r}$ and $|\varepsilon^*| \geq |1 - 1/q|\, c^{-2r}$.*

Proof. In view of the definition of $f(\varepsilon; H, \gamma)$ and equation (2.22), we have

$$|1 - q - \varepsilon^*| = \frac{|f(\varepsilon^*; H, \gamma) - f(1 - q)|}{|A(H, \gamma)|} = \frac{|1 - q|\, |Z_{st}(H; q, \gamma)|}{|A(H, \gamma)|}.$$

Note that

$$|A(H, \gamma)| \leq \sum_{A \subseteq E} \max\{|q|, 1/|q|\}\, |q|^{k(A)-1}\, |\gamma|^{|A|} \leq c^r. \tag{2.23}$$

Moreover, we have $|Z_{st}(H; q, \gamma)| \geq c^{-r}$ by hypothesis, so we conclude that $|1-q-\varepsilon^*| \geq |1-q|c^{-2r}$. Analogously, we find that

$$|\varepsilon^*| = \frac{|f(\varepsilon^*; H, \gamma) - f(0)|}{|A(H, \gamma)|} = \frac{|1 - 1/q|\, |Z_{s|t}(H; q, \gamma)|}{|A(H, \gamma)|} \geq \left|1 - \frac{1}{q}\right| c^{-2r}. \qquad \square$$

Lemma 2.41. *Let K be a real number with $K > 1$. Let q, γ_1 and γ_2 be real algebraic numbers such that $q > 1$, $\gamma_1 \in (-2, -1)$ and $\gamma_2 > 0$. Let us assume that we have access to an oracle for* Factor-K-NormPlanarTutte(q, γ_1, γ_2). *Then there exists an algorithm that takes as input a positive integer ρ and a planar graph H along with two distinct connected vertices s and t of H, and, for $\gamma = (\gamma_2 + 1)^\rho - 1$, this algorithm computes a representation of the algebraic number $Z_{s|t}(H; q, \gamma)/Z_{st}(H; q, \gamma)$ in polynomial time in ρ and the size of H. Moreover, if we have access to the more powerful oracle* Factor-K-NormTutte(q, γ_1, γ_2), *then we can remove the constraint that H is planar.*

Proof. Since Factor-K-NormPlanarTutte(q, z) is equally hard for any $K > 1$ (see Section 1.3), we may assume that $K = 1 + \eta$ for $\eta = 1/41$.

Let ρ, $H = (V, E)$ and s, t be the inputs of our algorithm. Let $n = |V|$ and $m = |E|$. Let $c = 2 \max\{|q|, 1/|q|\} \max\{\gamma, 1/\gamma\}$, so $c \geq 2$, and let $r = \max\{n, m\}$. Let H' be a copy of H with

an extra edge from s to t. Let γ' be a weight that we can implement and let $\varepsilon = \gamma' + 1$, so the pair $(1 + q/(\varepsilon - 1), \varepsilon)$ is (q, γ') written in (x, y) coordinates. We will choose γ' and argue how we can implement γ' later in the proof. When we say we implement ε, we mean that we implement the pair $(1 + q/(\varepsilon - 1), \varepsilon)$ in (x, y) notation or, equivalently, γ'.

Let τ be the weight function on H' that assigns the weight γ to the edges of H and the weight γ' to the new edge. Then, as was observed in [59, Lemma 2], we have

$$\begin{aligned} Z_{\text{Tutte}}(H'; q, \tau) &= Z_{st}(H; q, \gamma)(1 + \gamma') + Z_{s|t}(H; q, \gamma)\left(1 + \frac{\gamma'}{q}\right) \\ &= Z_{s|t}(H; q, \gamma)\left(1 - \frac{1}{q}\right) + \varepsilon\left(Z_{st}(H; q, \gamma) + \frac{1}{q} Z_{s|t}(H; q, \gamma)\right) \\ &= f(\varepsilon; H, \gamma), \end{aligned} \tag{2.24}$$

where $f(\varepsilon; H, \gamma)$ was introduced in (2.21). Hence, $Z_{\text{Tutte}}(H'; q, \tau)$ can be seen as a function, with variable ε, of the form $f(\varepsilon; H, \gamma) = B(H, \gamma) - \varepsilon A(H, \gamma)$, where $B(H, \gamma) = Z_{s|t}(H; q, \gamma)(1 - 1/q)$ and $A(H, \gamma) = -Z_{st}(H; q, \gamma) - q^{-1} Z_{s|t}(H; q, \gamma)$. This construction will be used several times in this section. Now we analise $f(-; H, \gamma)$ for our particular setting ($q > 1$). Since q and γ are positive, the quantities $Z_{st}(H; q, \gamma)$ and $Z_{s|t}(H; q, \gamma)$ are positive, so $A(H, \gamma)$ is negative. From $q > 1$ and (2.22), it follows that $f(0; H, \gamma) = B(H, \gamma) > 0$ and $f(1 - q; H, \gamma) < 0$, so $f(0; H, \gamma) f(1 - q; H, \gamma) < 0$ as we wanted. We conclude that the zero ε^* of $f(\varepsilon; H, \gamma)$ is in $(1 - q, 0)$. Note that $\varepsilon \in (1 - q, 0)$ if and only if $\gamma' \in (-q, -1)$. Moreover, we have

$$\begin{aligned} Z_{st}(H; q, \gamma) &\geq q\gamma^m \geq c^{-r}, \\ Z_{s|t}(H; q, \gamma) &\geq q^n \geq c^{-r}. \end{aligned} \tag{2.25}$$

This allow us to apply Lemma 2.40. Once we have all these properties of $f(\varepsilon; H, \gamma)$ at our disposal, we can proceed to describe our algorithm. Our algorithm also works for $q \in (-\infty, 0) \cap (0, 1)$ as long as $f(0; H, \gamma) f(1 - q; H, \gamma) < 0$ and the hypotheses of Lemma 2.40 hold. In the rest of the proof we will only use the fact that $q > 1$ one more time, but this will be made explicit and can easily be adapted to the case $q < 1$ as we will explain in Lemma 2.43.

Our algorithm computes a positive integer j_0 such that $c^{-j_0} \leq |q - 1|/2$. Let j be an integer with $j \geq j_0$. We will first show how to additively approximate $Z_{s|t}(H; q, \gamma)/Z_{st}(H; q, \gamma)$ with error at most $2|q|c^{-j}/|q - 1|$.

If we could efficiently implement the point $(1 - q/(\varepsilon - 1), \varepsilon)$ (in (x, y) coordinates) for any $\varepsilon \in [1 - q, 0]$ using only planar graphs, then our algorithm could perform the interval-shrinking technique explained at the beginning of this section. This would allow us to compute an interval of length at most c^{-j-4r} where the linear function $f(\varepsilon)$ has a zero, which would, in turn, provide us with the desired additive approximation, as we will see later. However, some difficulties arise since we do not know how to implement any specific real algebraic weight. This difficulty was overcome by Goldberg and Jerrum by developing Lemmas 2.7 and 2.8. Here we use the version of these lemmas given in Corollary 2.9. Let $y_1 = \gamma_1 + 1$, $x_1 = 1 + q/(y_1 - 1)$, $y_2 = \gamma_2 + 1$ and $x_2 = 1 + q/(y_2 - 1)$. Note that $y_1 \in (-1, 0)$, $y_2 > 1$ and $q \neq 0$. Hence, Corollary 2.9 allows

us to efficiently implement approximations of real algebraic numbers when applied with the parameters x_1, y_1, x_2, y_2. Every time we invoke Corollary 2.9 we will be using these parameters. We are going to use this corollary to implement approximations of $\varepsilon \in (1-q, 0)$. This is the only point where our algorithm uses the fact that $\gamma_1 \in (-2, -1)$ or, equivalently, $y_1 \in (-1, 0)$. In further lemmas where we study the case $q < 1$, we will have to implement approximations of $\varepsilon \in (0, 1-q)$ and, hence, we will get away with the weaker hypothesis $\gamma_1 \in (-1, 0)$, or, equivalently, $y_1 \in (0, 1)$. (This hypothesis is "weaker" in the sense that a 2-thickening of a $y_1 \in (-1, 0)$ implements a $y_1 \in (0, 1)$.)

We want to implement numbers ε' and ε'' so that $\varepsilon^* \in (\varepsilon', \varepsilon'') \subseteq (1-q, 0)$. Note that here we are using that $q > 1$. When $q < 1$ our algorithm would work on the interval $(0, 1-q)$ instead of $(1-q, 0)$. This paragraph is the last time that we use the hypothesis $q > 1$ in this proof. The argument given in this paragraph will be revisited when we deal with the case $q < 1$ in further lemmas. Our algorithm first applies the algorithm given in Corollary 2.9 with $y' = -(1-1/q)c^{-2r}/2$, k such that $|y_1|^k < |y'| < |y_1|^{-k}$ and $n = \lceil 2r \log_2(c) - \log_2(1-1/q) + 2 \rceil$. Note that $k = O(r)$ and $n = O(r)$. This procedure computes a theta graph and a weight function taking weights in $\{\gamma_1, \gamma_2\}$ that implement a point $(1 + q/(\varepsilon'' - 1), \varepsilon'')$ such that $|y' - \varepsilon''| \leq 2^{-n} \leq (1-1/q)c^{-2r}/4$ in polynomial time in $r = O(\text{size}(H))$. We have $-3(1-1/q)c^{-2r}/4 \leq \varepsilon'' \leq -(1-1/q)c^{-2r}/4$, so, by Lemma 2.40, we find that $\varepsilon^* < \varepsilon'' < 0$. Now our algorithm invokes again Corollary 2.9, this time with inputs $y' = 1 - q + (q-1)c^{-2r}/2$, k such that $|y_1|^k < |y'| < |y_1|^{-k}$ and $n = \lceil 2r \log_2(c) - \min\{0, \log_2(q-1)\} + 2 \rceil$. This implements $(1 + q/(\varepsilon' - 1), \varepsilon')$ with $|y' - \varepsilon'| \leq (q-1)c^{-2r}/4$, which gives $1 - q + (q-1)c^{-2r}/4 \leq \varepsilon' \leq 1 - q + 3(q-1)c^{-2r}/4$. Again by Lemma 2.40, we find that $1 - q < \varepsilon' < \varepsilon^*$. The interval $(\varepsilon', \varepsilon'')$ is the starting interval for the interval-shrinking procedure.

Let us assume that we are carrying out the interval-shrinking technique explained at the beginning of this section, so we have an interval $(\varepsilon', \varepsilon'')$ of length l where f changes sign. Let us also assume that we can implement the endpoints ε' and ε''. We want to find a subinterval of length at most $9l/10$ where f changes sign. We can assume that $l > c^{-j-4r}$, since otherwise we do not need to shrink the interval further. Let $p = 10$ be the number of subintervals into which $(\varepsilon', \varepsilon'')$ is partitioned by the interval-shrinking technique. We want to find numbers $\varepsilon_1, \ldots, \varepsilon_{p-1}$ such that we can implement the point $(1 + q/(\varepsilon_i - 1), \varepsilon_i)$ for every $i \in \{1, \ldots, p-1\}$ and, for $\varepsilon_0 = \varepsilon'$ and $\varepsilon_p = \varepsilon''$, we have $\varepsilon_i - \varepsilon_{i-1} \geq l/2p$ for every $i \in \{1, \ldots, p\}$, which is what is required to perform interval-shrinking. For each $i \in \{1, \ldots, p-1\}$, our algorithm computes $\varepsilon_i' = \varepsilon' + il/p$ and then it applies the algorithm given in Corollary 2.9 with $y' = \varepsilon_i'$, k such that $|y_1|^k < |y'| < |y_1|^{-k}$ and $n = \lceil (j + 4r) \log_2(c) + \log_2(4p) \rceil$. This procedure computes a graph and a weight function taking weights in $\{\gamma_1, \gamma_2\}$ that implement a point $(1 + q/(\varepsilon_i - 1), \varepsilon_i)$ such that $|\varepsilon_i' - \varepsilon_i| \leq 2^{-n} \leq c^{-j-4r}/(4p)$. This application of the procedure given in Corollary 2.9 takes polynomial time in j, r and k. Note that k is polynomial in r and j because $|1 - q| \geq |\varepsilon_i'| \geq l/p \geq c^{-j-4r}/p$ for any $i \in \{1, \ldots, p-1\}$. The algebraic numbers $\varepsilon', \varepsilon_1, \ldots, \varepsilon_{p-1}, \varepsilon''$ form a partition the interval $(\varepsilon', \varepsilon'')$. Our algorithm has computed theta (and, thus, planar) graphs that implement $(1 + q/(\varepsilon_i - 1), \varepsilon_i)$, so it can use

the oracle Factor-K-NormPlanarTutte(q, γ_1, γ_2) to multiplicatively approximate $f(\varepsilon_i)$ for every $i \in \{0, \dots, p\}$. Note that

$$\varepsilon_i - \varepsilon_{i-1} \geq \varepsilon'_i - \varepsilon'_{i-1} - c^{-j-4r}\frac{1}{2p} \geq \frac{l}{2p}$$

for every $i \in \{1, \dots, p\}$. Therefore, our algorithm can apply the interval-shrinking technique discussed at the beginning of this section to shrink $(\varepsilon', \varepsilon'')$.

To guarantee that this interval-shrinking technique computes an interval of length at most c^{-j-4r}, it suffices to subdivide the original interval $\lceil (j+4r)\log_{10/9}(c) + \log_{10/9}|1-q| \rceil$ times due to the fact that each iteration shrinks the interval to $9/10$ of its size. In [59] and [55] the authors used the information provided by this interval-shrinking procedure to solve the problem #Minimum Cardinality (s,t)-Cut for arbitrary graphs (not-necessarily planar). Here we follow a different approach that allows us to compute the representation of $Z_{s|t}(H;q,\gamma)/Z_{st}(H;q,\gamma)$.

Once our algorithm has computed an interval of length at most c^{-j-4r} where f has a zero, it implements a point $(1+q/(\hat{\varepsilon}-1), \hat{\varepsilon})$ such that $\hat{\varepsilon}$ is in this interval. This can be done by applying Corollary 2.9 with the same parameters as before other than y', which is set as the middle point of the computed interval. Let ε^* be the zero of f. Note that $|\hat{\varepsilon} - \varepsilon^*| \leq c^{-j-4r}$. Recall that $f(\varepsilon; H, \gamma) = B(H,\gamma) - A(H,\gamma)\varepsilon$. For a graph H' and a weight function τ as in (2.24), with $\gamma' = \hat{\varepsilon} - 1$ (which we can now implement as promised before (2.24)), we obtain

$$\left|Z_{Tutte}\left(H';q,\tau\right)\right| = |f(\hat{\varepsilon})| = |f(\hat{\varepsilon}) - f(\varepsilon^*)| \leq |A(H,\gamma)|c^{-j-4r} \leq c^{-j-3r}, \tag{2.26}$$

where we used the elementary bound $|A(H,\gamma)| \leq c^r$, which has been established in (2.23). By dividing by $Z_{st}(H;q,\gamma)$ in (2.24), which is non-zero, and rearranging the terms we find that

$$\frac{Z_{\text{Tutte}}(H';q,\tau)}{Z_{st}(H;q,\gamma)} = \hat{\varepsilon} + \left(1 + \frac{\hat{\varepsilon}-1}{q}\right)\frac{Z_{s|t}(H;q,\gamma)}{Z_{st}(H;q,\gamma)}.$$

Dividing by $1 + (\hat{\varepsilon}-1)/q = (q-1+\hat{\varepsilon})/q$ yields

$$\frac{qZ_{\text{Tutte}}(H';q,\tau)}{(q-1+\hat{\varepsilon})Z_{st}(H;q,\gamma)} = -\frac{\hat{\varepsilon}q}{1-q-\hat{\varepsilon}} + \frac{Z_{s|t}(H;q,\gamma)}{Z_{st}(H;q,\gamma)}. \tag{2.27}$$

We claim that $|1-q-\hat{\varepsilon}| \geq |1-q|c^{-2r}/2$. Recall that in view of Lemma 2.40, we have $|1-q-\varepsilon^*| \geq |1-q|c^{-2r}$. Hence, we obtain

$$|1-q-\hat{\varepsilon}| \geq |1-q-\varepsilon^*| - |\varepsilon^* - \hat{\varepsilon}| \geq |1-q|\,c^{-2r} - c^{-j-4r} \geq \frac{|1-q|}{2}c^{-2r},$$

where we used that $c^{-j-4r} \leq c^{-j_0}c^{-4r} \leq |q-1|\,c^{-4r}/2$ by definition of j_0. Therefore, we can apply this lower bound in conjunction with (2.25), (2.26) and (2.27) to conclude that

$$\left|\frac{Z_{s|t}(H;q,\gamma)}{Z_{st}(H;q,\gamma)} - \frac{\hat{\varepsilon}q}{1-q-\hat{\varepsilon}}\right| \leq \frac{2|q|\,|Z_{\text{Tutte}}(H';q,\tau)|}{|1-q|}c^{3r} \leq \frac{2|q|}{|1-q|}c^{-j}.$$

Our algorithm then computes $\hat{\varepsilon}q/(1-q-\hat{\varepsilon})$ as an approximation of $\alpha = Z_{s|t}(H;q,\gamma)/Z_{st}(H;q,\gamma)$. We have shown that α is a real algebraic number that we can additively approximate up to

an error at most $2|q|c^{-j}/|1-q|$ in polynomial time in j and the size of H. Technically, our approximation $\hat{\varepsilon}q/(1-q-\hat{\varepsilon})$ is another algebraic number. For this reason, our algorithm approximates $\hat{\varepsilon}q/(1-q-\hat{\varepsilon})$ by a rational number $\overline{\alpha}$ (with additive error at most $2|q|c^{-j}/|q-1|$) and uses this rational number as our approximation of α. The overall error that we make is then $|\alpha-\overline{\alpha}| \leq 4|q|c^{-j}/|1-q|$.

In view of Corollary 2.36, we have $d(\alpha) \leq d(q)d(\gamma) \leq d(q)d(\gamma_2)$, where we have used that $\gamma \in \mathbb{Q}(\gamma_2)$ and, thus, $d(\gamma) \leq d(\gamma_2)$. Moreover, Corollary 2.36 yields

$$H(\alpha) \leq \left(2^{m+1/2}e^{nh(q)+mh(\gamma)}\right)^{2d(q)d(\gamma)}.$$

Since $h(\gamma) = h((\gamma_2-1)^\rho - 1) \leq \rho(1+h(\gamma_2))$ by Lemma 2.35, our algorithm can compute a rational number D_{q,γ_2} with $D_{q,\gamma_2} > 1$ such that $H(\alpha) \leq D_{q,\gamma_2}^{\rho\,\mathrm{size}(H)}$. The only non-trivial step of this computation is upper bounding $h(q)$ and $h(\gamma_2)$ in terms of the degrees and usual heights of q and γ_2 as in (2.20). Let $d = d(q)d(\gamma_2) = O(1)$ and $U = D_{q,\gamma_2}^{\rho\,\mathrm{size}(H)}$. Let b be as in Lemma 2.37. Then we have $2^b = O(D_{q,\gamma_2}^{2d\,\rho\,\mathrm{size}(H)})$, so $b = O(\rho\,\mathrm{size}(H))$. By choosing j appropriately, we can use the algorithm that we have developed in this proof to find a rational approximation $\overline{\alpha}$ with $|\alpha-\overline{\alpha}| \leq 2^{-b}/(12d)$. As we have argued, this takes polynomial time in b and $\mathrm{size}(H)$. Since $b = O(\rho\,\mathrm{size}(H))$, we conclude that the computation of $\overline{\alpha}$ runs in polynomial time in ρ and $\mathrm{size}(H)$. Once we have computed this approximation, our algorithm invokes the algorithm given in Lemma 2.37 to determine the minimal polynomial of α in time $O(d^5(d+\log U)) = O(\rho\,\mathrm{size}(H))$. Finally, it remains to compute an interval of the real line where α is the only root of its minimal polynomial. Since α is a real algebraic number and we know its minimal polynomial, our algorithm can use Sturm sequences to isolate the real roots of this minimal polynomial. Then, by approximating α it decides which one of the computed intervals corresponds to α.

Finally, note that our algorithm also works for arbitrary graphs (not-necessarily planar) as long as our oracle provides us with reliable answers for any graph. □

Lemma 2.42. *Let q, γ_1 and γ_2 be real algebraic numbers such that $q > 1$, $\gamma_1 \in (-2,-1)$ and $\gamma_2 > 0$. Let us assume that we have access to an oracle for the computational problem* Sign-PlanarTutte(q,γ_1,γ_2). *Then there exists an algorithm that takes as input a positive integer ρ and a planar graph H along with two distinct connected vertices s and t of H, and, for $\gamma = (\gamma_2+1)^\rho - 1$, this algorithm computes a representation of the algebraic number $Z_{s|t}(H;q,\gamma)/Z_{st}(H;q,\gamma)$ in polynomial time in ρ and the size of H. Moreover, if we have access to the more powerful oracle* Sign-Tutte(q,γ_1,γ_2), *then we can remove the constraint that H is planar.*

Proof. The algorithm is exactly the same one of Lemma 2.41. The proof is analogous too. The only difference is in the interval-shrinking technique, where we split $(\varepsilon',\varepsilon'')$ into 4 intervals instead of 10 (so $p = 4$ in the proof), but this has been discussed at the beginning of this section. □

Lemma 2.43. *Let K be a real number with $K > 1$. Let q, γ_1 and γ_2 be real algebraic numbers such that $0 < q < 1$, $\gamma_1 \in (-1, 0)$ and $\gamma_2 > 0$. Let us assume that we have access to an oracle for* Factor-K-NormPlanarTutte(q, γ_1, γ_2). *Then there exists an algorithm that takes as input a positive integer ρ and a planar graph H along with two distinct connected vertices s and t of H, and, for $\gamma = (\gamma_2 + 1)^\rho - 1$, this algorithm computes a representation of the algebraic number $Z_{s|t}(H; q, \gamma)/Z_{st}(H; q, \gamma)$ in polynomial time in ρ and the size of H. Moreover, if we have access to the more powerful oracle* Factor-K-NormTutte(q, γ_1, γ_2), *then we can remove the constraint that H is planar.*

Proof. We claim that the algorithm presented in Lemma 2.41 also works in this setting. Let $f(\varepsilon; H, \gamma) = B(H, \gamma) - \varepsilon A(H, \gamma)$ as in (2.21). As we pointed out in the proof of Lemma 2.41, the algorithm works as long as $f(0; H, \gamma) f(1 - q; H, \gamma) < 0$ and the hypothesis of Lemma 2.40 hold. First, since q and γ are positive, equations (2.25) hold. It follows that $A(H, \gamma) = -Z_{st}(H; q, \gamma) - q^{-1} Z_{st}(H; q, \gamma) \neq 0$. Hence, the hypothesis of Lemma 2.40 hold. In view of (2.22) and the fact that $q \in (0, 1)$ and γ is positive, we have $f(0; H, \gamma) < 0$ and $f(1-q; H, \gamma) > 0$. We conclude that $f(0; H, \gamma) f(1 - q; H, \gamma) < 0$, as we wanted.

This time the interval-stretching technique applied in Lemma 2.41 runs on a subinterval $(\varepsilon', \varepsilon'')$ of $(0, 1 - q)$, so we only need to implement positive values of ε. For this reason, we can get away with the hypothesis $\gamma_1 \in (-1, 0)$ instead of the hypothesis $\gamma_1 \in (-2, -1)$, as was announced in the proof of Lemma 2.41. Finally, we must indicate how our algorithm implements the numbers ε' and ε'' so that $\varepsilon^* \in (\varepsilon', \varepsilon'') \subseteq (0, 1-q)$, as this was only done in Lemma 2.41 for $q > 1$. The argument that we give here also applies when $q < 0$. Let $y_1 = \gamma_1 + 1$, $x_1 = 1 + q/(y_1 - 1)$, $y_2 = \gamma_2 + 1$ and $x_2 = 1 + q/(y_2 - 1)$. We have $y_1 \in (0, 1)$, $y_2 > 1$, $q < 1$ and $q \neq 0$. Our algorithm first applies the algorithm given in Corollary 2.9 with $y' = |1 - 1/q|\, c^{-2r}/2$, k such that $|y_1|^k < |y'| < |y_1|^{-k}$ and $n = \lceil 2r \log_2(c) - \min\{0, \log_2 |1 - 1/q|\} + 2 \rceil$. Note that $k = O(r)$ and $n = O(r)$. This procedure computes a theta graph and a weight function taking weights in $\{\gamma_1, \gamma_2\}$ that implement a point $(1 + q/(\varepsilon' - 1), \varepsilon')$ such that $|y' - \varepsilon'| \leq 2^{-n} \leq |1 - 1/q| c^{-2r}/4$ in polynomial time in $r = O(\text{size}(H))$. We obtain $|1 - 1/q| c^{-2r}/4 \leq \varepsilon' \leq 3|1 - 1/q| c^{-2r}/4$, so, by Lemma 2.40, we find that $0 < \varepsilon' < \varepsilon^*$. Next our algorithm invokes again Corollary 2.9, this time with inputs $y' = 1 - q - (1 - q)c^{-2r}/2$, k such that $|y_1|^k < |y'| < |y_1|^{-k}$ and $n = \lceil 2r \log_2(c) - \min\{0, \log_2(1 - q)\} + 2 \rceil$. This implements $(1 + q/(\varepsilon'' - 1), \varepsilon'')$ with $|y' - \varepsilon''| \leq (1 - q)c^{-2r}/4$, which gives $1 - q - 3(1 - q)c^{-2r}/4 \leq \varepsilon'' \leq 1 - q + (1 - q)c^{-2r}/4$. Again by Lemma 2.40, we find that $\varepsilon^* < \varepsilon'' < 1 - q$. The interval $(\varepsilon', \varepsilon'')$ is the starting interval for the interval-shrinking procedure that we needed. □

Lemma 2.44. *Let q, γ_1 and γ_2 be real algebraic numbers such that $0 < q < 1$, $\gamma_1 \in (-1, -0)$ and $\gamma_2 > 0$. Let us assume that we have access to an oracle for the computational problem* Sign-PlanarTutte(q, γ_1, γ_2). *Then there exists an algorithm that takes as input a positive integer ρ and a planar graph H along with two distinct connected vertices s and t of H, and, for $\gamma = (\gamma_2 + 1)^\rho - 1$, this algorithm computes a representation of the algebraic number $Z_{s|t}(H; q, \gamma)/Z_{st}(H; q, \gamma)$ in polynomial time in ρ and the size of H. Moreover, if we have access to the more powerful oracle* Sign-Tutte(q, γ_1, γ_2), *then we can remove the constraint that H is*

planar.

Proof. The algorithm is exactly the same the one of Lemma 2.43, the only difference being in the interval-stretching technique as we have already explained. □

Lemma 2.45. *Let K be a real number with $K > 1$. Let q, γ_1 and γ_2 be real algebraic numbers such that $q < 0$, $\gamma_1 \in (-1, 0)$ and $\gamma_2 > 0$. Let us assume that we have access to an oracle for* Factor-K-NormPlanarTutte(q, γ_1, γ_2). *Then there exists an algorithm that takes as input:*

- *a positive integer ρ ;*
- *a planar graph $H = (V, E)$ such that, for $\gamma = (\gamma_2+1)^\rho - 1$, we have $\gamma \geq (8 \max\{|q|, 1/|q|\})^r$, where $r = \max\{|V|, |E|\}$;*
- *two distinct connected vertices s and t of H.*

This algorithm computes a representation of the algebraic number $Z_{s|t}(H; q, \gamma)/Z_{st}(H; q, \gamma)$ in polynomial time in ρ and the size of H. Moreover, for such inputs ρ, H and s, t, we have $Z_{st}(H; q, \gamma) \neq 0$ and $Z_{Tutte}(H; q, \gamma) \neq 0$. If we have access to the more powerful oracle Factor-K-NormTutte(q, γ_1, γ_2), *then we can remove the constraint that H is planar.*

Proof. We claim that the algorithm presented in Lemmas 2.41 and 2.43 also works in this setting. Let $n = |V|$ and $m = |E|$. Let $c = 2 \max\{|q|, 1/|q|\}\gamma$. We may assume that $r \geq 2$. First, let us assume that H is connected. Let $f(\varepsilon; H, \rho) = B(H, \gamma) - \varepsilon A(H, \gamma)$ as in (2.21), so $B(H, \gamma) = Z_{s|t}(H; q, \gamma)(1 - 1/q)$ and $A(H, \gamma) = -Z_{st}(H; q, \gamma) - q^{-1} Z_{s|t}(H; q, \gamma)$. Recall that we have to prove that the conditions of Lemma 2.40 hold, as well as the inequality $f(0; H, \gamma) f(1 - q; H, \gamma) < 0$. Let $\delta = (2 \max\{|q|, 1/|q|\})^r/\gamma$. Note that $0 < \delta \leq 1/4$ because $\gamma \geq (8 \max\{|q|, 1/|q|\})^r$. Each one of the (at most 2^m) terms in $Z_{st}(H; q, \gamma)$, other than the term with all edges in A, has absolute value at most $\gamma^{m-1}|q| \max\{|q|, 1\}^{n-1} \leq \delta 2^{-m} \gamma^m |q|$. Since H is connected, the term with all edges in A is $q\gamma^m$. Thus, we have the inequalities

$$\gamma^m q - \delta\gamma^m |q| \leq Z_{st}(H; q, \gamma) \leq \gamma^m q + \delta\gamma^m |q| < 0. \tag{2.28}$$

In particular, $Z_{st}(H; q, \gamma) \neq 0$. It also follows that

$$|Z_{st}(H; q, \gamma)| \geq \gamma^m |q|(1 - \delta) \geq \gamma^m |q| 3/4 \geq c^{-r},$$

which is one of the conditions of Lemma 2.40. Recall that an (s, t)-cut of H is a subset A of edges of H such that any path from s to t in H has an edge in A. The size of this (s, t)-cut is the cardinality of A. Let k be the size of a minimum cardinality (s, t)-cut in H, and let C be the number of (s, t)-cuts of size k. We study the terms $q^{k(A)}\gamma^{|A|}$ appearing in $Z_{s|t}(H; q, \gamma)$, so A is a subset of E such that s and t are not connected in (V, A). Note that such an A is the complement of an (s, t)-cut and, hence, $|A| \leq m - k$. Moreover, if A is not the complement of an (s, t)-cut of size k, then the absolute value of $q^{k(A)}\gamma^{|A|}$ is at most $\gamma^{m-k-1} q^2 \max\{1, |q|\}^{n-2} \leq \delta 2^{-m} \gamma^{m-k} q^2$. Thus, we have the inequalities

$$0 < C\gamma^{m-k} q^2 - \delta\gamma^{m-k} q^2 \leq Z_{s|t}(H; q, \gamma) \leq C\gamma^{m-k} q^2 + \delta\gamma^{m-k} q^2. \tag{2.29}$$

The inequalities (2.28) and (2.29) have been previously given in the proof of [59, Lemma 2]. As a consequence, we find that

$$\left|Z_{s|t}(H;q,\gamma)\right| \geq C\gamma^{m-k}q^2(1-\delta) \geq C\gamma^{m-k}q^2 3/4 \geq \gamma^{m-k}q^2 3/4 \geq c^{-r},$$

which is another one of the conditions of Lemma 2.40. In view of (2.22) and the facts that $q<0$ and we know the signs of $Z_{s|t}(H;q,\gamma)$ and $Z_{st}(H;q,\gamma)$, it follows that $f(0;H,\gamma)>0$ and $f(1-q;H,\gamma)<0$. Hence, we find that $f(0;H,\gamma)f(1-q;H,\gamma)<0$, as we wanted. Note that $A(H,\gamma)$ has to be non-zero because $f(-;H,\gamma)$ is non-constant as $f(0;H,\gamma)f(1-q;H,\gamma)<0$. This is the last condition of Lemma 2.40 that we had to check. We conclude that we can apply the algorithm given in the proof of Lemma 2.43 to compute $Z_{s|t}(H;q,\gamma)/Z_{st}(H;q,\gamma)$ in polynomial time in ρ and the size of H. Finally, we show that $Z_{\text{Tutte}}(H;q,\gamma)\neq 0$. This is not needed for the algorithm of Lemma 2.43, but is part of the statement of the current lemma. In light of (2.28) and (2.29), we have $|Z_{st}(H;q,\gamma)| \geq \gamma^m|q|(1-\delta)$ and $\left|Z_{s|t}(H;q,\gamma)\right| \leq C\gamma^{m-k}q^2(1+\delta)$. Note that

$$\gamma^m|q|(1-\delta) \geq \frac{3}{4}\gamma^m|q| > \frac{5}{4}C\gamma^{m-k}q^2 \geq \gamma^{m-k}q^2(1+\delta),$$

where we used that $\gamma \geq (8\max\{|q|,1/|q|\})^r \geq 8\cdot 2^m|q| > 5C|q|$ since $r\geq 2$. Therefore, we find that $|Z_{st}(H;q,\gamma)| > \left|Z_{s|t}(H;q,\gamma)\right|$. We conclude that

$$Z_{\text{Tutte}}(H;q,\gamma) = Z_{st}(H;q,\gamma) + Z_{s|t}(H;q,\gamma) < 0.$$

It remains to consider the case where H is not connected. Let $H_1,\ldots,H_l$ be the connected components of H, and let us assume that the vertices s and t are in H_1 without loss of generality. We have

$$\begin{aligned}
Z_{st}\left(H;q,\gamma\right) &= Z_{st}\left(H_1;q,\gamma\right) Z_{\text{Tutte}}\left(H_2;q,\gamma\right)\cdots Z_{\text{Tutte}}\left(H_l;q,\gamma\right);\\
Z_{s|t}\left(H;q,\gamma\right) &= Z_{s|t}\left(H_1;q,\gamma\right) Z_{\text{Tutte}}\left(H_2;q,\gamma\right)\cdots Z_{\text{Tutte}}\left(H_l;q,\gamma\right);\\
Z_{\text{Tutte}}\left(H;q,\gamma\right) &= Z_{\text{Tutte}}\left(H_1;q,\gamma\right) Z_{\text{Tutte}}\left(H_2;q,\gamma\right)\cdots Z_{\text{Tutte}}\left(H_l;q,\gamma\right).
\end{aligned}$$

We have already shown that $Z_{st}\left(H_1;q,\gamma\right)$, $Z_{st}\left(H_1;q,\gamma\right)$ and $Z_{\text{Tutte}}\left(H_j;q,\gamma\right)$ are non-zero for all j. Hence, we obtain $Z_{s|t}(H;q,\gamma)/Z_{st}(H;q,\gamma) = Z_{s|t}(H_1;q,\gamma)/Z_{st}(H_1;q,\gamma)$, and we can apply our algorithm to H_1 instead of H. Moreover, we have $Z_{st}\left(H;q,\gamma\right)\neq 0$ and $Z_{\text{Tutte}}(H;q,\gamma)\neq 0$ as we wanted. This finishes the proof. □

Lemma 2.46. *Let q, γ_1 and γ_2 be real algebraic numbers such that $q<0$, $\gamma_1\in(-1,0)$ and $\gamma_2>0$. Let us assume that we have access to an oracle for* Sign-PlanarTutte(q,γ_1,γ_2). *Then there exists an algorithm that takes as input:*

- *a positive integer ρ ;*
- *a planar graph $H=(V,E)$ such that, for $\gamma=(\gamma_2+1)^\rho-1$, we have $\gamma\geq(8\max\{|q|,1/|q|\})^r$, where $r=\max\{|V|,|E|\}$;*

- *two distinct connected vertices s and t of H.*

This algorithm computes a representation of the algebraic number $Z_{s|t}(H;q,\gamma)/Z_{st}(H;q,\gamma)$ in polynomial time in ρ and the size of H. Moreover, for such inputs ρ, H and s,t, we have $Z_{st}(H;q,\gamma) \neq 0$ and $Z_{Tutte}(H;q,\gamma) \neq 0$. If we have access to the more powerful oracle Sign-Tutte(q,γ_1,γ_2)*, then we can remove the constraint that H is planar.*

Proof. The algorithm is the same one as that of Lemma 2.45, the only difference being in the interval-stretching technique, as we have already explained. □

Now we deal with the last part of our reduction, where we reduce the computation of $Z_{\text{Tutte}}(G;q,\gamma)$ to the computation of $Z_{s|t}(H;q,\gamma)/Z_{st}(H;q,\gamma)$ on the subgraphs H of G. First, let us introduce some notation.

Definition 2.47. *We say that a pair (q,γ) of algebraic numbers is* zero-free *for a graph G if $q \neq 0$ and, for every subgraph H of G and every pair of distinct vertices s and t in the same connected component of H, the quantities $Z_{st}(H,q,\gamma)$ and $Z_{\text{Tutte}}(H,q,\gamma)$ are non-zero.*

Note that if (q,γ) is zero-free for G, then (q,γ) is also zero-free for any subgraph of H. We consider the following computational problems.

Name: RatioTutte(q,γ).
Instance: A (multi)graph $G=(V,E)$ such that (q,γ) is zero-free for G and two distinct vertices s and t in the same connected component of G.
Output: A representation of the algebraic number $Z_{s|t}(G;q,\gamma)/Z_{st}(G;q,\gamma)$.

Name: ZeroFreeTutte(q,γ).
Instance: A (multi)graph $G=(V,E)$ such that (q,γ) is zero-free for G.
Output: A representation of the algebraic number $Z_{\text{Tutte}}(G;q,\gamma)$.

We also consider the planar versions of these problems, RatioPlanarTutte(q,γ) and ZeroFreePlanarTutte(q,γ). Then we can express the last part of our reduction as a reduction between these two computational problems.

Lemma 2.48. *Let q and γ be algebraic numbers with $q \neq 0$. Then we have the reductions*

$$\text{ZeroFreePlanarTutte}(q,\gamma) \leq_T \text{RatioPlanarTutte}(q,\gamma),$$
$$\text{ZeroFreeTutte}(q,\gamma) \leq_T \text{RatioTutte}(q,\gamma).$$

Proof. First, we show ZeroFreePlanarTutte$(q,\gamma) \leq_T$ RatioPlanarTutte(q,γ). Let G be the input of ZeroFreePlanarTutte(q,γ). The reduction computes a representation of $Z_{\text{Tutte}}(G;q,\gamma)$ as follows. We assume that G is not a tree since it is known how to compute the Tutte polynomial of a tree in polynomial time [109, Example 2.1]. Then we can find an edge $e=(s,t)$ of G that is not a bridge. We are going to use the oracle for RatioPlanarTutte(q,γ)

to reduce the computation of $Z_{\text{Tutte}}(G;q,\gamma)$ to that of $Z_{\text{Tutte}}(G\setminus e;q,\gamma)$, where $G\setminus e$ is formed from G by deleting e. Note that if G is planar, then $G\setminus e$ is also planar. Since (q,γ) is zero-free for G, we have $Z_{st}(G;q,\gamma)\neq 0$. Let $\alpha = Z_{s|t}(G;q,\gamma)/Z_{st}(G;q,\gamma)$. First, note that

$$Z_{\text{Tutte}}(G;q,\gamma) = Z_{st}(G;q,\gamma) + Z_{s|t}(G;q,\gamma) = Z_{st}(G;q,\gamma)(1+\alpha).$$

By calling the oracle the algorithm obtains a representation of the factor $1+\alpha$. Since e is not a bridge, s and t are connected in $G\setminus e$, so, by calling the oracle again, the algorithm has access to a representation of the algebraic number $\beta = Z_{s|t}(G\setminus e;q,\gamma)/Z_{st}(G\setminus e;q,\gamma)$. We have

$$\begin{aligned} Z_{st}(G;q,\gamma) &= Z_{st}(G\setminus e;q,\gamma)(1+\gamma) + \gamma q^{-1} Z_{s|t}(G\setminus e;q,\gamma) \\ &= Z_{\text{Tutte}}(G\setminus e;q,\gamma)\left(\frac{1+\gamma}{1+\beta} + \gamma q^{-1}\frac{\beta}{1+\beta}\right), \end{aligned}$$

where we multiplied and divided by $Z_{\text{Tutte}}(G\setminus e;q,\gamma) = Z_{st}(G\setminus e;q,\gamma)(1+\beta)$, which is non-zero since (q,γ) is zero-free for G. Note that the fact that $Z_{\text{Tutte}}(G\setminus e;q,\gamma)\neq 0$ is equivalent to $\beta\neq -1$. We obtain

$$Z_{\text{Tutte}}(G;q,\gamma) = Z_{\text{Tutte}}(G\setminus e;q,\gamma)\left(1+\gamma+\gamma q^{-1}\beta\right)\frac{1+\alpha}{1+\beta}. \tag{2.30}$$

The algorithm then computes a representation of $Z_{\text{Tutte}}(G\setminus e;q,\gamma)$ recursively. Note that this reduction also works between the non-planar versions of the problems. □

In the rest of this section we put our reduction together. There is one result for each one of the cases $q>1$, $0<q<1$ and $q<0$ (see Lemmas 2.49, 2.50 and 2.53).

Lemma 2.49. *Let K be a real number with $K>1$. Let q, γ_1 and γ_2 be real algebraic numbers such that $q>1$, $\gamma_1\in(-2,-1)$ and $\gamma_2>0$. Then we have the following reductions:*

$$\begin{aligned} &\textsc{PlanarTutte}(q,\gamma_2) \leq_T \textsc{Factor-}K\textsc{-NormPlanarTutte}(q,\gamma_1,\gamma_2) \\ &\textsc{PlanarTutte}(q,\gamma_2) \leq_T \textsc{Sign-PlanarTutte}(q,\gamma_1,\gamma_2), \end{aligned}$$

where $\leq_T$ denotes a Turing reduction. Moreover, these reductions also hold for the analogous non-planar problems.

Proof. We claim that the problems $\textsc{PlanarTutte}(q,\gamma_2)$ and $\textsc{ZeroFreePlanarTutte}(q,\gamma_2)$ are equivalent. This follows from the fact that (q,γ_2) is zero-free for every graph G. Lemma 2.48 gives us a reduction from $\textsc{ZeroFreePlanarTutte}(q,\gamma_2)$ to $\textsc{RatioPlanarTutte}(q,\gamma_2)$. Recall that we have $q>0$, $\gamma_1\in(-2,-1)$ and $\gamma_2>0$. Thus, we can apply Lemma 2.41 with $\rho=1$ to obtain a reduction from the problem $\textsc{RatioPlanarTutte}(q,\gamma_2)$ to the problem $\textsc{Factor-}K\textsc{-NormPlanarTutte}(q,\gamma_1,\gamma_2)$, which gives the first reduction of the statement. The second reduction is derived analogously, but this time we apply Lemma 2.42 instead of Lemma 2.41. Finally, note that our reductions also hold for the non-planar version of the problems since the algorithms given in Lemma 2.41 and Lemma 2.42 work for arbitrary graphs (non-necessarily planar) as long as the oracle does. □

Lemma 2.50. *Let K be a real number with $K > 1$. Let q, γ_1 and γ_2 be real algebraic numbers such that $0 < q < 1$, $\gamma_1 \in (-1, 0)$ and $\gamma_2 > 0$. Then we have the following reductions:*

$$\textsc{PlanarTutte}(q, \gamma_2) \leq_T \textsc{Sign-PlanarTutte}(q, \gamma_1, \gamma_2),$$
$$\textsc{PlanarTutte}(q, \gamma_2) \leq_T \textsc{Factor-}K\textsc{-NormPlanarTutte}(q, \gamma_1, \gamma_2).$$

Moreover, these reductions also hold for the analogous non-planar problems.

Proof. The proof is analogous to that of Lemma 2.49; now, we instead combine Lemmas 2.48, 2.43 and 2.44. □

So far we have obtained reductions when $q > 1$ or $0 < q < 1$. To obtain a similar result when $q < 0$ we have to introduce the following variant of $\textsc{Tutte}(q, \gamma)$, where q is an algebraic number and γ is a positive real algebraic number.

Name: $\textsc{ThickenedTutte}(q, \gamma)$.
Instance: A (multi)graph $G = (V, E)$.
Output: A representation of the algebraic number $Z_{\text{Tutte}}(G; q, (\gamma+1)^{\rho(G)}-1)$, where $\rho(G)$ is the smallest positive integer such that $(\gamma+1)^{\rho(G)} - 1 > M(G)$ for $M(G) = (8\max\{|q|, 1/|q|\})^r$ and $r = \max\{|V|, |E|\}$.

We also consider the planar version of this problem, $\textsc{ThickenedPlanarTutte}(q, \gamma)$, where the input graph is promised to be planar.

Lemma 2.51. *Let q be an algebraic number and let γ be a real algebraic number with $\gamma > 0$. Then the problem $\textsc{ThickenedPlanarTutte}(q, \gamma)$ is #P-hard unless $q \in \{1, 2\}$, and the problem $\textsc{ThickenedTutte}(q, \gamma)$ is #P-hard unless $q = 1$.*

Proof. We are going to reduce $\textsc{PlanarTutte}(q, 2)$ to $\textsc{ThickenedPlanarTutte}(q, \gamma)$. The result then follows from the #P-hardness of $\textsc{PlanarTutte}(q, 2)$, cf. Theorem 2.39.

Let G be an m-edge instance of $\textsc{PlanarTutte}(q, 2)$. For $j = 1, \ldots, m$, let G_j be the graph obtained from G by j-thickening each of its edges. We have $M(G_j) = (8\max\{|q|, 1/|q|\})^{\max\{n, jm\}}$ so $M(G_j)$, and therefore $\rho(G_j)$, are non-decreasing in j. Let $\gamma_j = (\gamma+1)^{j\rho(G_j)} - 1$ and note that $Z_{\text{Tutte}}(G_j; q, (\gamma+1)^{\rho(G_j)} - 1) = Z_{\text{Tutte}}(G; q, \gamma_j)$. Note that the points $\gamma_1, \ldots, \gamma_m$ are distinct because $j\rho(G_j) \leq j\rho(G_{j+1}) < (j+1)\rho(G_{j+1})$ for every j. Moreover, their representation is polynomial in the size of G, and hence so is the representation of $Z_{\text{Tutte}}(G; q, \gamma_j)$.

The reduction constructs $G_1, \ldots, G_m$ and computes $Z_{\text{Tutte}}(G; q, \gamma_j)$ using the oracle for $\textsc{ThickenedPlanarTutte}(q, \gamma)$ with input G_j. By interpolation, we then recover the polynomial $Z_{\text{Tutte}}(G; q, x)$, whose degree is m when q is viewed as a constant, in time polynomial in the size of G. The reduction is then completed by evaluating $Z_{\text{Tutte}}(G; q, x)$ at $x = 2$.

Finally note that this reduction also works from $\textsc{Tutte}(q, 2)$ to $\textsc{ThickenedTutte}(q, \gamma)$. The only difference is that $\textsc{Tutte}(q, 2)$ is also #P-hard for $q = 2$ (see Theorem 2.38), so we also get #P-hardness in this case. □

We are going to reduce the problem ThickenedPlanarTutte(q, γ_2) to the problem Factor-K-NormPlanarTutte(q, γ_1, γ_2) for appropriate γ_1 and γ_2. In order to do so, we need to adapt Lemma 2.48 to this context. For this purpose, we consider the following computational problems.

Name: RatioThickenedTutte(q, γ).
Instance: A (multi)graph $G = (V, E)$, two distinct connected vertices s and t of G, and a positive integer ρ such that, for $\gamma_\rho = (\gamma + 1)^\rho - 1$, (q, γ_ρ) is zero-free for G and $\gamma_\rho > M(G)$, where $M(G) = (8 \max\{|q|, 1/|q|\})^r$ and $r = \max\{|V|, |E|\}$.
Output: A representation of the algebraic number $Z_{s|t}(G; q, \gamma_\rho)/Z_{st}(G; q, \gamma_\rho)$.

Name: ZeroFreeThickenedTutte(q, γ).
Instance: A (multi)graph $G = (V, E)$ and a positive integer ρ such that, for $\gamma_\rho = (\gamma + 1)^\rho - 1$, (q, γ_ρ) is zero-free for G and $\gamma_\rho > M(G)$, where $M(G) = (8 \max\{|q|, 1/|q|\})^r$ and $r = \max\{|V|, |E|\}$.
Output: A representation of the algebraic number $Z_{\text{Tutte}}(G; q, \gamma_\rho)$.

We also consider the planar versions of these problems, RatioThickenedPlanarTutte(q, γ) and ZeroFreeThickenedPlanarTutte(q, γ).

Lemma 2.52. *Let q and γ be algebraic numbers with $q \neq 0$. Then we have the reductions*

$$\text{ZeroFreeThickenedPlanarTutte}(q, \gamma) \leq_T \text{RatioThickenedPlanarTutte}(q, \gamma),$$
$$\text{ZeroFreeThickenedTutte}(q, \gamma) \leq_T \text{RatioThickenedTutte}(q, \gamma).$$

Proof. The reduction is almost exactly the one explained in Lemma 2.48. The only difference is that, for an input (G, ρ), each call to the oracle has as parameters a subgraph H of G, two vertices s and t determined in the reduction, and the same positive integer ρ. □

Lemma 2.53. *Let K be a real number with $K > 1$. Let q, γ_1 and γ_2 be real algebraic numbers such that $q < 0$, $\gamma_1 \in (-1, 0)$ and $\gamma_2 > 0$. Then we have the following reductions:*

$$\text{ThickenedPlanarTutte}(q, \gamma_2) \leq_T \text{Sign-PlanarTutte}(q, \gamma_1, \gamma_2),$$
$$\text{ThickenedPlanarTutte}(q, \gamma_2) \leq_T \text{Factor-}K\text{-NormPlanarTutte}(q, \gamma_1, \gamma_2).$$

Moreover, these reductions also hold for the analogous non-planar problems.

Proof. Let G and ρ be the inputs of ThickenedPlanarTutte(q, γ_2). Let H be a subgraph of G and let s and t be two distinct connected vertices of H. By applying Lemma 2.45 we find that $Z_{st}(H; q, (\gamma_2 + 1)^\rho - 1)$ and $Z_{\text{Tutte}}(H; q, (\gamma_2 + 1)^\rho - 1)$ are non-zero. Hence, $(q, (\gamma_2 + 1)^\rho - 1)$ is zero-free for G. This shows that ThickenedPlanarTutte(q, γ_2) reduces to ZeroFreeThickenedPlanarTutte(q, γ_2). Lemma 2.52 gives us a reduction from ZeroFreeThickenedPlanarTutte(q, γ_2) to RatioThickenedPlanarTutte(q, γ_2). Recall that we have $q < 0$, $\gamma_1 \in (-1, 0)$ and $\gamma_2 > 0$. Thus, Lemma 2.45 gives a reduction from

RatioThickenedPlanarTutte(q, γ_2) to Factor-K-NormPlanarTutte(q, γ_1, γ_2), which completes the proof for the first reduction of the statement. The second reduction is analogous, but this time we apply Lemma 2.46 instead of Lemma 2.45. Finally, note that our reductions also hold for the analogous non-planar problems since the algorithms given in Lemma 2.45 and Lemma 2.46 work for arbitrary graphs (non-necessarily planar) as long as the oracle does. □

2.5.6 The connection between approximate shifts and reductions

In this section we show how a polynomial-time approximate shift from (q, γ_1) to (q, γ_2) may allow us to reduce the problems of approximating the norm of the Tutte polynomial at (q, γ_2) to the same problem at (q, γ_1) (see Lemma 2.5). We also derive a similar result for the problem Distance-ρ-ArgTutte(q, γ) in Lemma 2.55.

Lemma 2.54. *Let q, γ_1 and γ_2 be algebraic numbers with $q \neq 0$ such that there is a polynomial-time series-parallel approximate shift from (q, γ_1) to (q, γ_2). Then there is an algorithm that has as input a graph G and a positive integer k and computes, in polynomial time in k and the size of G, a graph H and a representation of an algebraic number D with $D \neq 0$ such that*

$$\left| Z_{\mathit{Tutte}}(G; q, \gamma_2) - \frac{Z_{\mathit{Tutte}}(H; q, \gamma_1)}{D} \right| \leq 2^{-k}.$$

Moreover, if the graph G is planar, then the graph H is also planar, and if q and γ_1 are real, then D is also real.

Proof. Let $G = (V, E)$ and k be the inputs of the algorithm. Let $n = |V|$ and $m = |E|$. By the definition of series-parallel polynomial-time approximate shifts, for any positive integer j, one can compute, in polynomial time in j, a series-parallel graph J_j that γ_1-implements $\hat{\gamma}$ with $|\gamma_2 - \hat{\gamma}| \leq 2^{-j}$ for terminals s and t. By definition of implementations, we have $\hat{\gamma} = qZ_{st}(J_j; q, \gamma_1)/Z_{s|t}(J_j; q, \gamma_1)$ and $Z_{s|t}(J_j; q, \gamma_1) \neq 0$. We construct a graph G_j that is a copy of G where every edge f in G has been replaced by a copy of J_j as in Lemma 2.6, identifying the endpoints of f with s and t. In light of Lemma 2.6, we have

$$Z_{\mathrm{Tutte}}(G_j; q, \gamma_1) = \left(\frac{Z_{s|t}(J_j; q, \gamma_1)}{q^2} \right)^m Z_{\mathrm{Tutte}}(G; q, \hat{\gamma}).$$

We can compute a representation of $D_j = Z_{s|t}(J_j; q, \gamma_1)/q^2$ in polynomial time in the size of J_j because J_j is a series-parallel graph. However, note that this hypothesis is not essential as long as there is some way to compute a representation of D_j while constructing J_j. Note that $|\hat{\gamma}| \leq |\gamma_2| + 2^{-j}$, so $|Z_{\mathrm{Tutte}}(G; q, \gamma_2) - Z_{\mathrm{Tutte}}(G; q, \hat{\gamma})|$ is upper bounded by

$$\begin{aligned}
\sum_{A \subseteq E} |q|^{k(A)} \left| \gamma_2^{|A|} - \hat{\gamma}^{|A|} \right| &\leq \sum_{A \subseteq E} |q|^{k(A)} |\gamma_2 - \hat{\gamma}| \sum_{t=0}^{|A|-1} \left| \gamma_2^{|A|-1-t} \hat{\gamma}^t \right| \\
&\leq \sum_{A \subseteq E} |q|^{k(A)} |\gamma_2 - \hat{\gamma}| \, (|A| - 1) \left(|\gamma_2| + \frac{1}{2} \right)^{|A|-1} \\
&\leq |\gamma_2 - \hat{\gamma}| \, |q|^n \, 2^m (m-1) \left(|\gamma_2| + \frac{1}{2} \right)^{m-1}.
\end{aligned}$$

Hence, for j such that $2^{-j}|q|^n 2^m (m-1)(|\gamma_2|+1/2)^{m-1} \leq 2^{-k}$, which can be achieved for $j = O(\text{size}(G)+k)$, we obtain

$$\left| Z_{\text{Tutte}}(G;q,\gamma_2) - \frac{Z_{\text{Tutte}}(G_j;q,\gamma_1)}{D_j^m} \right| = |Z_{\text{Tutte}}(G;q,\gamma_2) - Z_{\text{Tutte}}(G;q,\hat{\gamma})| \leq 2^{-k}.$$

The algorithm returns $H = G_j$ and $D = D_j^m \neq 0$. Note that if G is planar, then $H = G_j$ is also planar by construction. If q and γ_1 are real, then the number $D = (Z_{s|t}(J_j;q,\gamma_1)/q^2)^m$ is clearly real too. □

In the rest of this section we use Lemma 2.54 to translate information about the function $Z_{\text{Tutte}}(-;q,\gamma_1)$ for certain graphs to information about $Z_{\text{Tutte}}(G;q,\gamma_2)$. This leads to the reductions given in Lemmas 2.5 and 2.55. These results are stated for polynomial series-parallel approximate shifts, but they would also hold even if the shifts are not series-parallel as long as, in the proof of Lemma 2.54, the graphs J_j are planar and we can compute $D_j = Z_{s|t}(J_j;q,\gamma_1)/q^2$ in polynomial time in the size of J_j.

We are now ready to prove Lemma 2.5, which was stated in the proof outline (Section 2.1) and which we restate here for convenience.

Lemma 2.5. *Let $q \neq 0$, γ_1 and $\gamma_2 \neq 0$ be algebraic numbers, and $K > 1$. For $j \in \{1,2\}$, let $y_j = \gamma_j + 1$ and $x_j = 1 + q/\gamma_j$. If there is a polynomial-time series-parallel approximate shift from (x_1,y_1) to (x_2,y_2), then we have a reduction from* Factor-K-NormTutte(q,γ_2) *to* Factor-K-NormTutte(q,γ_1). *This reduction also holds for the planar version of the problem.*

Proof. We are going to solve Factor-$4K$-NormTutte(q,γ_2) in polynomial time with the help of an oracle for Factor-K-NormTutte(q,γ_1). Recall that hardness of these problems does not depend on K (see Section 2.1). Let $C_{q,\gamma_2} > 1$ be the constant computed in Corollary 2.34 for the algebraic numbers q and $\gamma = \gamma_2$; so, for any graph G, either $Z_{\text{Tutte}}(G;q,\gamma_2) = 0$ or $|Z_{\text{Tutte}}(G;q,\gamma_2)| \geq C_{q,\gamma_2}^{-\text{size}(G)}$. Let $G = (V,E)$ be the input of the computational problem Factor-$4K$-NormTutte(q,γ_2). We assume that $Z_{\text{Tutte}}(G;q,\gamma_2) \neq 0$ since otherwise we can output anything. Let k be the smallest integer such that $2^{-k} \leq C_{q,\gamma_2}^{-\text{size}(G)}/2$. The reduction uses the algorithm given in Lemma 2.54 to compute a graph H and a representation of an algebraic number D with $D \neq 0$ such that

$$\left| Z_{\text{Tutte}}(G;q,\gamma_2) - \frac{Z_{\text{Tutte}}(H;q,\gamma_1)}{D} \right| \leq 2^{-k} \leq \frac{C_{q,\gamma_2}^{-\text{size}(G)}}{2} \leq \frac{|Z_{\text{Tutte}}(G;q,\gamma_2)|}{2}. \tag{2.31}$$

Therefore, we have

$$\frac{1}{2} \leq \frac{|Z_{\text{Tutte}}(H;q,\gamma_1)|}{D\,|Z_{\text{Tutte}}(G;q,\gamma_2)|} \leq \frac{3}{2}.$$

By invoking the oracle for Factor-K-NormTutte(q,γ_1), the reduction computes a rational number N with $N/K \leq |Z_{\text{Tutte}}(H;q,\gamma_1)| \leq KN$. The reduction also computes a non-zero rational number $\hat{D}$ such that $1/2 \leq D/\hat{D} \leq 2$. Then $\hat{N} = N/\hat{D}$ satisfies

$$\frac{\hat{N}}{4K} \leq |Z_{\text{Tutte}}(G;q,\gamma_2)| \leq 4K\hat{N},$$

so the reduction outputs $\hat{N}$ for Factor-$4K$-NormTutte(q, γ_2). Note that this reduction analogously applies to the planar case since the graph H is planar when G is planar (see Lemma 2.54). □

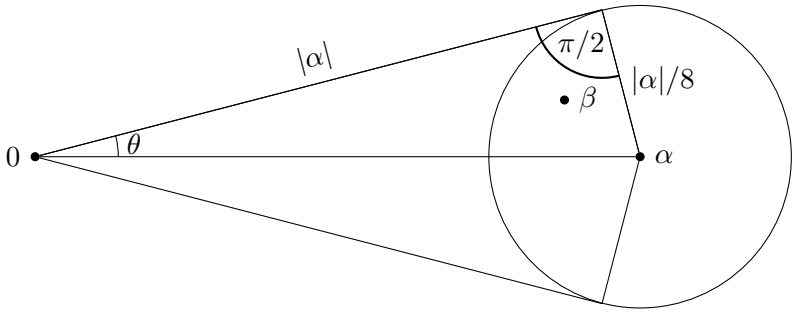
We next give the analogue of Lemma 2.5 for the argument.

Lemma 2.55. *Let q, γ_1 and γ_2 be algebraic numbers with $q \neq 0$. If there is a polynomial-time series-parallel approximate shift from (q, γ_1) to (q, γ_2), then we have the following reduction,* Distance-$5\pi/12$-ArgTutte$(q, \gamma_2) \leq_T$ Distance-$\pi/3$-ArgTutte(q, γ_1). *This reduction also holds for the planar version of the problem.*

Proof. Let C_{q,γ_2} be the constant computed in Corollary 2.34 for $\gamma = \gamma_2$. Let $G = (V, E)$ be the input of Distance-$\pi/2$-ArgTutte(q, γ_2). We assume that $Z_{\text{Tutte}}(G; q, \gamma_2) \neq 0$ since otherwise we can output anything. The reduction proceeds again similarly to that of Lemma 2.5. First, it applies Lemma 2.54 for appropriate k as in (2.31) to compute a graph H and a representation of a real algebraic number D with $D \neq 0$ such that

$$\left| Z_{\text{Tutte}}(G; q, \gamma_2) - \frac{Z_{\text{Tutte}}(H; q, \gamma_1)}{D} \right| \leq 2^{-k-2} \leq \frac{C_{q,\gamma_2}^{-\text{size}(G)}}{8} \leq \frac{|Z_{\text{Tutte}}(G; q, \gamma_2)|}{8}. \tag{2.32}$$

Let $\alpha = Z_{\text{Tutte}}(G; q, \gamma_2)$ and $\beta = Z_{\text{Tutte}}(H; q, \gamma_1)/D$, so (2.32) can be rewritten as $|\alpha - \beta| \leq |\alpha|/8$. We claim that $|\text{Arg}(\alpha) - \text{Arg}(\beta)| \leq \pi/24$. Since β is in the disc of centre α and radius $|\alpha|/8$, by basic geometry, we have



so $\sin(\theta) = 1/8$, where θ is the angle between 0, α and the intersection of the circle of radius $|\alpha|/8$ and center α with the tangent line that goes through 0. Since $\sin(\pi/24) > 1/8$, we conclude that $|\text{Arg}(\alpha) - \text{Arg}(\beta)| \leq \theta \leq \pi/24$ as we claimed. By invoking the oracle for Distance-$\pi/3$-ArgTutte(q, γ_1), the reduction computes a rational number $\hat{A}_1$ such that, for some $a_1 \in \arg(Z_{\text{Tutte}}(H; q, \gamma_1))$, we have $|a_1 - \hat{A}_1| \leq \pi/3$. Since the reduction has at its disposal a representation of the algebraic number D, it can compute (in polynomial time in the length of this representation) a rational number $\hat{A}_2$ such that, for some $a_2 \in \arg(D)$, we have $|a_2 - \hat{A}_2| \leq \pi/24$. The reduction outputs $\hat{A} = \hat{A}_1 - \hat{A}_2$. We claim that there is an argument a of α such that $|a - \hat{A}| \leq 5\pi/12$. Note that $b = a_1 - a_2$ is an argument of β. By the triangle inequality, we have $|b - \hat{A}| \leq |a_1 - \hat{A}_1| + |\hat{A}_2 - a_2| \leq 9\pi/24$. Let $a = \text{Arg}(\alpha) + (b - \text{Arg}(\beta))$, which is an argument of α. We conclude that

$$|a - \hat{A}| \leq |b - \hat{A}| + |a - b| = |b - \hat{A}| + |\text{Arg}(\alpha) - \text{Arg}(\beta)| \leq 5\pi/12.$$

This reduction analogously applies to the planar case since the graph H is planar when G is planar (see Lemma 2.54). □

One could actually change the angles $\rho_2 = 5\pi/12$ and $\rho_1 = \pi/3$ in the statement of Lemma 2.55 as long as $\rho_1 < \rho_2$, but $\rho_2 = 5\pi/12$ and $\rho_1 = \pi/3$ will suffice for our purposes.

2.5.7 Hardness for the Tutte polynomial

In this section we use the reductions of Section 2.5.5 to obtain intermediate hardness results that will be used to obtain our main theorems in the upcoming sections. We start with the following corollary which strengthens previous results of [59] (that applied to general graphs rather than planar).

Corollary 2.56. *Let $K > 1$ be a real number. Let $q \neq 0, 2$, and γ_1, γ_2 be real algebraic numbers with $\gamma_2 \in (-\infty, -2) \cup (0, \infty)$ and either*

- *$q > 1$, $\gamma_1 \in (-2, -1)$, or*
- *$q < 1$, $\gamma_1 \in (-1, 0)$.*

Then, Factor-K-NormPlanarTutte(q, γ_1, γ_2) *and* Sign-PlanarTutte(q, γ_1, γ_2) *are* #P*-hard.*

Proof. We consider first the case when $\gamma_2 > 0$. For q, γ_1, γ_2 as in the first item, the conclusion follows from Theorem 2.39 and the reductions given in Lemma 2.49. For the second item: when $q \in (0, 1)$, the result follows from the reductions given in Lemma 2.50 and Theorem 2.39, while for $q < 0$, the result follows from Lemmas 2.51 and 2.53.

The other case is when $\gamma_2 < -2$. Then, we can γ_2-implement $(\gamma_2 + 1)^2 - 1 > 0$ with a 2-thickening and proceed as in the previous case. □

Lemma 2.57. *Let K be a real number with $K > 1$. Let x, y be a real algebraic numbers such that $(x, y) \neq (-1, -1)$, $\min\{x, y\} \leq -1$ and $\max\{x, y\} < 0$. Let $q = (x-1)(y-1)$ and $\gamma = y - 1$. Then* Factor-K-NormPlanarTutte(q, γ) *and* Sign-PlanarTutte(q, γ) *are* #P*-hard.*

Proof. Note that $q > 2$. We claim that we can (x, y)-implement (x_1, y_2) with $y_1 \in (-1, 0)$, and (x_2, y_2) with $|y_2| > 1$ using planar (in fact, series-parallel) graphs. The result then follows by invoking Corollary 2.56 with $\gamma_1 = y_1 - 1$ and $\gamma_2 = y_2 - 1$.

The case $\min\{x, y\} < -1$ is treated in [59, Lemmas 8–11]. Hence, we may assume that $-1 \leq x < 0$ and $-1 \leq y < 0$. Since $(x, y) \neq (-1, -1)$ by hypothesis, there are two cases:

- $x = -1$ and $-1 < y < 0$. As pointed out in [59, Corollary 26], a 3-thickening from (x, y) implements the point $(x', y') = \left(1 - \frac{2}{1+y+y^2}, y^3\right)$ with $x' < -1$ and $y' \in (-1, 0)$, so the point (x', y') has already been studied in this proof.
- $-1 < x < 0$ and $y = -1$. This time we perform a 3-stretching from (x, y) to implement a point (x', y') with $x' \in (-1, 0)$ and $y' < -1$. □

Lemma 2.58. *Let $K > 1$ be a real number and q, x, y be real algebraic numbers with $\max\{|x|, |y|\} < 1$ and $q = (x-1)(y-1) > 32/27$. Then, for $\gamma = y - 1$,* Factor-K-NormPlanarTutte(q, γ) *and* Sign-PlanarTutte(q, γ) *are* #P*-hard, unless $q = 2$.*

Proof. In view of [59, Lemmas 12 and 15], we can (x, y)-implement points (x_1, y_1) and (x_2, y_2) with $y_1 \in (-1, 0)$ and $y_2 > 1$. These implementations only use series-parallel graphs. Hence, we can apply (the first item of) Corollary 2.56 with $\gamma_1 = y_1 - 1$ and $\gamma_2 = y_2 - 1$ to finish the proof. □

2.5.8 Proofs of the main theorems in this chapter

In this section we show how our main Theorems 1.1, 1.2, 1.3 and 1.4 follow from the #P-hardness results of Section 2.5.7. We start with Theorem 1.4.

Theorem 1.4. *Let $q > 2$ be a real, $\gamma \in \mathbb{C}\backslash\mathbb{R}$ be an algebraic number, and $K > 1$. Then,* Factor-K-NormPlanarTutte(q, γ) *and* Distance-$\pi/3$-ArgPlanarTutte(q, γ) *are* #P*-hard, unless $q = 3$ and $\gamma + 1 \in \{e^{2\pi i/3}, e^{4\pi i/3}\}$ when both problems can be solved exactly in polynomial time.*

Proof. Let $(x, y) \in \mathcal{H}_q$ be such that $y = \gamma + 1$. Consider the point $(x_2, y_2) \in H_q$ with $y_2 = -1/2$ and $x_2 = 1 + q/(y_2 - 1)$. Note that $x_2 = 1 - 2q/3 \leq 1 - 4/3 < 0$. There are two cases. Either $x_2 \leq -1$ and the point (x_2, y_2) satisfies the hypothesis of Lemma 2.57, or $-1 < x_2 < 0$ and the point (x_2, y_2) satisfies the hypothesis of Lemma 2.58. In any case, we conclude that Sign-PlanarTutte(q, γ_2) and Factor-K-NormPlanarTutte(q, γ_2) are #P-hard for $\gamma_2 = y_2 - 1$ when $q > 2$.

By Lemma 2.5 (for $\gamma_1 = \gamma$ and $\gamma_2 = \gamma_2$), we see that Factor-K-NormPlanarTutte(q, γ_2) reduces to Factor-K-NormPlanarTutte(q, γ), proving that the latter is #P-hard too. The proof for Distance-$\pi/3$-ArgPlanarTutte(q, γ) is analogous: first observe that since q, γ_2 are real and $5\pi/12 < \pi/2$, the problem Sign-PlanarTutte(q, γ_2) reduces (trivially) to Distance-$5\pi/12$-ArgPlanarTutte(q, γ_2). Moreover, applying Theorem 2.2 with x and y as above, $y' = y_2 \in (-1, 0)$ and $x' = x_2$, we have a polynomial-time series-parallel approximate shift from (x, y) to (x', y') or, equivalently, from (q, γ) to (q, γ_2). Using Lemma 2.55 with $\gamma_1 = \gamma$ and $\gamma_2 = \gamma_2$, we conclude that Distance-$5\pi/12$-ArgPlanarTutte(q, γ_2) reduces to Distance-$\pi/3$-ArgPlanarTutte(q, γ), proving that the latter is #P-hard, as wanted. □

Theorem 1.3. *Let $y \in \mathbb{C}\backslash\mathbb{R}$ be an algebraic number, and $K > 1$. Then,* Factor-K-NormIsing(y) *and* Distance-$\pi/3$-ArgIsing(y) *are* #P*-hard, unless $y = \pm i$ when both problems can be solved exactly in polynomial time.*

Proof. Let $q = 2, \gamma = y - 1, y_2 = -1/2, \gamma_2 = y_2 - 1$. From the result of Goldberg and Guo [55], the problems Factor-K-NormIsing(y_2) and Distance-$\pi/3$-ArgIsing(y_2) are #P-hard, hence Factor-K-NormPlanarTutte(q, γ_2) and Distance-$\pi/3$-ArgPlanarTutte(q, γ_2) are #P-hard as well, using that $Z_{\text{Ising}}(G; y_2) = Z_{\text{Tutte}}(G; 2, \gamma_2)$.

By applying Lemma 2.5 and Theorem 2.2 analogously to the proof of Theorem 1.4, we conclude that Factor-K-NormPlanarTutte(q,γ) and Distance-$\pi/3$-ArgPlanarTutte(q,γ) are #P-hard, and hence Factor-K-NormIsing(y) and Distance-$\pi/3$-ArgIsing(y), using that $Z_{\text{Ising}}(G;y) = Z_{\text{Tutte}}(G;2,\gamma)$. □

Theorem 1.1. *Let $q \geq 3$ be an integer, $y \in \mathbb{C}\backslash\mathbb{R}$ be an algebraic number, and $K > 1$. Then, the problems* Factor-K-NormPlanarPotts(q,y) *and* Distance-$\pi/3$-ArgPlanarPotts(q,y) *are* #P*-hard, unless $q = 3$ and $y \in \{e^{2\pi i/3}, e^{4\pi i/3}\}$ when both problems can be solved exactly in polynomial time.*

Proof. Just apply Theorem 1.4 to the integer q, and use $Z_{\text{Potts}}(G;q,y) = Z_{\text{Tutte}}(G;q,y-1)$. □

Theorem 1.2. *Let $q \geq 3$ be an integer, $y \in (-q+1,0)$ be a real algebraic number, and $K > 1$. Then* Factor-K-NormPlanarPotts(q,y) *and* Distance-$\pi/3$-ArgPlanarPotts(q,y) *are* #P*-hard, unless $(q,y) = (4,-1)$ when both problems can be solved exactly in polynomial time.*

Proof. Let $y \in (-q+1,0)$. The point (x,y) with $x = 1 + q/(y-1)$ satisfies $x \in (1-q,0)$, $(x,y) \neq (-1,-1)$ and $y < 0$. If $x \leq -1$ or $y \leq -1$, #P-hardness follows from Lemma 2.57. Otherwise, we have $q \geq 3$ and $x,y \in (-1,0)$, so hardness follows from Lemma 2.58. □

2.6 Further consequences of our results

In this final section, we discuss some further consequences of our techniques, as mentioned in Section 1.3.3.1. First, in Section 2.6.1, we explain how our results can be used to obtain hardness for Sign-PlanarTutte(q,γ) and Factor-K-NormPlanarTutte(q,γ) (and the non-planar version of these problems) at other parameters than the ones studied in Section 2.5.7, building on work of Goldberg and Jerrum [59]. Secondly, in Section 2.6.2, we apply our results to the problem of approximating the Jones polynomial of an alternating link, which is connected to the quantum complexity class BQP as explained in [20].

2.6.1 Hardness results for real algebraic parameters in the Tutte plane

The regions studied in Lemmas 2.57 and 2.58 have been studied by Goldberg and Jerrum [59], where they showed #P-hardness of SignPlanarTutte(q,γ) at several regions of the real algebraic plane. As we explained in Section 2.5.7, we obtain hardness at a point (q,γ) as long as we can γ-implement algebraic numbers γ_1 and γ_2 as in Corollary 2.56. Goldberg and Jerrum came up with multiple implementations that achieve the conditions of Corollary 2.56. By applying their implementations, we obtain #P-hardness for Factor-K-NormTutte$(q,y-1)$ in the same regions where they obtained #P-hardness of SignPlanarTutte(q,γ) in [59, Theorem 1].

Some of the implementations developed in [59] consist of planar graphs (as those used in Lemmas 2.57 and 2.58), so we can extend their results to the planar version of the problems for some of the previous regions.

Theorem 2.59. *Let q and γ be real algebraic numbers with $q \neq 0, 1, 2$. Let $y = \gamma + 1$ and $x = 1 + q/(y-1)$. The problems* Sign-PlanarTutte(q, γ) *and* Factor-K-NormPlanarTutte(q, γ) *are* #P*-hard when x, y are real algebraic numbers satisfying one of the following:*

1. $\min(x, y) \leq -1$, $\max(x, y) < 0$ *and* $(x, y) \neq (-1, -1)$,
2. $|x| > 1$, $|y| > 1$ *and* $xy < 0$,
3. $\max(|x|, |y|) < 1$ *and* $q > 32/27$,
4. $\max(|x|, |y|) < 1$, $q \leq 32/27$ *and* $x < -2y - 1$,
5. $\max(|x|, |y|) < 1$, $q \leq 32/27$ *and* $y < -2x - 1$.

Proof. The proof follows from the following results of [59], which show how to implement γ_1 and γ_2 with a planar (actually series-parallel) graph as in Corollary 2.56 for each of the regions in the statement.

Item 1 follows from Lemma 2.57. For Item 2, note that $q < 0$, so we have to implement $\gamma_1 \in (-1, 0)$ and $\gamma_2 \notin [-2, 0]$. We choose $\gamma_2 = y - 1$ and γ_1 as implemented in [59, Lemma 16]. Item 3 follows from Lemma 2.57. For Item 4, we implement $\gamma_1 \in (-1, 0)$ and $\gamma_2 \notin [-2, 0]$; the implementations are as in [59, Lemmas 14 and 15]. For Item 5, we implement $\gamma_1 \in (-1, 0)$ and $\gamma_2 \notin [-2, 0]$; the implementations are as in [59, Lemmas 13 and 15]. □

The complexity of approximating the Tutte polynomial of a planar graph has previously been studied in [58] and [84]. Our result on this matter (Theorem 2.59) strengthens the results of [58] in three directions. First, we also study the complexity of determining the sign of the Tutte polynomial. Secondly, we find new regions where the approximation problem is hard. These regions are 3, 4 and 5, as well as the points in region 1 such that $q \leq 5$ and $q \neq 3$. Finally, we prove #P-hardness, whereas in [58] hardness was obtained under the hypothesis that RP $\neq$ NP.

For $q \in \mathbb{Z}^+$, let $P(G; q)$ count the number of proper q-colourings of a graph G. The *chromatic polynomial* of G is the only polynomial that agrees with $P(G; q)$ on positive integers. It is well-known that $P(G; q) = Z_{\text{Tutte}}(G; q, -1)$, see for instance [109]. The value $q = 32/27$ appearing in Theorem 2.59 is, in some sense, a phase transition for the complexity of computing the sign of $P(G, q)$: this sign depends upon G in an essentially trivial way for $q < 32/27$ [72, Theorem 5] and its computation is #P-hard for $q > 32/27$, see [59] for an in detail discussion of the relevance of the phase transition $q = 32/27$.

2.6.2 Hardness results for the Jones polynomial

We briefly review some relevant facts about links and the Jones polynomial that relate it to the Tutte polynomial on graphs, see [121] for their definitions. Let $V_L(T)$ denote the Jones polynomial of a link L. By a result of Thistlethwaite, when L is an alternating link with associated planar graph $G(L)$, we have $V_L(t) = f_L(t)T(G(L); -t, -t^{-1})$, where $f_L(t)$ is an easily-computable factor that is plus or minus a half integer power of t, and $T(G; x, y)$ is the

Tutte polynomial of G in the (x, y)-parametrisation [114, 121]. Moreover, every planar graph is the graph of an alternating link [121, Chapter 2]. Hence, we can translate our results on the complexity of approximating the Tutte polynomial of a planar graph to the complexity of approximating the Jones polynomial of an alternating link, and obtain #P-hardness results for approximating $V_L(t)$. More formally, we consider the following problems for $K > 0$ and $\rho > 0$.

Name: Factor-K-NormJones(t).
Instance: A link L.
Output: If $V_L(t) = 0$, the algorithm may output any rational number. Otherwise, it must output $\hat{N} \in \mathbb{Q}$ such that $\hat{N}/K \leq |V_L(t)| \leq K\hat{N}$.

Name: Distance-ρ-ArgJones(q, γ).
Instance: A link L.
Output: If $V_L(t) = 0$, the algorithm may output any rational number. Otherwise, it must output $\hat{A} \in \mathbb{Q}$ such that, for some $a \in \arg(V_L(t))$, we have $|\hat{A} - a| \leq \rho$.

Corollary 2.60. *Let K be a real number with $K > 1$. Let t be an algebraic number with $\mathrm{Re}(t) > 0$. Then* Factor-K-NormJones(t) *and* Distance-$\pi/3$-ArgJones(t) *are* #P*-hard unless $t \in \{1, -e^{2\pi i/3}, -e^{4\pi i/3}\}$ when both problems can be solved exactly.*

Proof. Let us consider the point $(x, y) = (-t, -t^{-1})$ in the Tutte plane. Note that $t \in \{1, -e^{2\pi i/3}, -e^{4\pi i/3}\}$ if and only if (x, y) is one of the special points $(-1, -1)$, $(e^{4\pi i/3}, e^{2\pi i/3})$ and $(e^{2\pi i/3}, e^{4\pi i/3})$, where the Jones polynomial of a link can be exactly evaluated in polynomial time in the size of the link [73]. Let us assume that t is not one of these three values. We have $q = (-t - 1)(-t^{-1} - 1) = 2 + 2\mathrm{Re}(t) > 2$. When t is non-real, in view of Theorem 1.4, Factor-K-NormPlanarTutte($q, y - 1$) and Distance-$\pi/3$-ArgPlanarTutte($q, y - 1$) are #P-hard and the result follows. When t is real, note that $y < 0$, $x < 0$ and $q > 2$. Thus, either (x, y) is such that $\max\{|x|, |y|\} \geq 1$ and $(x, y) \neq (-1, -1)$, so hardness is covered in region 1 of Theorem 2.59, or $\max\{|x|, |y|\} < 1$, so hardness is covered in region 3 of Theorem 2.59. □

The case $t = e^{2\pi i/5}$ of Corollary 2.60 is particularly relevant due to its connection with quantum computation. This connection between approximate counting and the quantum complexity class BQP was explored by Bordewich, Freedman, Lovász and Welsh in [20], where they posed the question of determining the complexity of the following problem:

Name: Sign-Real-PlanarTutte(q, γ)
Instance: A planar (multi)graph G.
Output: Determine whether $\mathrm{Re}(Z_{\mathrm{Tutte}}(G; q, \gamma)) \geq 0$ or $\mathrm{Re}(Z_{\mathrm{Tutte}}(G; q, \gamma)) \leq 0$.

The non-planar version of Sign-Real-PlanarTutte(q, γ) has been studied in [55, Section 5], where it was shown that determining the sign of the real part of the Tutte polynomial is #P-hard in certain cases that include $t = e^{2\pi i/5}$. Our results on the complexity of

Sign-PlanarTutte(q, γ) allow us to adapt the argument in [55] to answer the question asked in [20].

Corollary 2.61. *Consider the point* $(x, y) = (\exp(-a\pi i/b), \exp(a\pi i/b))$*, where* a *and* b *are positive integers such that* $1/2 < a/b < 3/2$ *and* $a \neq b$*. Let* $q = (x-1)(y-1)$ *and* $\gamma = y - 1$*. Then* $q \in (2, 4)$ *and* Sign-Real-PlanarTutte(q, γ) *is* #P*-hard.*

Proof. The proof is essentially the same one as that of [55, Theorem 1.7]. First, note that

$$q = (x-1)(y-1) = 2 - x - y = 2 - \exp(-a\pi i/b) - \exp(a\pi i/b) = 2 - 2\cos(a\pi/b),$$

which is real. Since $1/2 < a/b < 3/2$ and $a \neq b$, we have $q \in (2, 4)$. A b-thickening allows us to (x, y)-implement $(1-q/2, -1)$. Since Sign-PlanarTutte$(q, -2)$ is #P-hard (see Theorem 2.59), we conclude that Sign-Real-PlanarTutte(q, γ) is #P-hard. □

Corollary 2.61 includes the case where $a = 3$ and $b = 5$. In this case, we have $x = \exp(-a\pi i/b) = -\exp(\pi i)\exp(-3\pi i/5) = -\exp(2\pi i/5)$ and $y = x^{-1}$. That is, $(x, y) = (-t, -t^{-1})$ for $t = \exp(2\pi i/5)$, which is the point of interest in [20].

Chapter 3

The complexity of approximating the complex-valued Ising model on bounded degree graphs

◦ This chapter is based on the following publication:
Andreas Galanis, Leslie A. Goldberg, and Andres Herrera-Poyatos. The complexity of approximating the complex-valued Ising model on bounded degree graphs. *SIAM J. Discrete Math.*, 36(3):2159–2204, 2022. doi:10.1137/21M1454043.

Organisation of this chapter

This chapter contains the proofs of the results stated in Section 1.4 on the approximability of the partition function of the Ising model on bounded degree graphs, for non-real edge interactions. This chapter is organised as follows. First, in Section 3.1 we provide a full outline of our proofs so as to make it easier for the reader to follow this chapter. In Section 3.2 we introduce the preliminary material needed in our proofs. In Section 3.3 we prove Theorem 1.5, which gives our zero-free region for the partition function of the Ising model, and Corollary 1.6 on easiness of approximation within this region. In Section 3.4 we prove Theorem 1.7 on inapproximability of the partition function for most non-real edge interactions. In Section 3.5 we give explicit evidence that zeros imply hardness of approximation and use these results to find more edge interactions where the approximation problem is #P-hard. Finally, in Section 3.6 we generalise the implementation results of [15] so that they can be applied to other two spin systems, including the Ising model. This section is independent of the rest of this chapter, and the result presented may have applications outside the scope of the Ising model.

3.1 Proof outline

In the proof of Theorem 1.5 we use the SAW tree construction of Godsil and Weitz [54, 120] to reduce the study of zero-free regions of partition functions on graphs to the study of zero-free regions of partition functions on trees (see Section 3.2.1 for details). The partition function of a two-spin system on a tree admits a recurrence expression that can be studied to find zero-free regions on trees. This approach has been successfully applied in the literature for the Ising

model and other partition functions [86, 13, 16]. We remark here that in this chapter we use β instead of y (as we did in Chapter 2) to denote the edge interaction, following standard notation in the Ising model literature. In our work we exploit the properties of the Mobius function $h_\beta(z) = (\beta z + 1)/(\beta + z)$ appearing in this recurrence for the Ising model with edge interaction β. This Mobius function satisfies the equality

$$\frac{h_\beta(z) - 1}{h_\beta(z) + 1} = \frac{(\beta - 1)(z - 1)}{(\beta + 1)(z + 1)}, \tag{3.1}$$

which neatly relates properties of $(\beta - 1)/(\beta + 1)$ to properties of the partition function of the Ising model on trees, and greatly simplifies the derivation of our zero-free region. The translation of Theorem 1.5 to an FPTAS for the partition function then follows from the work of Barvinok, Patel and Regts [9, 101], see the proof of Corollary 1.6, via an approximation algorithm that computes the first $O(\log \text{size}(G))$ coefficients of the Taylor series of $\log Z_{\text{Ising}}(G; \beta)$.

In order to obtain our inapproximability results, we construct graphs H with maximum degree at most Δ and two distinguished vertices s, t with degree 1 such that substituting an edge in the host graph with (H, s, t) has the effect of altering the edge interaction β of the original edge to a new edge interaction β'. In this case, we say that H (β, Δ)-implements β', see Section 3.2.3 of the preliminaries for a formal definition. The fact that the terminals s and t have degree 1 will be important to preserve the maximum degree of graphs in our constructions; in fact, this bounded degree restriction made the gadgets developed in Chapter 2 unapplicable in the current chapter, as their maximum degree depends on size(G). As explained in Chapter 2, implementations have played an important role in proofs of hardness of evaluating and approximating partition functions, and they are the main tool to reduce exact computation to approximate computation via a binary search [15, 59]. Initiated in [73, 121], these constructions have now become more elaborate in recent inapproximability results [59, 15, 26], see also Chapter 2, using connections to the iteration of complex dynamical systems. The following definition captures the relevant framework for our implementations. We remark first, that in this chapter we also work with algebraic numbers and computations are performed following the computational model described in Section 2.2.3 of Chapter 2. In this chapter we denote by $\mathbb{A}$ the set of real algebraic numbers and we denote by $\mathbb{C}_\mathbb{A}$ the set of complex algebraic numbers.

Definition 3.1. *Let $\Delta \geq 3$ be an integer and $\beta \in \mathbb{C}_\mathbb{A}$. We say that the pair (Δ, β) implements the complex plane (resp. the real line) in polynomial time for the Ising model if there is an algorithm such that, on input $\lambda \in \mathbb{C}_\mathbb{A}$ (resp. $\lambda \in \mathbb{A}$) and rational $\varepsilon > 0$, computes a graph G that (Δ, β)-implements a complex number $\hat{\lambda}$ with $|\lambda - \hat{\lambda}| \leq \varepsilon$. The running time of this algorithm must be polynomial in the size of the representations of λ and ε.*

Our main contribution is that we can (Δ, β) implement the real line (and, in fact, the complex plane), for those pairs (Δ, β) given in the following lemma.

Lemma 3.2. *Let Δ be an integer with $\Delta \geq 3$ and let $\beta \in \mathbb{C}_\mathbb{A} \setminus \mathbb{R}$ with $\beta \notin \{i, -i\}$ and $1/\sqrt{\Delta - 1} < |\beta - 1|/|\beta + 1|$. Then the pair (Δ, β) implements the complex plane in polynomial time for the Ising model.*

The requirement that it be possible to implement the real line is the main bottleneck when reducing exact computation to approximate computation. Even though it is possible to identify some parameter values which enable the implementation of the real line, the complete determination of this set of parameter values which make this possible seems out of reach, see, for instance, [59, 15]. Let us compare Lemma 3.2 to Theorem 2.2 of Chapter 2. We note that in the unbounded degree setting we are able to implement part of the real line for all non-real edge interactions (other than easy/exceptional points of the Potts model), whereas here we require the added hypothesis that the edge interaction β satisfies $1/\sqrt{\Delta-1} < |\beta-1|/|\beta+1|$. This hypothesis arises when restricting the maximum degree of the gadgets in our constructions, as we will see in the rest of this chapter.

The proof of Lemma 3.2 uses connections with complex dynamics (as opposed to the proofs in Chapter 2), following recent developments in the area. The main idea in this line of works is to analyse what can be implemented with trees, which can be done via understanding the properties of the underlying dynamical system. A key difference in the case of the Ising model relevant to previous works is that vertex-style implementations are useless; due to the perfect symmetry of the Ising model nothing interesting can be implemented through that route. Instead, we have to consider more elaborate edge gadgets, cf. Section 3.4.2, and obtain tree-style recursions for them. Surprisingly, we are able to recover the tree-recursions for vertex activities (even though our gadgets are not trees and simulate edge activities instead), albeit with a bit different value of β which yields the square root in Lemma 3.2. We leave as a tantalising open problem how to remove this square root, which seems inherent in our edge-style approach.

The good news is that once this edge-framework of the gadgets is in place, we can adapt suitably the arguments given in [15]. We have in fact generalised these arguments in Section 3.6, so that they are more amenable to be used for other spin systems. A quick summary of the main idea behind Section 3.6 is as follows. We assume that we have access to a recursively-constructed gadget that implements a weight $f(z)$ assuming that we can implement z (for us, this is the gadget given in Section 3.4.2). Then we apply results of complex dynamics to the function f in order to understand which points we can implement by iterating f, which involves studying the neighbourhood of fixed points of f. There are two steps in the constructions. In the first step, we show how to implement approximations of any number near a fixed point of f. In the second step, this implementation result is translated to implementing the complex plane in polynomial time when the fixed point under consideration is repelling. As explained in Section 3.6, it is important in this generalisation that the function f is of the form $g(z^d)$, where g is a Mobius map. The results of [15] are derived for $g(z) = 1/(1+\lambda z)$, where λ is an activity of the independent set polynomial. As noted in Section 3.6 significant extra work is needed to generalise these results to any Mobius map g.

To conclude this section, we comment on the connection between zeros of the partition function and hardness. It turns out that our hardness and implementation results can be applied to conclude hardness of approximation at some zeros of the partition function, such as the zero β_0 plotted in Figure 1.2, which satisfies $1/(\Delta-1) < |\beta_0-1|/|\beta_0+1| < 1/\sqrt{\Delta-1}$. Our main

result on this matter is the following lemma.

Lemma 3.3. *Let Δ be an integer with $\Delta \geq 3$. Let $\beta \in \mathbb{C}_{\mathbb{A}} \setminus (\mathbb{R} \cup \{i, -i\})$. Let us assume that (Δ, β) implements the edge interaction -1. Then* IsingNorm$(\Delta, \beta, 1.01)$ *and* IsingArg$(\Delta, \beta, \pi/3)$ *are* #P*-hard.*

Typically, if we have a graph G with maximum degree Δ such that $Z_{\text{Ising}}(G; \beta) = 0$, then this can be used to (Δ, β)-implement -1 (provided that we can make the terminals have degree 1) and conclude hardness. See Lemma 3.43, where we conclude hardness of approximation based on Lemma 3.3 and appropriate graphs with zero partition function. This is the first result of this style for the Ising model, though building a connection between zeros and inapproximability for bounded-degree graphs has also been explored thoroughly in a recent work [37] for the independence polynomial. These observations lead us to propose the following conjecture.

Conjecture 3.4. *Let Δ be an integer with $\Delta \geq 3$ and let $\beta \in \mathbb{C}_{\mathbb{A}}$ with $\beta \notin \mathbb{R} \cup \{i, -i\}$. If there is a graph G with maximum degree at most Δ such that $Z_{\text{Ising}}(G; \beta) = 0$, then the problems* IsingNorm$(\beta, \Delta, 1.01)$ *and* IsingArg$(\beta, \Delta, \pi/3)$ *are* #P*-hard.*

We make progress toward Conjecture 3.4 in Corollary 3.45, where we have to weaken the result, concluding hardness of IsingNorm$(\beta, \Delta, 1.01)$ and IsingArg$(\beta, \Delta, \pi/3)$ when the graph G has maximum degree at most $\Delta - 1$. Unfortunately, our implementation results seem not enough to prove the full conjecture.

3.2 Preliminaries

3.2.1 The tree of self-avoiding walks

In this section we recall some results concerning the self-avoiding walk tree (SAW tree) of a graph and its connection to the partition function of the Ising model. SAW trees were introduced in the study of partition functions by Godsil in [54] to study the matching polynomial. SAW trees gained in popularity after the work of Weitz on the independent set polynomial [120]. The idea of Godsil and Weitz was reducing the study of the partition function of a two-spin system on graphs to the study of the same partition function on trees. This idea is at the core of our proof of Theorem 1.5.

Intuitively the SAW tree T of a graph $G = (V, E)$ and a vertex $v \in V$ is constructed by considering all of the self-avoiding walks from v in G and storing these in a tree T. The root of T is the walk consisting of the single vertex v, and two self-avoiding walks are connected in T if one of them is a strict sub-walk of the other with maximal length. We refer to [86, Appendix A] for a formal construction. Some of the leaves of the tree T are pinned according to a systematic procedure that is described in [86, Appendix A]; a description of this systematic procedure is not needed for our proofs, hence we omit it here. It is important to note that if G has maximum degree Δ, then every node of T has at most $d := \Delta - 1$ children, except possibly the root of

T, which might have Δ children. We will use the following result that relates $Z_{\text{Ising}}(G;x)$ and $Z_{\text{Ising}}(T;x)$.

Proposition 3.5 ([86, Proposition B.1])**.** *Let G be a connected graph and let v be a vertex of G. Let T be the SAW tree of (G,v). Then the polynomial $Z_{\text{Ising}}(G;x)$ divides the polynomial $Z_{\text{Ising}}(T;x)$. In particular, if $\beta \in \mathbb{C}$ is such that $Z_{\text{Ising}}(T;\beta) \neq 0$, then it also holds that $Z_{\text{Ising}}(G;\beta) \neq 0$.*

As a consequence of Proposition 3.5, we can translate zero-free results for trees to zero-free results for graphs. Our proof of Theorem 1.5 uses this approach. In the rest of this section we recall some tools to study the partition function of the Ising model of trees.

Definition 3.6. *Let T be a tree (possibly with some pinned leaves) and let v be its root. For each $j \in \{0,1\}$, we define $Z_v^j(T;\beta)$ as the sum of $\beta^{m(\sigma)}$ over the configurations σ of T that have $\sigma(v) = j$, so $Z_{\text{Ising}}(T;\beta) = Z_v^0(T;\beta) + Z_v^1(T;\beta)$. We define the ratio*

$$R(T,v;\beta) = \frac{Z_v^1(T;\beta)}{Z_v^0(T;\beta)}.$$

The ratio $R(T,v;\beta)$ is a rational function on β. If $Z_v^0(T;\beta) \neq 0$, we note that $Z_{\text{Ising}}(T;\beta) = 0$ if and only if $R(T,v;\beta) = -1$, so we can study the zeros of the partition function by studying these ratios. It turns out that the ratios $R(T,v,\beta)$ can be computed recursively. Let us consider the function

$$F_{\beta,k}(z_1,\dots,z_k) = \prod_{j=1}^{k} h_\beta(z_j),$$

where $h_\beta(z) = (\beta z+1)/(\beta+z)$ for any $z \in \mathbb{C}$. Then if $(T_1,v_1),\dots,(T_d,v_k)$ are the trees with roots v_j hanging from the root of T, one can check that

$$R(T,v;\beta) = F_{\beta,k}(r_1,\dots,r_k), \tag{3.2}$$

where $r_j = R(T_j,v_j;\beta)$ for all $j \in [k] := \{1,\dots,k\}$, see for instance [86][1].

3.2.2 Computing with algebraic numbers

Our algorithmic and hardness results (Corollary 1.6 and Theorem 1.7) involve algebraic edge interactions. We refer to Section 2.2.3 of Chapter 2 for an explanation of how we represent and compute with algebraic numbers. Here we need the folowing operation that we did not use in Chapter 2. For $x \in \mathbb{C}$ and $r > 0$ real we denote $B(x,r) = \{z \in \mathbb{C} : |z-x| < r\}$, $\overline{B}(x,r) = \{z \in \mathbb{C} : |z-x| \leq r\}$ and $C(x,r) = \{z \in \mathbb{C} : |z-x| = r\}$. In some of our algorithms we have to check, for $z,x \in \mathbb{C}_{\mathbb{A}}$ and $r \in \mathbb{A}$ with $r > 0$, if $z \in B(z,x)$ or $z \in \overline{B}(z,x)$. Note that $|z-x|$ is a real algebraic number, and thus, we can check if $|z-x| < r$ or $|z-x| = r$ in polynomial time in the sizes of x, z and r.

[1] In [86] the authors work with $Z_G(b) = Z_{\text{Ising}}(G;1/b)b^{|E(G)|}$, so they get $h_b(z) = (b+z)/(bz+1)$ instead.

3.2.3 Implementing weights, series compositions and parallel compositions

Here we define the concept of implementations and series and parallel compositions for the Ising model. These concepts have been used several times to obtain hardness results for partition functions, see Section 2.2.2 for definition for the Tutte polynomial or [40] for definitions for the graph homomorphism partition function. In this chapter we restrict ourselves to the partition function of the Ising model so that the notation is more straightforward to use, hence the notation differs from that of Chapter 2.

We will make use of the following notation. Let $H = (V, E)$ be a graph and let s and t be two distinct vertices of H. For $j, k \in \{0, 1\}$ we define

$$Z_{st}^{jk}(H;\beta) = \sum_{\substack{\sigma\colon V\to\{0,1\}\\ \sigma(s)=j,\,\sigma(t)=k}} \beta^{m(\sigma)}.$$

The *interaction matrix of* H at (s, t) is the matrix

$$I_{st}(H;\beta) = \begin{bmatrix} Z_{st}^{00}(H;\beta) & Z_{st}^{01}(H;\beta) \\ Z_{st}^{10}(H;\beta) & Z_{st}^{11}(H;\beta) \end{bmatrix}. \tag{3.3}$$

We say that the graph H *β-implements* the weight w if there are vertices s and t in H such that the interaction matrix $I_{st}(H;\beta)$ is of the form

$$C \begin{bmatrix} w & 1 \\ 1 & w \end{bmatrix}$$

for some complex number C with $C \neq 0$ or, equivalently, $Z_{st}^{01}(H;\beta) \neq 0$ and we have $Z_{st}^{11}(H;\beta)/Z_{st}^{01}(H;\beta) = w$. One can check that this is equivalent to the definition of implementation given in Section 2.2.2 when $q = 2$. We recall here that the point of implementations is that if we substitute an edge e with weight w of a graph G by the graph H (identifying the endings of e with the vertices s and t), the value of the partition function stays the same up to the factor $C = Z_{st}^{01}(H;\beta)$, see Lemma 2.6 in Chapter 2 for an accurate statement. Hence, if we have an oracle to evaluate the partition function of the Ising model at β and we know C, we can use this oracle to evaluate this partition function at w. This idea is exploited in many hardness reductions, see Chapter 2 and the references therein. In this chapter we are interested in graphs with bounded degree, so in order to use this construction while maintaining the maximum degree of the graphs involved, the vertices s and t should have degree 1 in H. This is formalised in the following definition.

Definition 3.7. *Let $\Delta \geq 2$ be an integer and $\beta \in \mathbb{C} \setminus \{0\}$. Let G be a graph. We say that G (Δ, β)-implements the edge interaction $\beta' \in \mathbb{C}$ if G has maximum degree at most Δ and distinct vertices s and t of degree 1 such that G β-implements β' with the terminals s and t. We say that (Δ, β) implements the edge interaction $\beta' \in \mathbb{C}$ if there is a graph G that (Δ, β)-implements β'. More generally, we say that (Δ, β)-implements a set of edge interactions $S \subseteq \mathbb{C}$ if (Δ, β) implements β' for any $\beta' \in S$.*

It is important to know that this bounded-degree of implementations also presents the transitivity property, that is, if H (Δ, β)-implements the weight w and J (Δ, w)-implements the weight γ, it is not difficult to construct a graph that (Δ, β)-implements γ, which follows from Lemma 2.6 in Section 2.2.2.

We now reintroduce the concepts of parallel composition and series composition following our Ising notation, which allows us to express the implemented weights in terms of the interaction matrix defined in (3.3). This will simplify some of the calculations in this chapter. Let $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ be two graphs. For each $j \in \{1, 2\}$, let $s_j, t_j \in V_j$ be two distinct vertices.

1. Recall that the *parallel composition* of (H_1, s_1, t_1) and (H_2, s_2, t_2) is the graph H constructed by considering the union of H_1 and H_2 and identifying s_1 with s_2 and t_1 with t_2. One can easily check that the interaction matrix $I_{s_1t_1}(H; y)$ is the Hadamard product (or component-wise product) of the interaction matrices $I_{s_1t_1}(H_1; y)$ and $I_{s_2t_2}(H_2; y)$. Hence, if (H_j, s_j, t_j) implements w_j for $j \in \{1, 2\}$, then (H, s_1, t_1) implements $w = w_1w_2$.

2. Recall that the *series composition* of (H_1, s_1, t_1) and (H_2, s_2, t_2) is the graph H constructed by considering the union of H_1 and H_2 and identifying t_1 with s_2. One can easily check that the interaction matrix $I_{s_1t_2}(H; y)$ is the product of the interaction matrices $I_{s_1t_1}(H_1; y)$ and $I_{s_2t_2}(H_2; y)$. Hence, if (H_j, s_j, t_j) implements w_j for $j \in \{1, 2\}$, (H, s_1, t_1) implements the edge interaction $w = (w_1w_2 + 1)/(w_1 + w_2)$. Note that this operation is commutative, the series composition of (H_1, s_1, t_1) and (H_2, s_2, t_2) implements the same weight as the series composition of (H_2, s_2, t_2) and (H_1, s_1, t_1).

Series compositions are particularly helpful when working with graphs with bounded degree. In our constructions we usually consider the series composition of a graph H that β-implements a weight w and a path of length 1 with edge interaction β. This allows us to have a terminal vertex with degree 1 in the resulting graph. This construction implements the edge interaction

$$h_\beta(w) = \frac{\beta w + 1}{\beta + w}. \tag{3.4}$$

The Mobius map h_β arises very frequently in this work and plays an important role in our arguments, as we highlighted in the proof outline.

3.2.4 Iteration of complex rational maps

In Section 3.6 we extend the work on implementations for the independent set polynomial given in [15] to a more general setting so that these results can be applied to other partition functions, such as the partition function of the Ising model. The technique developed in [15] uses several results from complex dynamics that we recall here. These complex dynamics results are also used in this section when implementing edge interactions for the Ising model. We gather all this material in this section. We refer to [100] for an introduction to Riemann surfaces and to [11, 91] for an introduction to complex dynamics.

By $\widehat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ we denote the Riemann sphere. The Riemann sphere is a metric space with the *chordal metric* $d(\cdot, \cdot)$, given by

$$d(z,w) = \frac{2|z-w|}{\left(1+|z|^2\right)^{1/2}\left(1+|w|^2\right)^{1/2}}, \quad \text{and} \quad d(z,\infty) = \lim_{w\to\infty} d(z,w) = \frac{2}{\left(1+|z|^2\right)^{1/2}}.$$

The Riemann sphere is a Riemann surface, meaning that locally the Riemann sphere is homeomorphic to open subsets of $\mathbb{C}$. One can translate several results from complex analysis to Riemann surfaces. An example of such a result is the *open mapping theorem*, see, for example, [100, Theorem 2.2.2].

Proposition 3.8 (Open mapping theorem for Riemann surfaces, [100, Theorem 2.2.2])**.** *Let X and Y be Riemann surfaces. If $\phi\colon X \to Y$ is a non-constant holomorphic mapping, then ϕ is open, that is, $\phi(O)$ is an open subset of Y for any open set $O \subseteq X$.*

One can show that the set of holomorphic functions on the Riemann sphere is exactly the set of rational functions. A rational function of degree d is a d-fold map on $\widehat{\mathbb{C}}$. Hence, the automorphisms on the Riemann sphere are precisely the rational functions of degree 1. These are also known as *Mobius maps* or *Mobius transformations*. We use the following two properties of Mobius maps.

Proposition 3.9 ([100, Theorem 5.7.3, part (f)])**.** *If C is a circle in $\widehat{\mathbb{C}}$ (i.e., C is a circle in $\mathbb{C}$ or $C = L \cup \{\infty\}$ for some line L in $\mathbb{C}$), then the image of C under any Mobius map is also a circle in $\widehat{\mathbb{C}}$.*

Proposition 3.10 ([100, Proof of Theorem 5.8.2])**.** *Let $a \in \mathbb{C}$ with $|a| < 1$, $\theta \in \mathbb{R}$ and let $\phi(z) = e^{i\theta}(az+1)/(\overline{a}+z)$. Then the Mobius map ϕ fixes the circle $C(0,1)$.*

It is well-known that holomorphic complex maps are locally Lipschitz and this is exploited in [15]. Here we use a global Lipschitz property on the Riemann sphere, see Lemma 3.11.

Lemma 3.11 ([11, Theorem 2.3.1])**.** *Let f be a rational map. Then f is a Lipschitz map on the Riemann sphere, that is, there is a constant $L > 0$ such that $d(f(z), f(w)) \leq L d(z,w)$ for every $z, w \in \hat{\mathbb{C}}$, where d is the chordal metric.*

We conclude this section by introducing some results from complex dynamics. For a non-negative integer n we denote by f^n the n-fold iterate of n (for $n = 0$, f^0 denotes the identity map). Let $f\colon \widehat{\mathbb{C}} \to \widehat{\mathbb{C}}$ be a rational map. Suppose that $\omega \in \widehat{\mathbb{C}}$ is a fixed point of f. If $\omega \in \mathbb{C}$, the *multiplier* of f at ω is defined as $f'(\omega)$. If $\omega = \infty$, the *multiplier* of f at ω is defined as $1/f'(\infty)$. The behaviour of the iterates f^n near a fixpoint is characterised in terms of the multiplier q of f at this fixpoint. With this in mind, there are three types of fixpoints: *attracting* if $|q| < 1$, *neutral* or *indifferent* if $|q| = 1$, and *repelling* if $|q| > 1$. We also need to introduce the *Julia* set of f. We refer to [11] for a definition, here we only use the two following properties of Julia sets, Lemma 3.12 and Theorem 3.13.

Lemma 3.12 ([91, Lemma 4.6])**.** *Let $f\colon \widehat{\mathbb{C}} \to \widehat{\mathbb{C}}$ be a rational map. Every repelling fixpoint of f belongs to the Julia set of f.*

A set U is a neighbourhood of x if it contains a ball $B(x,r)$ for some $r > 0$. The exceptional set of a rational map f is the set of points $z \in \widehat{\mathbb{C}}$ such that $[z] = \{z' \in \widehat{\mathbb{C}} : f^n(z') = f^m(z)$ for some integers $n, m \geq 0\}$ is finite.

Theorem 3.13 ([11, Theorem 4.2.5])**.** *Let $f\colon \widehat{\mathbb{C}} \to \widehat{\mathbb{C}}$ be a rational map with exceptional set E_f. Let z_0 be a point in the Julia set of f and let U be a neighbourhood of z_0. Then $\bigcup_{n=0}^{\infty} f^n(U) = \widehat{\mathbb{C}} \setminus E_f$.*

The exceptional points of f can be characterised as follows.

Lemma 3.14 ([91, Lemma 4.9] and [11, Theorem 4.1.2])**.** *Let $f\colon \widehat{\mathbb{C}} \to \widehat{\mathbb{C}}$ be a complex rational map of degree at least 2, and let E_f be its exceptional set. Then, $|E_f| \leq 2$. Moreover,*

- *if $E_f = \{\zeta\}$, then ζ is a fixed point of f with multiplier 0;*
- *if $E_f = \{\zeta_1, \zeta_2\}$ where $\zeta_1 \neq \zeta_2$, then ζ_1, ζ_2 have multiplier 0 and either they are fixed points of f, or $f(\zeta_1) = \zeta_2$ and $f(\zeta_2) = \zeta_1$.*

3.3 Easiness: a zero-free region for the Ising model

In this section we prove Theorem 1.5 and Corollary 1.6. We also compare Theorem 1.5 to the zero-free regions appearing in Figure 1.1.

3.3.1 Proof of Theorem 1.5

First, we introduce some notation that will be used repeatedly in this work.

Definition 3.15. *Let $\delta > 0$. We define $\mathcal{R}(\delta)$ as the set of complex numbers z such that $|(z-1)/(z+1)| \leq \delta$.*

Definition 3.15 allows us to conveniently restate Theorem 1.5 as $Z_{\text{Ising}}(G;\beta) \neq 0$ for any graph G with maximum degree at most Δ and any $\beta \in \mathcal{R}(\varepsilon_\Delta)$, where $\varepsilon_\Delta = \tan(\pi/(4\Delta - 4))$. Proposition 3.16 gives some properties of the region $\mathcal{R}(\delta)$ that we need in our proofs. See Section 3.2.2 for a definition of $B(x,r)$, $\overline{B}(x,r)$ and $C(x,r)$.

Proposition 3.16. *Let $\delta > 0$. The region $\mathcal{R}(\delta)$ satisfies the following properties:*

1. *We have $1 \in \mathcal{R}(\delta)$ and $-1 \notin \mathcal{R}(\delta)$.*
2. *If $\beta \in \mathcal{R}(\delta)$, then $\beta^{-1} \in \mathcal{R}(\delta)$.*
3. *The map $\phi(z) = (z-1)/(z+1)$ has the following property. We have $\phi(C(0,1)) = i\mathbb{R} = \phi^{-1}(C(0,1))$, and $\{z \in \mathbb{C} : \mathrm{Re}(z) > 0\} = \phi^{-1}(B(0,1))$. In particular, if $\delta = 1$, then $\mathcal{R}(\delta)$ is the set of complex numbers z with $\mathrm{Re}(z) \geq 0$.*

4. *If $\delta \in (0,1)$, then $\mathcal{R}(\delta)$ is the closed disk $\overline{B}(c_\delta, r_\delta)$ with centre $c_\delta = (1+\delta^2)/(1-\delta^2)$ and radius $r_\delta = 2\delta/(1-\delta^2)$. Moreover, in this case for every $z \in \mathcal{R}(\delta)$ we have $|z| \leq c_\delta + r_\delta = (1+\delta)/(1-\delta)$.*

Proof. We prove each property separately.

1. This property is trivial.

2. Note that $(z^{-1}-1)/(z^{-1}+1) = (1-z)/(1+z) = -(z-1)/(1+z)$, so $\beta \in \mathcal{R}(\delta)$ if and only if $\beta^{-1} \in \mathcal{R}(\delta)$.

3. One can check that the inverse of $\phi(z) = (z-1)/(z+1)$ is the Mobius map $\phi^{-1}(y) = -(1+y)/(y-1)$. Hence, $|\phi(z)| = 1$ if and only if $|\phi^{-1}(z)| = 1$, which happens exactly when $|z+1| = |z-1|$ or, equivalently, $z \in i\mathbb{R}$. This proves $\phi(C(0,1)) = i\mathbb{R} = \phi^{-1}(C(0,1))$. Note that $|z-1| < |z+1|$ if and only if $|\mathrm{Re}(z)-1| < |\mathrm{Re}(z)+1|$. The latter is equivalent to $\mathrm{Re}(z) > 0$. This shows that $\{z \in \mathbb{C} : \mathrm{Re}(z) > 0\} = \phi^{-1}(B(0,1))$, and the result follows.

4. We note that $\mathcal{R}(\delta) = \{z \in \mathbb{C} : |\phi(z)| \leq \delta\} = \phi^{-1}(\overline{B}(0,\delta))$. We claim that ϕ^{-1} sends the circle $C(0,\delta)$ to the circle $C(c_\delta, r_\delta)$, where c_δ and r_δ are as in the statement. As ϕ^{-1} is a Mobius map, $\phi^{-1}(C(0,\delta))$ is a circle or a line of $\mathbb{C}$, see Section 3.2.4 on rational maps. We take 3 points in the circle $C(0,\delta)$ and show that they are in $C(c_\delta, r_\delta)$. The three points are $\delta, -\delta$ and δi. One can easily check that $\phi^{-1}(\delta) = (1+\delta)/(1-\delta) = c_\delta + r_\delta$ and $\phi^{-1}(\delta) = (1-\delta)/(1+\delta) = c_\delta - r_\delta$. We also have

$$\phi^{-1}(\delta i) - c_\delta = \frac{i-\delta}{i+\delta} - \frac{1+\delta^2}{1-\delta^2} = -\frac{1+i\delta}{i+\delta}\frac{2\delta}{1-\delta^2},$$

so $|\phi^{-1}(\delta i) - c_\delta| = r_\delta$ as we wanted. We conclude that $\phi^{-1}(C(0,\delta)) = C(c_\delta, r_\delta)$. Since ϕ^{-1} is holomorphic in $B(0,1)$ and $\phi^{-1}(0) = 1 \in \overline{B}(c_\delta, r_\delta)$, we obtain $\phi^{-1}(\overline{B}(0,\delta)) = \overline{B}(c_\delta, r_\delta)$ as we wanted. Finally, the point in $\overline{B}(c_\delta, r_\delta)$ with the largest norm is $c_\delta + r_\delta = (1+\delta)/(1-\delta)$. □

Remark 3.17. *Let $\alpha, \beta \in [-\pi, \pi]$. Then it is well-known that if $\alpha, \beta, \alpha+\beta \notin \pi/2 + \pi\mathbb{Z}$, we have*

$$\tan(\alpha+\beta) = \frac{\tan(\alpha)+\tan(\beta)}{1-\tan(\alpha)\tan(\beta)}.$$

In particular, we obtain the equality

$$\tan(2\alpha) = \frac{2\tan(\alpha)}{1-\tan(\alpha)^2}.$$

Let $f(x) = 2x/(1-x^2)$. This function is strictly increasing in $x \in [-1,1]$. Hence, if we have $\alpha \in [-\pi/4, \pi/4]$ and $\tan(2\alpha) = f(\delta)$ for some $\delta \in (0,1)$, we can conclude that $\tan(\alpha) = \delta$. This argument will be used in the proof of Theorem 1.5.

Lemma 3.18. *Let $\delta, \varepsilon > 0$, let $\beta \in \mathcal{R}(\delta)$ and let $h_\beta(z) = (\beta z + 1)/(\beta + z)$. Then $h_\beta(\mathcal{R}(\varepsilon)) \subseteq \mathcal{R}(\delta\varepsilon)$. Moreover, we have $h_\beta(\infty) = \beta \in \mathcal{R}(\delta)$.*

Proof. It is straightforward to check that for any $z \in \mathbb{C}$ with $z \neq -1$, we have Equation (3.1), namely

$$\frac{h_\beta(z)-1}{h_\beta(z)+1} = \frac{(\beta-1)(z-1)}{(\beta+1)(z+1)}.$$

The result now follows from (3.1) and the definition of $\mathcal{R}(\varepsilon), \mathcal{R}(\delta\varepsilon)$. □

We are now ready to prove Theorem 1.5.

Theorem 1.5. *Let Δ be an integer with $\Delta \geq 3$. Let $G = (V, E)$ be a graph of maximum degree at most Δ. Let $\varepsilon_\Delta = \tan(\pi/(4(\Delta-1))) \in (0,1)$. Then $Z_{\text{Ising}}(G;\beta) \neq 0$ for all $\beta \in \mathbb{C}$ with $|\beta-1|/|\beta+1| \leq \varepsilon_\Delta$.*

Proof. Let $\beta \in \mathcal{R}(\varepsilon_\Delta)$. In light of Proposition 3.5, we only have to prove that $Z_{\text{Ising}}(T;\beta) \neq 0$ for all trees T with maximum degree at most Δ with possibly some pinned leaves. Let v be the root of such a tree T. We are going to prove that $R(T,v;\beta) \neq -1$ and (unless T consists of a single vertex, pinned to 1, in which case the Theorem is trivial) that $Z^0_v(T,v;\beta) \neq 0$. Note that both assertions combined imply that

$$Z_{\text{Ising}}(T;\beta) = Z^0_v(T,v;\beta)\left(1+\frac{Z^1_v(T,v;\beta)}{Z^0_v(T,v;\beta)}\right) = Z^0_v(T,v;\beta)\,(1+R(T,v;\beta)) \neq 0$$

as we want.

First, we restrict ourselves to trees such that every node has at most $d := \Delta - 1$ children and possibly some its leaves are pinned. We claim that for such a tree T with root v we have

1. $R(T,v;\beta) \in \mathcal{R}(1) \cup \{\infty\}$, that is, $R(T,v;\beta)$ has non-negative real part or $R(T,v;\beta) = \infty$ (Proposition 3.16);
2. if T has height at least 1, then $Z^0_v(T,v;\beta) \neq 0$ (a tree with only one vertex has height 0 by definition).

We carry out the proof by induction on the height of the tree. Let us consider the case when the tree T consists of only one vertex. Depending on whether the vertex is pinned or not, either $R(T,v;\beta) = 1$ and $Z^0_v(T;\beta) = 1$, or $R(T,v;\beta) \in \{0,\infty\}$ and $Z^0_v(T;\beta) \in \{1,0\}$. In either case, $R(T,v;\beta) \in \mathcal{R}(1) \cup \{\infty\}$.

Now let T be a tree of height $l > 0$ and let us assume that our claim holds for any of the desired trees with height at most $l-1$. Let T be a tree of height l such that all nodes have at most d children. Let v be its root and let $(T_1, v_1), \ldots, (T_k, v_k)$ be the trees hanging from this root. By assumption, $k \leq d$. Let $r_j = R(T_j, v_j; \beta)$ for all $j \in [k]$. In view of (3.2), we have

$$R(T,v;\beta) = \prod_{j=1}^{k} h_\beta(r_j). \tag{3.5}$$

By our induction hypothesis, $r_j \in \mathcal{R}(1) \cup \{\infty\}$ for all $j \in [k]$. In light of Lemma 3.18, we find that $h_\beta(r_j) \in \mathcal{R}(\varepsilon_\Delta)$ for all $j \in [k]$. This property will be enough to ensure that the product

in (3.5) yields a complex number with non-negative real part. Let us study the argument of any element of $\mathcal{R}(\varepsilon_\Delta)$. By a trigonometry reasoning shown in Figure 3.1, the argument of any element in $\mathcal{R}(\varepsilon_\Delta)$ is in the interval $[-\theta, \theta]$ for θ such that

$$\tan(\theta) = \frac{r_{\varepsilon_\Delta}}{\sqrt{c_{\varepsilon_\Delta}^2 - r_{\varepsilon_\Delta}^2}} = \frac{2\varepsilon_\Delta}{1-\varepsilon_\Delta^2},$$

where r_δ and c_δ are defined in Proposition 3.16. In view of Remark 3.17 for $\alpha = \theta/2$, we conclude that $\tan(\theta/2) = \varepsilon_\Delta$ and, thus, $\theta/2 = \arctan(\varepsilon_\Delta) = \pi/(4d)$. Therefore, the complex number $R(T, v; \beta)$ is the product of k numbers with argument in $[-\theta, \theta] = [-\pi/(2d), \pi/(2d)]$, so its argument is in $[-k\theta, k\theta] \subseteq [-\pi/2, \pi/2]$, where we used $k \leq d$. This is equivalent to saying that $R(T, v; \beta)$ has non-negative real part as we wanted. Note that when $l \geq 1$, we have also shown that $R(T, v; \beta) \in \mathbb{C}$.

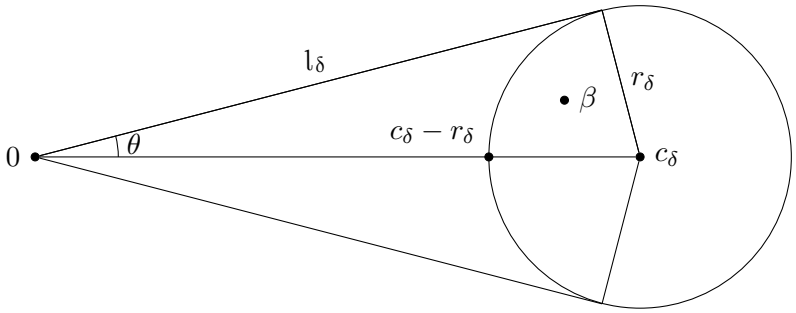


Figure 3.1: The disk $\mathcal{R}(\delta)$.

Let us now prove that $Z_v^0(T, v; \beta) \neq 0$. We have

$$Z_v^0(T; \beta) = \prod_{j=1}^{k} \left(\beta Z_{v_j}^0(T_j; \beta) + Z_{v_j}^1(T_j; \beta) \right). \tag{3.6}$$

If T_j has height at least 1, then

$$\beta Z_{v_j}^0(T_j; \beta) + Z_{v_j}^1(T_j; \beta) = Z_{v_j}^0(T_j; \beta)\left(\beta + R(T_j, v_j; \beta)\right) \neq 0,$$

where we used that $Z_{v_j}^0(T_j; \beta) \neq 0$ and $\mathrm{Re}(\beta + R(T_j, v_j; \beta)) > 0$ by the induction hypothesis (recall that $\mathrm{Re}(\beta) > 0$). If T_j has height 0, that is, T_j has only one vertex, then, depending on whether this vertex is pinned or not,

$$\beta Z_{v_j}^0(T_j; \beta) + Z_{v_j}^1(T_j; \beta) \in \{1, \beta, 1+\beta\}.$$

Therefore, the product in (3.6) is a product of complex numbers that are non-zero, so $Z_v^0(T; \beta) \neq 0$ as we wanted.

Finally, to prove the Theorem, we consider a tree T with maximum degree at most Δ and possibly some pinned leaves. Let v be its root and let $(T_1, v_1), \ldots, (T_k, v_k)$ be the trees hanging from this root. By the claim, $R(T_j, v_j; \beta) \in \mathcal{R}(1) \cup \{\infty\}$ for all $j \in [k]$. By Lemma 3.18,

$h_\beta(r_j) \in \mathcal{R}(\varepsilon_\Delta)$ for all $j \in [k]$, so the argument of $h_\beta(r_j)$ is in $[-\pi/(2d), \pi/(2d)]$ for all $j \in [k]$. It follows from this fact, $k \leq \Delta$, and (3.5), that the argument of $R(T, v; \beta)$ is in

$$[-k\pi/(2d), k\pi/(2d)] \subseteq [-\Delta\pi/(2d), \Delta\pi/(2d)] \subseteq [-3\pi/4, 3\pi/4],$$

where we used that $\Delta \geq 3$. In particular, $R(T, v; \beta)$ is not a negative real number, so $R(T, v; \beta) \neq -1$. The fact that $Z_v^0(T; \beta) \neq 0$ follows analogously from (3.6), $Z_v^0(T; \beta)$ is a product of non-zero complex numbers. □

Corollary 1.6. *Let Δ be an integer with $\Delta \geq 3$. Let β be an algebraic number such that $|\beta - 1|/|\beta + 1| < \varepsilon_\Delta$, where $\varepsilon_\Delta = \tan(\pi/(4(\Delta - 1)))$. Then there is an algorithm that, on inputs a graph G with maximum degree at most Δ and a rational $\varepsilon > 0$, runs in time $\mathrm{poly}(\mathrm{size}(G), 1/\varepsilon)$ and outputs $\hat{Z} = Z_{\mathrm{Ising}}(G; \beta)e^z$ for some complex number z with $|z| \leq \varepsilon$.*

Proof. The proof follows from combining Theorem 1.5, the work of Patel and Regts [101] and the work of Barvinok [8][2]. Let G and $\varepsilon > 0$ be the inputs of our algorithm. We consider the polynomial $q_{G,\beta}(z) = Z_{\mathrm{Ising}}(G; 1+z(\beta-1))$. We want to give an FPTAS for $q_{G;\beta}(1) = Z_{\mathrm{Ising}}(G; \beta)$. We claim that, on graphs with maximum degree at most Δ, we can compute the k-th coefficient of $q_{G,\beta}(z)$ in polynomial time in 2^k and the size of G. This claim is proved for the more general case of the graph homomorphism partition function in the proof of [101, Theorem 6.1]. Recall that 1 and β are in the interior of the disk $\mathcal{R}(\varepsilon_\Delta)$ (Proposition 3.16) so this is also true of an open interval around the line segment between them. Hence, there is $\delta > 0$ such that $1 + z(\beta - 1) \in \mathcal{R}(\varepsilon_\Delta)$ for all $z \in R_\delta$, where R_δ is a strip of the form $R_\delta = \{z \in \mathbb{C} : -\delta \leq \mathrm{Re}(z) \leq 1 + \delta, |\mathrm{Im}(z)| \leq \delta\}$. In light of Theorem 1.5, we conclude that $q_{G,\beta}(z) \neq 0$ for all $z \in R_\delta$. In [8, Section 2.2.2] Barvinok constructs a polynomial ϕ_δ and a real number $b_\delta > 1$ such that $\phi_\delta(0) = 0$, $\phi_\delta(1)$ and $\phi_\delta(z) \in R_\delta$ for any $z \in \overline{B}(0, b_\delta)$. Note that the polynomial $p_{G,\beta}(z) = q_{G,\beta}(\phi_\delta(z))$ does not vanish in $\overline{B}(0, b_\delta)$. Finally, we compute an approximation of $p_{G,\beta}(1) = Z_{\mathrm{Ising}}(G; \beta)$ as in [8, Lemma 2.2.1] using the truncated Taylor series of $\log p_{G,\beta}(z)$. The algorithm of Barvinok uses $O(\log(\deg(p_{G,\beta})/\varepsilon)) = O(\log(\mathrm{size}(G)/\varepsilon))$ coefficients of the Taylor series of $\log p_{G,\beta}(z)$. Here the implicit "O" notation depends only on β. These coefficients can be computed using the algorithm of Patel and Regts in polynomial time in $\mathrm{size}(G)$ and $1/\varepsilon$. We conclude that [8, Lemma 2.2.1] computes Y such that $|\log p_{G,\beta}(1) - Y| \leq \varepsilon$ in polynomial time in $\mathrm{size}(G)$ and $1/\varepsilon$. Let $z = \log p_{G,\beta}(1) - Y$ and $\hat{Z} = \exp(Y)$. Then we have $\hat{Z} = Z_{\mathrm{Ising}}(G; \beta)e^z$ and $|z| \leq \varepsilon$ as we wanted. □

3.3.2 Comparing Theorem 1.5 to the state of the art

In this section we gather all the results we are aware of on the zeros of the partition function of the Ising model and compare them to Theorem 1.5. We show that our result extends the state of the art significantly.

[2]The idea presented in the proof of Corollary 1.6 is known among experts, see for example, [86]; we include it here for completeness. We note that the only properties of $\mathcal{S} = \{z \in \mathbb{C} : |z - 1|/|z + 1| < \varepsilon_\Delta\}$ needed are that $\mathcal{S}$ is open and $\{t + (1 - t)\beta : t \in [0, 1]\} \subseteq \mathcal{S}$ for all $\beta \in \mathcal{S}$.

Results on the zeros of the graph homomorphism partition function can be particularised to the Ising model. Let $G = (V, E)$ be an undirected graph, possibly with multiple edges or loops, and let $A = (a_{ij})$ be a $k \times k$ symmetric matrix of complex numbers. Recall that the *graph homomorphism partition function* is defined as

$$\hom(G; A) = \sum_{\phi\colon V \to [k]} \prod_{\{u,v\} \in E} a_{\phi(u)\phi(v)},$$

where $[k]$ denotes $\{1, \dots, k\}$, see (1.3). When $k = 2$ and $a_{11} = a_{22}$ we have

$$\hom(G; A) = a_{12}^{|E|} \sum_{\phi\colon V \to \{1,2\}} \prod_{\substack{\{u,v\} \in E: \\ \phi(u) = \phi(v)}} \frac{a_{11}}{a_{12}} = a_{12}^{|E|} Z_{\text{Ising}}\left(G; \frac{a_{11}}{a_{12}}\right), \tag{3.7}$$

recovering the partition function of the Ising model as a particular case.

To the best of our knowledge, the best result on the zeros of the graph homomorphism partition function known up to date is the following result of Barvinok.

Theorem 3.19 ([8, Theorem 7.1.4])**.** *For a positive integer Δ, let*

$$\delta_\Delta = \max\left\{\sin\left(\frac{\alpha}{2}\right)\cos\left(\Delta\frac{\alpha}{2}\right) : 0 < \alpha < \frac{2\pi}{3\Delta}\right\}. \tag{3.8}$$

Then for any graph $G = (V, E)$ with maximum degree at most Δ, we have $\hom(G; A) \neq 0$ for any complex symmetric matrix A with dimension $k \times k$ such that $|1 - a_{ij}| \leq \delta_\Delta$ for any $i, j \in \{1, \dots, k\}$.

Theorem 3.19 can be naively translated to the Ising model by considering matrices of the form

$$\begin{bmatrix} \beta & 1 \\ 1 & \beta \end{bmatrix}.$$

For those matrices, Theorem 3.19 says that $Z_{\text{Ising}}(G, \beta) \neq 0$ when $|1 - \beta| \leq \delta_\Delta$. One can obtain a stronger result for the Ising model if we apply (3.7) together with Theorem 3.19.

Corollary 3.20. *Let Δ be a positive integer, let δ_Δ as in (3.8) and let*

$$\beta \in \bigcup_{a \in \overline{B}(1, \delta_\Delta)} \frac{1}{a}\overline{B}(1, \delta_\Delta).$$

Then $Z_{\text{Ising}}(G, \beta) \neq 0$ for any graph G with maximum degree at most Δ.

Proof. We can write $\beta = a_{11}/a_{12}$ for $a_{11}, a_{12} \in B(1, \delta_\Delta)$. We consider the matrix

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{11} \end{bmatrix}.$$

By (3.7) and Theorem 3.19 we have $Z_{\text{Ising}}(G, \beta) = a_{12}^{-|E|}\hom(G; A) \neq 0$ for any graph $G = (V, E)$ with maximum degree at most Δ. □

The case $a = 1$ is the naive application of Theorem 3.19 mentioned after Theorem 3.19. Taking $a = 1/\sqrt{\beta}$ in Corollary 3.20 gives the following corollary that can be found in the work of Mann and Bremner [89].

Corollary 3.21 ([89, Corollary 7])**.** *Let Δ be a positive integer, let δ_Δ as in (3.8) and let $\beta \in \mathbb{C}$ such that $|1 - 1/\sqrt{\beta}| \leq \delta_\Delta$ and $|1 - \sqrt{\beta}| \leq \delta_\Delta$. Then $Z_{\text{Ising}}(G, \beta) \neq 0$ for any graph G with maximum degree at most Δ.*

Proof. This is a particular case of Corollary 3.20 where a is set to $1/\sqrt{\beta}$. □

In Lemma 3.22 we show that the sets $\bigcup_{a \in \overline{B}(1,\delta)} \overline{B}(1, \delta)/a$ and $\mathcal{R}(\delta)$ are related.

Lemma 3.22. *For any $\delta \in (0, 1/2]$, we have*

$$\left[\frac{1-\delta}{1+\delta}, \frac{1+\delta}{1-\delta}\right] \subseteq \bigcup_{a \in \overline{B}(1,\delta)} \frac{1}{a}\overline{B}(1, \delta) \subseteq \mathcal{R}\left(2\delta/\sqrt{3}\right).$$

Proof. The first inclusion follows from the fact that

$$\left[\frac{1-\delta}{1+\delta}, \frac{1+\delta}{1-\delta}\right] \subseteq \frac{1}{1+\delta}\overline{B}(1, \delta) + \frac{1}{1-\delta}\overline{B}(1, \delta).$$

In the rest of the proof we focus on the second inclusion. First, let us consider a of the form $a = 1 + \delta e^{i\theta}$ for some $\theta \in [0, 2\pi)$. We show that $\overline{B}(1, \delta)/a \subseteq \mathcal{R}(2\delta/\sqrt{3})$. Note that $\overline{B}(1, \delta)/a = \overline{B}(1/a, \delta/|a|)$. Since $\overline{B}(1, \delta)/a$ and $\mathcal{R}(2\delta/\sqrt{3})$ are convex, we only have to show that the border of $\overline{B}(1, \delta)/a$ is contained in $\mathcal{R}(2\delta/\sqrt{3})$. Let β be in the border of $\overline{B}(1, \delta)/a$. We can write $\beta = (1 + \delta e^{\tau i})/a = (1 + \delta e^{\tau i})/(1 + \delta e^{\theta i})$ for some $\tau \in [0, 2\pi)$. We have

$$\frac{\beta - 1}{\beta + 1} = \delta \frac{e^{\tau i} - e^{\theta i}}{2 + \delta(e^{\tau i} + e^{\theta i})}. \tag{3.9}$$

The norm of the right hand size of (3.9) is bounded by $2\delta/\sqrt{3}$ when $\delta \in (0, 1/2]$. This can be shown using `Mathematica` (see Section 3.3.3 for the code). We highlight that the fact that $\delta \in (0, 1/2]$ is needed for this bound as the norm of the right hand size of (3.9) is unbounded when δ gets close to 1. We conclude that β is in $\mathcal{R}(2\delta/\sqrt{3})$ as we wanted.

Now we consider the case when $a = 1 + re^{i\theta}$ for some $r \in (0, \delta)$. Let $f(z) = 1/z$ and $\Omega = f(\overline{B}(1, \delta))$. We claim that the set $f(\overline{B}(1, \delta))$ is convex. Let us finish the proof assuming this claim. The map f maps the border of $\overline{B}(1, \delta)$ to the border of $f(\overline{B}(1, \delta))$. Thus, we can write $1/a = \lambda f(a_1) + (1 - \lambda)f(a_2)$ for some $\lambda \in (0, 1)$ and a_1 and a_2 in the circle of centre 1 and radius δ. We obtain,

$$\frac{1}{a}\overline{B}(1, \delta) = \lambda\frac{1}{a_1}\overline{B}(1, \delta) + (1 - \lambda)\frac{1}{a_2}\overline{B}(1, \delta),$$

which is contained in $\mathcal{R}(2\delta/\sqrt{3})$ due to the convexity of $\mathcal{R}(2\delta/\sqrt{3})$ (Proposition 3.16, $\mathcal{R}(2\delta/\sqrt{3})$ is a closed disk) and the fact that $\overline{B}(1, \delta)/a_1$ and $\overline{B}(1, \delta)/a_2$ are contained in $\mathcal{R}(2\delta/\sqrt{3})$ as we argued at the beginning of this proof.

Finally we prove that $f(\overline{B}(1,\delta))$ is a convex set. The map $f(z) = 1/z$ is a Mobius map, so it sends lines and circles to lines and circles, see Section 3.2.4 on rational maps. The points $f(1-\delta), f(1+\delta), f(1+i\delta)$ are not aligned so f sends the circle of center 1 and radius δ to a circle C determined by the points $1/(1-\delta), 1/(1+\delta)$ and $1/(1+i\delta)$. Note that $f(\overline{B}(1,\delta)) \subseteq \mathbb{C}$ as $0 \notin \overline{B}(1,\delta)$. Moreover, $f(1) = 1$ is in the disk determined by the circle C that contains $1/(1-\delta), 1/(1+\delta)$ and $1/(1+i\delta)$, so $f(\overline{B}(1,\delta))$ is precisely the smallest closed disk that contains C and, in particular, convex. □

As a consequence of Lemma 3.22, for δ_Δ is as in (3.8), whenever $\delta_\Delta \leq 1/2$, the non-zero regions of the partition function of the Ising model given by Corollaries 3.20 and 3.21 are contained in $\mathcal{R}((2/\sqrt{3})\delta_\Delta)$. Recall that the zero-free region given in Theorem 1.5 is $\mathcal{R}(\varepsilon_\Delta)$, where $\varepsilon_\Delta = \tan(\pi/(4\Delta-4))$. By the definition of $\mathcal{R}(\delta)$ (see Definition 3.15), we have $\mathcal{R}(2\delta_\Delta/\sqrt{3}) \subseteq \mathcal{R}(\varepsilon_\Delta)$ if and only if $2\delta_\Delta/\sqrt{3} \leq \varepsilon_\Delta$. In the remaining of this section we compare $2\delta/\sqrt{3}\delta_\Delta$ and ε_Δ. Figure 3.2 shows that ε_Δ is significantly larger than $2\delta/\sqrt{3}\delta_\Delta$ and that $\delta_\Delta(\Delta-1) < 1$. Thus, $\delta_\Delta > 1/2$ when $\Delta \geq 3$ and we can apply Lemma 3.22 to conclude that Theorem 1.5 improves the results of Barvinok, Mann and Bremner (Corollaries 3.20 and 3.21) considerably, particularly for the case $\Delta = 3$. See also Figure 1.1. The limit of $\varepsilon_\Delta(\Delta-1)$ is $\pi/4 = 0.785...$, whereas we have numerically checked that $2\delta/\sqrt{3}\delta_\Delta(\Delta-1)$ tends to 0.64789.... Thus, our result is stronger for all Δ, and in the limit as $\Delta \to \infty$.

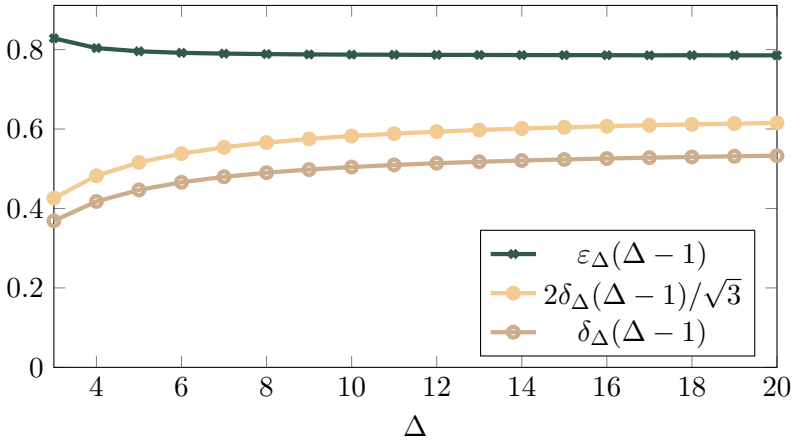


Figure 3.2: Plot of the quantities $\varepsilon_\Delta(\Delta-1)$, and $\delta_\Delta(2/\sqrt{3})(\Delta-1)$.

We recall here for completeness the approximability of $Z_{\text{Ising}}(G;\beta)$ when β is a positive real, see Section 1.2 of the introduction for a in-depth overview. The partition function of the anti-ferromagnetic Ising model (corresponding to the case $0 < \beta < 1$) has an FPTAS when β is in the uniqueness region of the infinite Δ-regular tree [107]. This uniqueness region turns out to be the interval $((\Delta-2)/\Delta, \Delta/(\Delta-2))$. When $\beta > 1$ (corresponding to the ferromagnetic Ising model) the partition function has an FPRAS on arbitrary graphs (with no restrictions on the degree) by the work of Jerrum and Sinclair [76]. However, in the case of the anti-ferromagnetic Ising model ($\beta \in (0,1)$) this uniqueness/non-uniqueness phase transition is also a computational

transition for the complexity of approximating the partition function of the Ising model: unless $\mathsf{RP} = \mathsf{NP}$, for all $\Delta \geq 3$, there is no FPRAS for approximating the partition function on graphs of maximum degree Δ when $\beta \in (0, (\Delta-2)/\Delta)$ [52]. Interestingly, the uniqueness interval $((\Delta-2)/\Delta, \Delta/(\Delta-2))$ is contained in a complex zero-free region of the partition function of the Ising model.

Theorem 3.23 ([86, Theorem 1.2])**.** *Let Δ be an integer with $\Delta \geq 3$. For any $\beta \in ((\Delta-2)/\Delta, \Delta/(\Delta-2))$, there exists a $\delta > 0$ such that for all $\beta' \in \mathbb{C}$ with $|\beta' - \beta| < \delta$, we have $Z_{\text{Ising}}(G; \beta') \neq 0$ for any graph G with maximum degree at most Δ.*

The argument given in the proof [86, Theorem 1.2] uses continuity to prove the existence of $\delta > 0$ as in the statement. Hence, the zero-free region is not given explicitly. We note that Theorem 3.23 cannot be extended to include more edge interactions $\beta \in (0, (\Delta-2)/\Delta)$ unless $\mathsf{RP} = \mathsf{NP}$ as, by the work of Patel and Regts [101], this would imply easiness of approximating $Z_{\text{Ising}}(G; \beta)$ on graphs with maximum degree Δ.

A recent paper of Barvinok and Barvinok gives another region where $Z_{\text{Ising}}(G; \beta)$ is non-zero [9]. This result actually applies to the multivariate Ising model with a field but it can be stated for our particular case as follows.

Theorem 3.24 ([9, Theorem 1.1])**.** *Let Δ be a positive integer with $\Delta \geq 3$. Let $a \in \mathbb{C}$ and let $\beta = e^{2a}$. Suppose that for some $0 < \delta < 1$ we have $|\text{Re}(a)| < (1-\delta)/\Delta$ and $|\text{Im}(a)| \leq \delta^2/(10\Delta)$. Then $Z_{\text{Ising}}(G; \beta) \neq 0$ for any graph G with maximum degree at most Δ.*

Generally Theorems 1.5 and 3.24 are incomparable for Δ large enough, both of them cover edge interactions that escape from the other result. However, for $\Delta = 3$ Barvinok's region is contained in the region $\mathcal{R}(\tan(\pi/(4\Delta-4)))$ covered by Theorem 1.5. This is depicted in Figure 1.1, where all the regions introduced in this section have been plotted for $\Delta = 3$.

3.3.3 `Mathematica` code for the proof of Lemma 3.22

The following Mathematica code shows that, for any $\delta \in (0, 1/2]$ and $\tau, \theta \in [0, 2\pi)$, we have

$$\left| \frac{e^{\tau i} - e^{\theta i}}{2 + \delta(e^{\tau i} + e^{\theta i})} \right| \leq \frac{2}{\sqrt{3}}, \tag{3.10}$$

which was promised in the proof of Lemma 3.22. The output of the code is `False`. The code uses the (rigorous) `Resolve` function of Mathematica. Here the variables `cos1` and `sin1` take all pair of real values $(\texttt{cos1}, \texttt{sin1})$ such that $\texttt{cos1}^2 + \texttt{sin1}^2 = 1$. In the code, the variables `cos1` and `sin1` are parametrised by a number $\texttt{t1} \in [-1, 1]$ and a sign $\texttt{s1} \in \{1, -1\}$. The variables `cos2` and `sin2` are defined analogously for a new set of parameters $\texttt{t2} \in [-1, 1]$ and $\texttt{s2} \in \{1, -1\}$. Thus, the variable `Z` represents all possible complex numbers of the form $(e^{\tau i} - e^{\theta i})/(2 + \delta(e^{\tau i} + e^{\theta i}))$ for $\tau, \theta \in [0, 2\pi)$, and `X` and `Y` correspond to the real and imaginary parts of `Z`. Finally, the variable `delta` of our code corresponds with $\delta \in (0, 1/2]$. The method resolve shows that there is no set of parameters `t1, t2, s1, s2, delta` such that $\texttt{X}^2 + \texttt{Y}^2 > 4/3$, thus, proving (3.10).

```
cos1 = 2 t1/(1 + t1^2);
sin1 = s1 (1 - t1^2)/(1 + t1^2);
cos2 = 2 t2/(1 + t2^2);
sin2 = s2 (1 - t2^2)/(1 + t2^2);
Z = ComplexExpand[((cos1 + I sin1) - (cos2 + I sin2))/(2 +
      delta (cos1 + cos2 + I (sin1 + sin2)))];
X = Simplify[(Z + ComplexExpand[Conjugate[Z]])/2];
Y = Simplify[(Z - ComplexExpand[Conjugate[Z]])/(2 I)];
Resolve[Exists[{t1, t2, s1, s2, delta},
  X^2 + Y^2 > 4/3 && -1 <= t1 <= 1 && -1 <= t2 <= 1 &&
   0 < delta <= 1/2 && (s1 == 1 || s1 == -1) && (s2 == 1 || s2 == -1)]]
```

3.4 Hardness results: proof of Theorem 1.7

In this section we prove Theorem 1.7. Our hardness proof uses the reduction developed in Chapter 2, based on the binary search technique of Goldberg and Jerrum [59]. Goldberg and Jerrum developed this reduction to obtain #P-hardness results for determining the sign of the Tutte polynomial, recall that the Tutte polynomial includes the partition function of the Ising model as a particular case with the change of variables $q = 2$ and $\gamma = \beta - 1$. This reduction has been further refined in [55] to obtain #P-hardness results for the problem of approximating the norm of the Tutte polynomial. Further refinements have been obtained in Chapter 2, where we give a reduction from exact evaluation of the Tutte polynomial to approximation of this polynomial with complex edge interactions. This later refinement is particularly useful when obtaining hardness results for restricted families of graphs for which exact evaluation of the Tutte polynomial remains hard. Recall that in Chapter 2 we exploited this to prove hardness of approximation for planar graphs whereas here we exploit this reduction to obtain hardness of approximation for bounded-degree graphs for the partition function of the Ising model.

In order to apply the reduction given in Chapter 2 there are a few technical results that we have to develop. The reduction is based on the binary search / interval shrinking technique of Goldberg and Jerrum [59] and this requires us to be able to implement approximations of any real edge interaction efficiently. We formalised this property in Definition 3.1 (recall that we denote by $\mathbb{A}$ the set of real algebraic numbers and we denote by $\mathbb{C}_{\mathbb{A}}$ the set of complex algebraic numbers).

Our work shows that we can implement the complex plane in polynomial time for most pairs (Δ, β). These pairs (Δ, β) are those where $\beta \notin \mathbb{R}$ and $|(\beta - 1)/(\beta + 1)| > 1/\sqrt{\Delta - 1}$, see Lemma 3.2. If we could extend Lemma 3.2 to other pairs (Δ, β), then we could automatically extend Theorem 1.7 to these pairs. In other words, the limiting factor in the proof of Theorem 1.7 is being able to (Δ, β) implement the real line. In fact, most of our work is devoted to this task. The proof of Lemma 3.2 heavily uses the results of [15] as an input. In [15], the authors

(Δ, λ) implement the complex plane in polynomial time for the independent set polynomial for most complex activities λ[3]. Their arguments apply results of complex dynamics in conjunction with the tree recurrence for the independent set polynomial. It turns out that the arguments presented in [15] can be generalised so that they can be applied to other spin systems, and we do so in Section 3.6. We refer to [15, Section 2] or our Section 3.6 for a description of this complex dynamics approach.

This section is organised as follows First, in Section 3.4.1 we introduce the framework needed to implement the real line in polynomial-time (using Section 3.6 as an input). We remark that Section 3.6 is independent of the proofs presented here and can be read on its own. In Section 3.4.2 we use this framework to prove Lemma 3.2. Then in Section 3.4.3 we use Lemma 3.2 in conjunction with the reductions of Chapter 2 to prove our hardness results.

3.4.1 Ising and Mobius programs

In this section we introduce the framework that we use to implement the real line in polynomial time for the Ising model. Our proofs are based on the techniques developed in [15] for the hardcore model. The idea behind the implementation results of [15] is the following one. First, we have to come up with a recursively-constructed gadget that implements a weight $f(z_1, z_2, \ldots, z_d)$ assuming that we can implement $z_1, \ldots, z_d$. Then we apply results of complex dynamics to the function f in order to understand which points we can implement by iterating f. As we will see, it is important that the function f is of the form $g(z_1 z_2 \cdots z_d)$, where g is a Mobius map. In [15] the function f naturally arises from the tree-recurrence for vertex implementations in the hardcore model. Unfortunately vertex-style implementations are useless in the Ising model; due to the perfect symmetry nothing interesting can be implemented through that route. Hence, we need to devise another way to obtain this type of recurrence in the Ising model. This is done in Proposition 3.26 for the Mobius map g_β, which is introduced in Definition 3.25.

Definition 3.25. *Let $\Delta \geq 3$ and $\beta \in \mathbb{C}$, and set $d := \Delta - 1$. Let $h_\beta(x) = (\beta x + 1)/(\beta + x)$ and let $g_\beta(x) = h_\beta(h_\beta(x))$. An Ising-program for β is a sequence $a_0, a_1, \ldots$, starting with $a_0 = \beta$ and satisfying*

$$a_k = g_\beta(a_{i_{k,1}} \cdots a_{i_{k,d_k}}) \quad \text{for } k \geq 1,$$

where $d_k \in [d]$ and $i_{k,1}, \ldots, i_{k,d_k} \in \{0, \ldots, k-1\}$. We say that the Ising program $a_0, a_1, \ldots$ generates $x \in \mathbb{C}$ if there exists an integer $k \geq 0$ such that $a_k = x$.

We use these definitions for h_β and g_β several times in the rest of Section 3.4. We work with Ising-programs from a computational point of view. We represent an element a_k of an Ising-program by the tuples $(i_{j,1}, \ldots, i_{j,d_j})$ for $j \in \{2, \ldots, k\}$, so computing a_k means computing its representation as a sequence of tuples. Proposition 3.26 gives a gadget that implements the edge-interactions generated by an Ising-program.

[3] Here λ is a vertex activity of the independent set polynomial, and a graph G with terminal v (Δ, λ)-implements λ' if $\deg(v) = 1$ and $\lambda' = R(G, v; \lambda)$

Proposition 3.26. *Let $\Delta \geq 3$ and $\beta \in \mathbb{C}$. Suppose that $a_0, a_1, \ldots$ is an Ising-program for β. Then, for every $k \geq 0$, we can compute from the representation of a_k a graph H_k with maximum degree at most Δ that (Δ, β)-implements the edge interaction a_k. This computation takes* $\mathrm{poly}(\Delta, k)$ *steps.*

Proof. Set $d := \Delta - 1$. We give a recursive algorithm for the task of the statement. For $k = 0$, our algorithm outputs the graph with two vertices and one edge joining them. This graph implements the edge interaction $a_0 = \beta$. For $k > 0$, first our algorithm computes recursively graphs $H_0, \ldots, H_{k-1}$ such that H_j (Δ, β)-implements a_j for every $j \in \{0, \ldots, k-1\}$. Since $a_0, a_1, \ldots$ is an Ising-program, we have $a_k = g_\beta(a_{i_{k,1}} \cdots a_{i_{k,d_k}})$ for $d_k \in [d]$ and some indexes $i_{k,1}, \ldots, i_{k,d_k} \in \{0, \ldots, k-1\}$. We have access to these indexes since we have access to the representation of a_k. Our algorithm constructs H_k as the series composition of the following graphs: H_0, the parallel composition of the graphs $H_{i_{k,1}}, \ldots, H_{i_{k,d_k}}$, and H_0. The graph H_k implements the same edge interaction as that implemented by the graph shown in Figure 3.3.

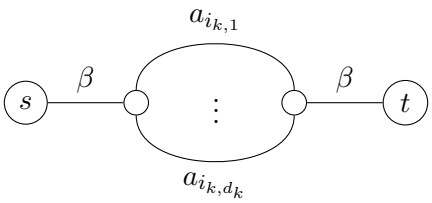


Figure 3.3: The recursive construction for H_k.

By the properties of series and parallel compositions, see (3.4), the graph H_k implements the edge interaction $h_\beta(h_\beta(a_{i_{k,1}} \cdots a_{i_{k,d_k}})) = a_k$. Note that constructing H_k from $H_0, \ldots, H_{k-1}$ takes $\mathrm{poly}(\Delta, k)$ steps, so in total our algorithm has performed at most k times that number of steps. □

Note that g_β is the composition of two Mobius maps and, thus, is a Mobius map. Hence, Ising-programs can be viewed as a particular case of Mobius-programs (see Definition 3.27).

Definition 3.27. *Let $d \geq 2$ be an integer, g be a Mobius map and $a_0 \in \mathbb{C}$. A Mobius-program for g and d starting at a_0 is a sequence of complex numbers $a_0, a_1, \ldots$ of the form*

$$a_k = g(a_{i_{k,1}} \cdots a_{i_{k,d_k}}) \quad \text{for } k \geq 1,$$

where $d_k \in [d]$ and $i_{k,1}, \ldots, i_{k,d_k} \in \{0, \ldots, k-1\}$. We say that the Mobius-program $a_0, a_1, \ldots$ generates $x \in \mathbb{C}$ if there exists a non-negative integer k such that $a_k = x$. We usually omit d when its value is clear from the context.

In [15] the authors studied the points that can be generated by those Mobius-programs starting at $a_0 = \lambda$ for

$$g(x) = \frac{1}{1 + \lambda x},$$

where λ is an activity for the independent set polynomial. They called this program a hardcore-program. The study of hardcore-programs is at the core of the hardness results for the independent set polynomial derived in [15]. It turns out that their results on hardcore-programs can be generalised to our setting of Mobius-programs. In short, their techniques imply that under some hypothesis we can efficiently generate approximations of any complex number with Mobius-programs algorithmically. First, let us introduce some notation that we use to generalise their results.

Definition 3.28. *Let $d \geq 2$ be an integer and let g be a Mobius map. Let $a_0 \in \mathbb{C}$. We say that $\gamma \in \mathbb{C} \setminus \mathbb{R}$ is program-approximable for g, d and a_0 if for each $\varepsilon > 0$, there is a Mobius-program for g and d starting at a_0 that generates a number $x \in \mathbb{C} \setminus \mathbb{R}$ with $0 < |\gamma - x| \leq \varepsilon$.*

Definition 3.29. *Let $d \geq 2$ be an integer and let g be a Mobius map with coefficients in $\mathbb{C}_{\mathbb{A}}$. Let $a_0 \in \mathbb{C}_{\mathbb{A}}$. We say that $\gamma \in \mathbb{C}$ is densely program-approximable in polynomial time for g, d and a_0 if there is $r_\gamma \in \mathbb{A}_{>0}$ such that for each positive integer k there is an algorithm whose inputs are a rational $\varepsilon > 0$ and $\lambda' \in B(\gamma, r_\gamma) \cap \mathbb{C}_{\mathbb{A}}$ that computes, in polynomial time in $\mathrm{size}(\varepsilon)$ and $\mathrm{size}(\lambda')$, k distinct complex numbers $x_1, x_2, \ldots, x_k$ generated by Mobius-programs for g starting at a_0 with $|\lambda' - x_j| \leq \varepsilon$ for all $j \in [k]$.*

In both definitions, we usually omit d when its value is clear from the context.

In [15] the authors consider a fixed point of $f(x) := g(x^d)$ that is program-approximable for their choice of g and a_0 and show that this fixed point is densely program-approximable in polynomial time for g and a_0. Then they use this property in conjunction with results of complex dynamics to generate approximations of any complex number when the fixed point under consideration is repelling. This idea is made precise in Lemmas 3.30 and 3.31. We include our proofs of Lemmas 3.30 and 3.31 in Section 3.6, which require significant extra work as the versions of these results for the hardcore model given in [15] exploit the properties of the Mobius function $1/(1 + \lambda x)$ and, thus, cannot be directly generalised.

Lemma 3.30 ([15, Proposition 2.6 for Mobius-programs])**.** *Let d be an integer with $d \geq 2$ and let g be a Mobius map with coefficients in $\mathbb{C}_{\mathbb{A}}$. Let $f(x) := g(x^d)$ and let ω be a fixed point of f. Let us assume that the following assumptions hold.*

1. *ω is program-approximable for g, d and $a_0 \in \mathbb{C}$;*
2. *$\omega \neq 0$, $g'(\omega^d) \notin \{0, \infty\}$ and $g''(\omega^d) \neq \infty$;*
3. *Let $z := f'(\omega)/d = g'(\omega^d)\omega^{d-1}$. We have $0 < |z| < 1$ and $z \notin \mathbb{R}$.*

Then ω is densely program-approximable in polynomial time for g, d and a_0.

Lemma 3.31 ([15, Proposition 2.2 for Mobius-programs])**.** *Let d be an integer with $d \geq 2$ and let g be a Mobius map with coefficients in $\mathbb{C}_{\mathbb{A}}$ such that $g(\infty) \in \mathbb{C}$. Let $\omega \in \mathbb{C}$ be a repelling fixed point of $f(z) := g(z^d)$ that is densely program-approximable in polynomial time for g and*

$a_0 \in \mathbb{C}_{\mathbb{A}}$. Let E_f be the exceptional set of the rational map f. If $0, \infty \notin E_f$, then the following holds.

There is a polynomial-time algorithm such that, on input $\lambda \in \mathbb{C}_{\mathbb{A}}$ and rational $\varepsilon > 0$, computes an element a_k of a Mobius-program for g starting at a_0 with $|\lambda - a_k| \leq \varepsilon$.

In order to apply Lemmas 3.30 and 3.31, first one has to find a fixed point ω with the properties described in Lemma 3.30. When applying this result to the Ising model, we will set $\omega = 1$. Then one has to find the region of activities / edge interactions where the fixed point is repelling. All this work is carried out in Section 3.4.2.

3.4.2 Proof of Lemma 3.2

In this section we use the framework introduced in Section 3.4.1 to prove Lemma 3.2. This proof strongly uses the properties of the map h_β, which naturally arises in the context of the Ising model. The proof is divided into several technical lemmas. First, we show that $\omega = 1$ is a program-approximable fixed point for the Ising model (Lemma 3.33). Then we prove Lemma 3.2 when $1/\sqrt{\Delta - 1} < |\beta - 1|/|\beta + 1| < 1$. Finally, we address the cases $|\beta - 1|/|\beta + 1| = 1$ and $|\beta - 1|/|\beta + 1| > 1$ separately, as they do not directly follow from the results of Section 3.4.1. We will use the following remark.

Remark 3.32. *Let $\beta, x \in \mathbb{C} \setminus \{1, -1\}$. Then it is straightforward to check that*

$$\frac{h_\beta(x) - 1}{h_\beta(x) + 1} = \frac{(\beta - 1)(x - 1)}{(\beta + 1)(x + 1)}.$$

This equation was observed in (3.1) and plays a key role in the proof of Theorem 1.5. By induction we conclude that, for any positive integer n,

$$\frac{h_\beta^n(x) - 1}{h_\beta^n(x) + 1} = \left(\frac{\beta - 1}{\beta + 1}\right)^n \left(\frac{x - 1}{x + 1}\right).$$

By rearranging this equation, we obtain, for any positive integer n,

$$h_\beta^n(x) = -1 - \frac{2}{\left(\frac{\beta-1}{\beta+1}\right)^n \left(\frac{x-1}{x+1}\right) - 1} = 1 + \frac{2}{\left(\frac{\beta+1}{\beta-1}\right)^n \left(\frac{x+1}{x-1}\right) - 1}.$$

Therefore, we have

$$g_\beta^n(x) = -1 - \frac{2}{\left(\frac{\beta-1}{\beta+1}\right)^{2n} \left(\frac{x-1}{x+1}\right) - 1} = 1 + \frac{2}{\left(\frac{\beta+1}{\beta-1}\right)^{2n} \left(\frac{x+1}{x-1}\right) - 1}.$$

Lemma 3.33. *Let $\beta \in \mathbb{C}$ with $\beta \notin \{i, -i\} \cup \mathbb{R}$. Then there is an Ising-program $a_0, a_1, \ldots$ such that the following holds. For every $\varepsilon > 0$, there is a positive integer k such that $0 < |1 - a_k| \leq \varepsilon$ and $a_k \notin \mathbb{R}$.*

Proof. We note $(x+1)/(x-1) = 1 + 2/(x-1) \in \mathbb{R}$ if and only if $2/(x-1) \in \mathbb{R}$ or, equivalently, $x \in \mathbb{R}$. Thus, we have $(\beta+1)/(\beta-1) \notin \mathbb{R}$. These facts are used repeatedly in this proof. There are three cases:

Case 1: $0 < |\beta-1|/|\beta+1| < 1$. First, let us describe the Ising-program. We define $a_0 = \beta$ and $a_j = g_\beta(a_{j-1})$ for every j with $j \geq 1$. Note that this is an Ising-program. Since $a_j = g_\beta^j(\beta)$ for $j \geq 1$, by Remark 3.32 we obtain

$$a_j - 1 = \frac{2}{\left(\frac{\beta+1}{\beta-1}\right)^{2j+1} - 1}, \tag{3.11}$$

By hypothesis we have $|\beta+1|/|\beta-1| > 1$, so the right hand side of (3.11) converges to 0. Moreover, since $(\beta+1)/(\beta-1) \notin \mathbb{R}$, there are infinitely many positive integers j such that the right hand side of (3.11) is not real. Therefore, we can find a positive integer k with $0 < |1 - a_k| \leq \varepsilon$ and $a_k \notin \mathbb{R}$.

Case 2: $|\beta-1|/|\beta+1| > 1$. First, we give an Ising-program $b_0, b_1, \ldots$ with the property that b_j converges to -1. We define $b_0 = \beta$ and $b_j = g_\beta(b_{j-1})$ for every j with $j \geq 1$. By Remark 3.32 we have

$$b_j + 1 = -\frac{2}{\left(\frac{\beta-1}{\beta+1}\right)^{2j+1} - 1},$$

so $b_j + 1$ converges to 0 because $|\beta-1|/|\beta+1| > 1$. Once we have this Ising program, we define $a_0 = \beta$, $a_{2j-1} = b_j$ and $a_{2j} = g_\beta(a_{2j-1}^2) = g_\beta(b_j^2)$ for all $j \geq 1$. From Remark 3.32 we obtain

$$\frac{a_{2j}-1}{a_{2j}+1} = \frac{g_\beta(b_j^2)-1}{g_\beta(b_k^2)+1} = \left(\frac{\beta-1}{\beta+1}\right)^2 \frac{b_j^2-1}{b_j^2+1}. \tag{3.12}$$

The right hand side of (3.12) converges to 0, so a_{2j} converges to 1. Moreover, (3.12) in combination with $b_j = g_\beta^j(\beta)$ and Remark 3.32 gives $(b_j+1)/(b_j-1) = ((\beta-1)/(\beta+1))^{-2j-1}$ and

$$\frac{a_{2j}-1}{a_{2j}+1} = \left(\frac{\beta-1}{\beta+1}\right)^2 \frac{(b_j+1)(b_j-1)^2}{(b_j-1)(b_j^2+1)} = \left(\frac{\beta-1}{\beta+1}\right)^{-2j+1} \frac{(b_j-1)^2}{b_j^2+1}.$$

Since $(\beta-1)(\beta+1)$ is not real and $(b_j-1)^2/(b_j^2+1)$ converges to 2, there are infinitely many values of j such that $(a_{2j}-1)/(a_{2j}+1)$ is not real. Equivalently, there are infinitely many values of j such that a_{2j} is not real. Hence, for every $\varepsilon > 0$, there is a positive integer k such that $0 < |1 - a_{2k}| \leq \varepsilon$ and $a_{2k} \notin \mathbb{R}$.

Case 3: $|\beta-1|/|\beta+1| = 1$. Then we note that $\beta \in \mathbb{R}i$ (Proposition 3.16, Item 3). We can write $\beta = ci$ with c a real number with $c \notin \{0, 1, -1\}$, where we used that $\beta \notin \{0, i, -i\}$. We consider $\gamma = g_\beta(\beta^2)$. We claim that $\gamma \notin \{i, -i\} \cup \mathbb{R}$ and $|\gamma-1|/|\gamma+1| > 1$. Assuming this, we obtain our Ising-program as $b_0 = \beta$, $b_1 = \gamma$ and $b_j = a_{j-1}$ for all $j \geq 2$, where $a_0, a_1, \ldots$ is the Ising-program of Case 2 with $\beta = \gamma$. We study γ to conclude the proof. We note that γ is the edge interaction implemented by the series composition of three edges with edge interactions β, β^2 and β. Recall that series compositions are commutative when it comes to the weight they implement (see Section 3.2.3) and that the weight implemented by the series composition of

two graphs implementing w_1 and w_2 is $(w_1w_2+1)/(w_1+w_2) = h_{w_1}(w_2) = h_{w_2}(w_1)$. Thus, we have $\gamma = h_\beta(h_\beta(\beta^2))$. This is also the edge interaction implemented by the series composition of three edges with edge interactions β, β and β^2, so we can also write $\gamma = h_{\beta^2}(h_\beta(\beta))$. From the expression $\gamma = h_\beta(h_\beta(\beta^2))$ and Remark 3.32 we find that

$$\left|\frac{\gamma-1}{\gamma+1}\right| = \left|\frac{g_\beta(\beta^2)-1}{g_\beta(\beta^2)+1}\right| = \left|\frac{\beta-1}{\beta+1}\right|^2 \left|\frac{\beta^2-1}{\beta^2+1}\right| = \left|\frac{1+c^2}{1-c^2}\right| > 1,$$

where we used that $c \neq \pm 1$. In particular, we have $\gamma \neq \pm i$. From the expression $\gamma = h_{\beta^2}(h_\beta(\beta))$ we are going to show that $\gamma \notin \mathbb{R}$, which would complete the proof. We have

$$h_{\beta^2}(x) = h_{-c^2}(x) = \frac{-c^2x+1}{-c^2+x} = \frac{\left(-c^2x+1\right)\left(\overline{x}-c^2\right)}{|x-c^2|^2} = \frac{-c^2|x|^2+\overline{x}+c^4x-c^2}{|x-c^2|^2}.$$

Since $c^4 \neq 1$ because $c \neq \pm 1$, we find that $h_{\beta^2}(x)$ is non-real for any non-real x. In particular this is the case for $x = h_\beta(\beta)$ as $h_\beta(\beta) = (1+\beta^2)/(2\beta) = -i(1-c^2)/(2c) \notin \mathbb{R}$. We conclude that $\gamma = h_{\beta^2}(h_\beta(\beta))$ is not real as we wanted. □

Using the notation of Section 3.6 (see Definition 3.28), the statement of Lemma 3.33 implies "1 is program-approximable for g_β and $a_0 = \beta$ for any $\beta \in \mathbb{C} \setminus (\mathbb{R} \cup \{i, -i\})$". This is one of the three conditions that we have to check to apply Lemma 3.30 with $\omega = 1$ in our current setting. Lemma 3.34 shows that the two other conditions hold for some edge interactions β.

Lemma 3.34. *Let d be an integer with $d \geq 2$ and let $\beta \in \mathbb{C} \setminus (\mathbb{R} \cup \{i, -i\})$. Let $f_\beta(x) = g_\beta(x^d)$, where g_β is as in Definition 3.25. Let $z = f'_\beta(1)/d$. If $0 < |\beta-1|/|\beta+1| < 1$ and $|\beta| \neq 1$, then*

1. *$g'_\beta(1) \notin \{0, \infty\}$, $g''_\beta(1) \neq \infty$;*
2. *$0 < |z| < 1$ and $z \notin \mathbb{R}$.*

Proof. Let us determine z, $g'_\beta(1)$ and $g''_\beta(1)$. We have $h'_\beta(x) = (\beta^2-1)/(\beta+x)^2$ and $h''_\beta(x) = 2(1-\beta^2)/(\beta+x)^3$. Hence, we obtain $g'_\beta(1) = h'_\beta(h_\beta(1))h'_\beta(1) = (\beta-1)^2/(\beta+1)^2$. Since $0 < |\beta-1|/|\beta+1| < 1$, we have $0 < |g'_\beta(1)| < 1$, so $g'_\beta(1) \notin \{0, \infty\}$. Moreover, from $z = f'(1)/d = g'_\beta(1)$, we obtain $0 < |z| < 1$. Note that $(\beta-1)^2/(\beta+1)^2 \in \mathbb{R}$ if and only if $(\beta-1)/(\beta+1) \in \mathbb{R} \cup \mathbb{R}i$. Also note that $(\beta-1)/(\beta+1) = 1-2/(\beta+1)$, so $(\beta-1)/(\beta+1) \in \mathbb{R}$ if and only if $\beta \in \mathbb{R}$. If $(\beta-1)/(\beta+1) = ci$ for some $c \in (-1, 1)$, then we obtain

$$\beta = \frac{1+ci}{1-ci} = \frac{1-c^2}{1+c^2} + \frac{2c}{1+c^2}i,$$

so $|\beta|^2 = 1$. Since $\beta \notin \mathbb{R}$ and $|\beta| \neq 1$ by hypothesis, we find that $z = g'_\beta(1) = (\beta-1)^2/(\beta+1)^2 \notin \mathbb{R}$ as we wanted. Finally, let us determine $g''_\beta(1)$. We have $g''_\beta(1) = -4(\beta-1)^2b/\beta+1)^4 \notin \{0, \infty\}$, where we used that $\beta \notin \{1, -1\}$. This finishes the proof. □

Remark 3.35. *The map $f_\beta(z) = g_\beta(z^d)$ does not have exceptional points. To see this, we apply Lemma 3.14. First, let us determine the points of f_β with multiplier 0. We have*

$$f_\beta(z) = \frac{(\beta^2+1)z^d+2\beta}{\beta^2+1+2\beta z^d} \qquad \text{and} \qquad f'_\beta(z) = dz^{(d-1)}\frac{(\beta^2-1)^2}{(1+2\beta z^d+\beta^2)^2},$$

so the only point with multiplier 0 *is* $z = 0$*. However,* 0 *is not a fixed point of* f_β *because* $f_\beta(0) = 2\beta/(1+\beta^2)$*, so* f_β *does not have any exceptional points.*

Now we combine all the results obtained so far in this section obtaining Corollaries 3.36 and 3.37.

Corollary 3.36. *Let* Δ *be an integer with* $\Delta \geq 3$ *and let* $\beta \in \mathbb{C}_{\mathbb{A}} \setminus \mathbb{R}$ *with* $|\beta| \neq 1$ *and* $0 < |\beta - 1|/|\beta + 1| < 1$*. There is a rational number* $r \in (0,1)$ *and a polynomial-time algorithm such that, on input* $\lambda \in B(1,r) \cap \mathbb{C}_{\mathbb{A}}$ *and rational* $\varepsilon > 0$*, computes a graph* G *that* (Δ, β)*-implements a complex number* $\hat{\lambda}$ *with* $|\lambda - \hat{\lambda}| \leq \varepsilon$*.*

Proof. Set $d := \Delta - 1$. Lemma 3.33 and Lemma 3.34 provide us with the three conditions that the fixed point $\omega = 1$ of $f_\beta(x) = g_\beta(x^d)$ has to satisfy to apply Lemma 3.30. We find that 1 is densely program-approximable in polynomial time for g_β, d and $a_0 = \beta$. In terms of Ising-programs, this gives (Definition 3.29 with $k = 1$) that there is $r > 0$ and an algorithm, on inputs a rational $\varepsilon > 0$ and $\lambda \in B(1,r) \cap \mathbb{C}_{\mathbb{A}}$, that computes, in polynomial time in size(ε) and size(λ) a complex number $\hat{\lambda}$ generated by an Ising-program with $|\lambda - \hat{\lambda}| \leq \varepsilon$. This can be then translated to the result given in the statement by applying Proposition 3.26. □

Corollary 3.37. *Let* Δ *be an integer with* $\Delta \geq 3$ *and let* $\beta \in \mathbb{C}_{\mathbb{A}} \setminus \mathbb{R}$ *with* $|\beta| \neq 1$ *and* $1/\sqrt{\Delta - 1} < |\beta - 1|/|\beta + 1| < 1$*. Then the pair* (Δ, β) *implements the complex plane in polynomial time for the Ising model.*

Proof. Set $d := \Delta - 1$. The proof starts the same way as the proof of Corollary 3.36. The difference here is that once we show that 1 is densely program-approximable in polynomial time for g_β, d and $a_0 = \beta$, we use this property to apply Lemma 3.31. First, we have two check the other two hypothesis of Lemma 3.31. The first hypothesis that 1 is a repelling fixed point of f_β or, equivalently, $|f'_\beta(1)| > 1$. This follows from $1/\sqrt{\Delta - 1} < |\beta - 1|/|\beta + 1|$ since $f'_\beta(1) = d(\beta - 1)^2/(\beta + 1)^2$. The second hypothesis is that 0 and ∞ are not exceptional points of the rational map f_β, which holds because f_β does not have exceptional points, see Remark 3.35. We conclude by Lemma 3.31 that there is a polynomial-time algorithm such that, on input $\lambda \in \mathbb{C}_{\mathbb{A}}$ and rational $\varepsilon > 0$, computes an element a_k of an Ising-program with $|\lambda - a_k| \leq \varepsilon$. The result now follows by applying the algorithm of Proposition 3.26 to translate the obtained Ising-program to a graph that (Δ, β)-implements a_k. □

Finally, we extend Corollaries 3.36 and 3.37 to the rest of the complex plane when possible.

Lemma 3.38. *Let* Δ *be an integer with* $\Delta \geq 3$ *and let* $\beta \in \mathbb{C}_{\mathbb{A}} \setminus \mathbb{R}$ *with* $\beta \notin \{i, -i\}$*. There is a rational number* $r \in (0,1)$ *and a polynomial-time algorithm such that, on input* $\lambda \in B(1,r) \cap \mathbb{C}_{\mathbb{A}}$ *and rational* $\varepsilon > 0$*, computes a graph* G *that* (Δ, β)*-implements a complex number* $\hat{\lambda}$ *with* $|\lambda - \hat{\lambda}| \leq \varepsilon$*.*

Proof. We recall that $|\beta - 1|/|\beta + 1| < 1$ if and only if $\mathrm{Re}(\beta) > 0$, and $|\beta - 1|/|\beta + 1| = 1$ if and only if $\mathrm{Re}(\beta) = 0$, see Proposition 3.16, Item 3. We distinguish three cases based on this observation:

Case 1: $|\beta| \neq 1$ and $0 < |\beta - 1|/|\beta + 1| < 1$. This case is exactly Corollary 3.36.

Case 2: $|\beta| = 1$ and $0 < |\beta - 1|/|\beta + 1| < 1$. We have $\mathrm{Re}(\beta) > 0$. We consider the edge interaction $\beta' = h_\beta(\beta)$ that is implemented by a path of length two with weights β. We have $\beta' = (\beta^2 + 1)/(2\beta) = (\beta + \beta^{-1})/2 = \mathrm{Re}(\beta) \in (0, 1)$, where we used that $|\beta| = 1$ and $\mathrm{Re}(\beta) > 0$. We now consider the Mobius map $h_{\beta'}(x) = (\beta' x + 1)/(\beta' + x) = (\beta' x + 1)/(\overline{\beta'} + x)$, where we used that β' is real. It is well known that this Mobius map fixes $\{x \in \mathbb{C} : |x| = 1\}$ (Proposition 3.10). Moreover, $h_{\beta'}(0) = 1/\mathrm{Re}(\beta) \in (1, \infty)$. Hence, the Mobius map $h_{\beta'}$ sends the open unit disk $\mathbb{D} := B(0, 1)$ to $\widehat{\mathbb{C}} \setminus \mathbb{D}$ and it sends $\widehat{\mathbb{C}} \setminus \mathbb{D}$ to $\mathbb{D}$. We conclude that $g_{\beta'}(\mathbb{D}) = \mathbb{D}$. Therefore, $\beta \cdot \beta' \in \mathbb{D}$ and $\gamma := g_{\beta'}(\beta \cdot \beta') \in \mathbb{D}$. We can implement the edge interaction γ using the graph given in Figure 3.4. We have

$$h_{\beta'}(x) = \frac{\beta'^2 x + \beta' + \beta' |x|^2 + \overline{x}}{|\beta' + x|^2},$$

so, since $\beta' \in (0, 1)$, the Mobius map $h'_\beta(x)$ sends points with positive real part to points with positive real part, and non-real points to non-real points. Hence, the Mobius map $g_{\beta'}(x) = h_{\beta'}(h_{\beta'}(x))$ also has these properties. We conclude that $\gamma = g_{\beta'}(\beta \cdot \beta')$ has positive real part and is not real. Putting all this together, γ is a non-real number with $|\gamma| < 1$ and $0 < |\gamma - 1|/|\gamma + 1| < 1$, so γ is in the first case of this proof. We can translate the algorithm of the first case of this proof for γ to an algorithm for β because we can (Δ, β)-implement γ, see Section 3.2.3 for the transitivity property of implementations.

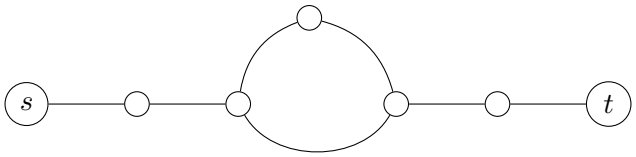


Figure 3.4: A graph that $(3, \beta)$-implements γ.

Case 3: $|\beta - 1|/|\beta + 1| \geq 1$. We can use the Ising program of Lemma 3.33 to generate $a_k \in \mathbb{C} \setminus \mathbb{R}$ with $|1 - a_k| < 1/2$. We can (Δ, β)-implement a_k with the help of Proposition 3.26. Note that $0 < |a_k - 1|/|a_k + 1| < 1$, so the edge interaction a_k is in one of the first two cases of the proof. Again from the transitivity property of implementations, we can translate the algorithm of the first two cases for a_k to an algorithm for β, concluding the proof. □

Lemma 3.2. *Let Δ be an integer with $\Delta \geq 3$ and let $\beta \in \mathbb{C}_{\mathbb{A}} \setminus \mathbb{R}$ with $\beta \notin \{i, -i\}$ and $1/\sqrt{\Delta - 1} < |\beta - 1|/|\beta + 1|$. Then the pair (Δ, β) implements the complex plane in polynomial time for the Ising model.*

Proof. Set $d := \Delta - 1$. Let r be the positive real number given in Lemma 3.38. Let $f_\beta(x) = g_\beta(x^d)$. As argued in Corollary 3.37, 1 is a repelling fixed point of f_β so, by Lemma 3.12, 1 belongs to the Julia set of f and, thus, by Theorem 3.13, $\bigcup_{n=0}^{\infty} f_\beta^n(B(1, r)) = \widehat{\mathbb{C}} \setminus E_{f_\beta}$, where E_{f_β} is the set of exceptional points of f_β. In view of Remark 3.35, E_{f_β} is empty. Let $\gamma = 10(1 + i)$. There is a positive integer N such that $\gamma \in f_\beta^N(B(1, r))$. Thus, there is $x^* \in B(1, r) \cap \mathbb{C}_{\mathbb{A}}$ such that

$f^N(x^*) = \gamma$. By the continuity of the rational function f^N at x^*, there is $\delta \in (0, r)$ such that $|f^N(x^*) - f^N(x)| \leq 0.01$ for every $x \in B(x^*, \delta)$. The constants γ and 0.01 are not chosen to be optimal but to make the notation and proof simpler for the reader. In view of Lemma 3.38 we can compute (in constant time) a graph G that (Δ, β)-implements a complex number $\hat{x}$ with $|x^* - \hat{x}| \leq \delta$. Let $\hat{\gamma} = f_\beta^N(\hat{x})$. Note that we can (Δ, β) implement $\hat{\gamma}$ using the construction from Proposition 3.26 and the fact that we can (Δ, β) implement $\hat{x}$. By continuity, we have $|\gamma - \hat{\gamma}| \leq 0.01$. Note that $\hat{\gamma}$ is in $\mathbb{C}_{\mathbb{A}}$. Moreover, we have $\mathrm{Re}(\hat{\gamma}) > 0$, so $|\hat{\gamma} - 1|/|\hat{\gamma} + 1| < 1$. From $|\gamma - \hat{\gamma}| \leq 0.01$ and the triangle inequality we have

$$\left|\frac{\hat{\gamma} - 1}{\hat{\gamma} + 1}\right| = \left|\frac{(\hat{\gamma} - \gamma) + \gamma - 1}{(\hat{\gamma} - \gamma) + \gamma + 1}\right| \geq \frac{|\gamma - 1| - 0.01}{|\gamma + 1| + 0.01} > 1/\sqrt{2}.$$

Hence, the edge interaction $\hat{\gamma}$ is in the region covered by Corollary 3.37, so the pair $(\Delta, \hat{\gamma})$ implements the complex plane in polynomial time for the Ising model. We conclude that the pair (Δ, β) implements the complex plane in polynomial time for the Ising model thanks to the transitivity property of implementations. □

The complex dynamic argument presented in the proof of Lemma 3.2 is one of the main ideas behind the results of [15] and is applied twice in Section 3.6. The proof of Lemma 3.2 is simpler than the ones presented in Section 3.6 because here we are only trying to approximate γ instead of approximating any number in a neighbourhood of γ. This allows us to use the continuity of f^N at x^* instead of having to use Lipschitz properties of f^N and careful approximations of the quantities involved.

Remark 3.39. *Lemma 3.2 can be extended to other points with $1/\sqrt{\Delta - 1} > |\beta - 1|/|\beta + 1|$. However, we have not found a systematic way to do this. Rather we are aware of points β with $1/\sqrt{\Delta - 1} > |\beta - 1|/|\beta + 1| > 1/(\Delta - 1)$ that can be used to (Δ, β)-implement edges interactions that are covered by Lemma 3.2. For example, this is the case of those points β such that there is a "nice" graph G with $Z_{\mathrm{Ising}}(G; \beta) = 0$. This is made precise in Section 3.5.*

3.4.3 Reducing exact computation to approximate computation

In this section we use our implementation results to prove the hardness of approximating the partition function of the Ising model on bounded degree graphs. A basic building block for the reduction is the binary search (interval-shrinking) technique developed by Goldberg and Jerrum in the context of the Tutte polynomial [59]. Since the partition function of the Ising model is a special case of the Tutte polynomial, this building block is also applicable here. The interval-stretching technique requires us to be able to implement the real line in polynomial time, and this is the motivation behind the results of Section 3.4.2.

We use the version of the interval-shrinking technique that we have developed on Chapter 2, as it is the first such reduction that applies in the context of non-real edge interactions. Moreover, the reduction developed in Chapter 2 is particularly relevant for us because the starting point for the hardness result is the problem of exactly evaluating the Tutte polynomial, and crucially

this problem remains #P-hard even in the $q = 2$ case (corresponding to the Ising model) and even when the input is restricted to be a 3-regular graph [82] (which we require here). In order to apply this reduction, we have to re-define the computational problems considered including the parameter Δ corresponding to the maximum degree of the input graph, thus here we briefly re-introduce the reduction given in Section 3.4.2 under our new notation, which should also help the reader to follow this section. Then we show how this reduction applies to bounded degree graphs.

First, let us recall some definitions. We have $Z_{\text{Ising}}(G;\beta) = Z_{\text{Tutte}}(G;2,\beta-1)$, where Z_{Tutte} is the Tutte polynomial as in (1.5), see, for instance, [109]. Let s and t be two distinct vertices of G. In Section 3.4.2 we defined

$$Z_{st}(G;q,\gamma) = \sum_{\substack{A\subseteq E:\\ s \text{ and } t \text{ in the same component}}} q^{k(A)}\gamma^{|A|}$$

Analogously, we defined $Z_{s|t}$ as the contribution to $Z_{\text{Tutte}}(G;q,\gamma)$ from the configurations $A \subseteq E$ such that s and t are in different connected components in (V,A). That is, $Z_{s|t}(G;q,\gamma) = Z_{\text{Tutte}}(G;q,\gamma) - Z_{st}(G;q,\gamma)$. We now introduce the computational problems that we are interested in, for any rational numbers $q > 0$, $\gamma > 0$, any integer $\Delta \geq 3$ and any $\beta \in \mathbb{C}_{\mathbb{A}}$.

Name: $\textsc{Ising}(\Delta,\beta)$.
Instance: A graph $G = (V,E)$ with maximum degree at most Δ.
Output: The number $Z_{\text{Ising}}(G;\beta) \in \mathbb{C}_{\mathbb{A}}$.

Name: $\textsc{RatioTutte}(\Delta,q,\gamma)$.
Instance: A graph $G = (V,E)$ with maximum degree at most Δ and an edge (s,t) of G.
Output: The rational number $Z_{s|t}(G;q,\gamma)/Z_{st}(G;q,\gamma)$.

In Chapter 2 we defined $\textsc{RatioTutte}(\Delta,q,\gamma)$ more generally; there are no restrictions on the maximum degree of the input graph and the vertices s and t are only required to be in the same connected component of G. Moreover, q and γ could be any non-zero algebraic numbers (possibly non-real or negative real), so we had to study carefully the possibility that $Z_{st}(G;q,\gamma) = 0$. Thus, our simplified version of $\textsc{RatioTutte}(\Delta,q,\gamma)$ requires a slightly simpler argument to conclude Lemmas 3.40 and 3.41.

Lemma 3.40 (Bounded degree version of Lemmas 2.41 and 2.42 for the Ising model)**.** *Let K be a real number with $K > 1$. Let $\Delta \geq 3$ be an integer and let $\beta \in \mathbb{C}_{\mathbb{A}}$ such that (Δ,β) implements the real line in polynomial time. Let $y \in \mathbb{C}$ with $y > 1$. Then we have the reductions*

$$\textsc{RatioTutte}(\Delta,2,y-1) \leq_T \textsc{IsingNorm}(\Delta,\beta,K),$$
$$\textsc{RatioTutte}(\Delta,2,y-1) \leq_T \textsc{IsingArg}(\Delta,\beta,\pi/3).$$

Proof. The proof is the almost the same as that of Lemmas 2.41 and 2.42. Here we indicate how we adapt the reduction of Lemmas 2.41 and 2.42 to graphs with maximum degree Δ. First, let us translate our Ising notation to the notation used in the proofs of Chapter 2. In the original proof we have two weights $\gamma_1 \in (-2,-1)$ and $\gamma_2 > 0$ and access to an oracle that approximates the norm or determines the sign of the multivariate Tutte polynomial on weighted graphs with weights in $\{\gamma_1, \gamma_2\}$. Note that determining the sign reduces to additively approximating the argument of this polynomial with error at most $\pi/3$, so we can use our oracle $\textsc{IsingArg}(\Delta, \beta, \pi/3)$ instead. The purpose of the weights γ_1 and γ_2 is implementing the real line in polynomial time for the Tutte polynomial (using Corollary 2.9). Here the role of these weights is performed by β. Hence, every time Corollary 2.9 is used in the proof of Lemmas 2.41 and 2.42 we use the fact that (Δ, β) implements the real line in polynomial time instead. The reduction of Lemmas 2.41 and 2.42 computes the ratios $Z_{s|t}(H;q,\gamma)/Z_{st}(H;q,\gamma)$ for some positive number γ that can be implemented using γ_1 and γ_2. Here we set $\gamma = y - 1$ instead. The only relevant properties of γ in the proof of Lemmas 2.41 and 2.42 are $\gamma > 0$ and the fact that γ can be implemented exactly.

There are two differences between this proof and the proof of Lemmas 2.41 and 2.42. Let H and (s,t) be the inputs of $\textsc{RatioTutte}(\Delta, 2, y-1)$. The first difference in the proof is that we restrict ourselves to computing ratios $Z_{s|t}(H;q,\gamma)/Z_{st}(H;q,\gamma)$ where (s,t) is an edge of H. This is so that all the graphs considered in the reduction have maximum degree at most Δ. The original proof applies one of the oracles $\textsc{IsingNorm}(\Delta, \beta, K)$ and $\textsc{IsingArg}(\Delta, \beta, \pi/3)$ to a copy of H with an extra edge joining s and t. This extra edge has a weight γ' that is updated repeatedly during the binary search. The weight γ' is implemented using Corollary 2.9 or, in our case, using the fact that (Δ, β) implements the real line in polynomial time. Instead of adding an extra edge between s and t, here we modify the edge (s,t) so that its weight is $\gamma \cdot \gamma'$, producing the same effect as adding an extra edge from s to t with weight γ'. This time we have to implement $\gamma \cdot \gamma'$ instead. Let H' be the graph obtained by copying H and substituting the edge (s,t) with an appropriate graph that (Δ, β)-implements $\gamma \cdot \gamma'$. Then the graph H' also has maximum degree at most Δ. Moreover, for $\varepsilon = \gamma' + 1$ we have, see (2.24),

$$\begin{aligned} Z_{\text{Tutte}}(H';q,\gamma) &= Z_{st}(H;q,\gamma)(1+\gamma') + Z_{s|t}(H;q,\gamma)\left(1+\frac{\gamma'}{q}\right) \\ &= Z_{s|t}(H;q,\gamma)\left(1-\frac{1}{q}\right) + \varepsilon\left(Z_{st}(H;q,\gamma) + \frac{1}{q}Z_{s|t}(H;q,\gamma)\right) \\ &= f(\varepsilon;H,\gamma), \end{aligned}$$

where $f(\varepsilon;H,\gamma)$ is the linear function to which the binary search will be performed. The purpose of the binary search is finding a zero of $f(\varepsilon;H,\gamma)$, which allows us to compute the ratio $Z_{s|t}(H;q,\gamma)/Z_{st}(H;q,\gamma)$.

The second difference is that we cannot implement γ exactly. We can bypass this by using a very close approximation $\hat{\gamma}$ of γ instead. We use the fact that we can (Δ, β)-implement $\hat{\gamma}$ with $|\gamma - \hat{\gamma}| \leq \delta$ in polynomial time in the size of δ. We perform the binary search on

$f(\varepsilon; H, \hat{\gamma})$ instead. This allows us to compute the number $Z_{s|t}(H; q, \hat{\gamma})/Z_{st}(H; q, \hat{\gamma})$. We can choose δ with $\text{size}(\delta) \in \text{poly}(\text{size}(\varepsilon), \text{size}(H))$ such that $|Z_{s|t}(H; q, \hat{\gamma}) - Z_{s|t}(H; q, \gamma)| \leq \varepsilon$ and $|Z_{st}(H; q, \hat{\gamma}) - Z_{st}(H; q, \gamma)| \leq \varepsilon$, see for instance Lemma 2.54. Therefore, the error that we make by outputting $Z_{s|t}(H; q, \hat{\gamma})/Z_{st}(H; q, \hat{\gamma})$ instead of $Z_{s|t}(H; q, \gamma)/Z_{st}(H; q, \gamma)$ can be made to be at most ε by choosing δ with $\text{size}(\delta) \in \text{poly}(\text{size}(\varepsilon), \text{size}(H))$ thanks to the lower and upper bounds on $|Z_{s|t}(H; q, \cdot)|$ and $|Z_{st}(H; q, \cdot)|$, see Section 2.5.1 for these bounds. We conclude that we can compute $Z_{s|t}(H; q, \gamma)/Z_{st}(H; q, \gamma)$ exactly. This can be done using the algorithm of Kannan, Lenstra and Lovász stated in Lemma 2.37, as we did for algebraic numbers in the proof of Lemma 2.41, or a simpler version of Lemma 2.37 if we restrict this lemma to rational numbers. □

Lemma 3.41 (Particular case of Lemma 2.48)**.** *Let $\Delta \geq 3$ be an integer and let $\beta \in \mathbb{Q}$ with $\beta > 0$. Then we have the reduction*

$$\text{Ising}(\Delta, \beta) \leq_T \text{RatioTutte}(\Delta, 2, \beta - 1).$$

Proof. The reduction given in the proof of Lemma 2.48 applies with the change of variables $q = 2$ and $\gamma = \beta - 1$. It is important to note that this reduction only invokes the oracle for $\text{RatioTutte}(\Delta, 2, \beta - 1)$ with inputs (G, s, t) such that $e = (s, t)$ is an edge of G. The reduction reduces the computation of $Z_{\text{Tutte}}(G; q, \gamma)$ to that of $Z_{\text{Tutte}}(G \setminus e; q, \gamma)$, $Z_{s|t}(G; q, \gamma)/Z_{st}(G; q, \gamma)$ and $Z_{s|t}(G \setminus e; q, \gamma)/Z_{st}(G \setminus e; q, \gamma)$, where $G \setminus e$ is the graph G without the edge e. Hence, all the calls to the oracle $\text{RatioTutte}(\Delta, 2, \beta - 1)$ involve subgraphs of G, that have maximum degree at most Δ. Finally, because $q > 0$ and $\gamma > 0$ in our setting, we do not have to consider the cases when $Z_{st}(G; q, \gamma) = 0$, simplifying the result. □

Now we have the tools to obtain the desired reductions and the proof of Theorem 1.7.

Lemma 3.42 (Lemma 2.49 for the Ising model)**.** *Let K be a real number with $K > 1$. Let $\Delta \geq 3$ be an integer and let $\beta \in \mathbb{C}_{\mathbb{A}}$ such that (Δ, β)-implements the real line in polynomial time. Let $y \in \mathbb{C}$ with $y > 1$. Then we have the reductions*

$$\text{Ising}(\Delta, y) \leq_T \text{IsingNorm}(\Delta, \beta, K),$$
$$\text{Ising}(\Delta, y) \leq_T \text{IsingArg}(\Delta, \beta, \pi/3).$$

Proof. This result follows directly from combining Lemmas 3.40 and 3.41. The proof of Lemma 2.49 takes a bit more work than this lemma because one has to be careful about possible zeros of $Z_{st}(G; q, \gamma)$. □

Theorem 1.7. *Let Δ be an integer with $\Delta \geq 3$ and let $\beta \in \mathbb{C}$ be an algebraic number such that $\beta \notin \mathbb{R} \cup \{i, -i\}$ and $|\beta - 1|/|\beta + 1| > 1/\sqrt{\Delta - 1}$. Then the problems $\text{IsingNorm}(\beta, \Delta, 1.01)$ and $\text{IsingArg}(\beta, \Delta, \pi/3)$ are #P-hard.*

Proof. Our hardness theorem now follows from combining Lemmas 3.2 and 3.42 in conjunction with the fact that $\text{Ising}(3, y)$ is #P-hard for any $y > 1$ [82]. □

Since $1/\sqrt{\Delta-1}$ converges to 0 as Δ diverges, Theorem 1.7 gives a new proof of Theorem 1.3 which says that IsingNorm$(\beta, \infty, 1.01)$ and IsingArg$(\beta, \infty, \pi/3)$ (where there are no restrictions on the maximum degree of the input graph) are #P-hard for any algebraic number $\beta \in \mathbb{C} \setminus (\mathbb{R} \cup \{i, -i\})$.

3.5 Zeros of the partition function and hardness

In this section we give explicit evidence that zeros of the partition function imply hardness of approximation for the Ising model when the edge interaction β is not in $\mathbb{R} \cup \{i, -i\}$. These are the first results that explicitly link zeros to hardness of approximation that we are aware of. Our main technical result Lemma is 3.3, which shows that implementing -1 implies hardness of approximation of the partition function of the Ising model. We then use zeros of the partition function to implement -1 and conclude hardness in Lemma 3.43 and Corollary 3.45. Our proofs use the hardness and implementation results of Section 3.4. Finally, in Corollary 3.44, we give an example of an edge interaction β in the region $\mathcal{R}(1/\sqrt{2})$ and a graph G with maximum degree 3 such that $Z_{\text{Ising}}(G; \beta) = 0$, showing that the hardness region given in Theorem 1.7 is not optimal.

Lemma 3.3. *Let Δ be an integer with $\Delta \geq 3$. Let $\beta \in \mathbb{C}_{\mathbb{A}} \setminus (\mathbb{R} \cup \{i, -i\})$. Let us assume that (Δ, β) implements the edge interaction -1. Then* IsingNorm$(\Delta, \beta, 1.01)$ *and* IsingArg$(\Delta, \beta, \pi/3)$ *are* #P*-hard.*

Proof. There are two cases. The first case is when $|\beta - 1|/|\beta + 1| > 1/\sqrt{\Delta - 1}$. Then, since $\beta \notin \mathbb{R} \cup \{i, -i\}$, we know that the problems IsingNorm$(\Delta, \beta, 1.01)$ and IsingArg$(\beta, \Delta, \pi/3)$ are #P-hard (Theorem 1.7). In the rest of the proof we assume that $|\beta - 1|/|\beta + 1| \leq 1/\sqrt{\Delta - 1}$. We are going to reduce the approximation problems at (Δ, γ) to the approximation problems at (Δ, β) for some γ such that IsingNorm$(\Delta, \gamma, 1.01)$ and IsingArg$(\Delta, \gamma, \pi/3)$ are #P-hard. In this reduction we will use the fact that we can (Δ, β) implement the edge interaction -1. Let $\alpha \in \mathbb{C}$ be some edge interaction that we can (Δ, β)-implement. We fine-tune α later in the proof. We consider the weighted graph J given in Figure 3.5. By the properties of series and parallel compositions, this graph implements the edge interaction $\gamma := h_\beta(h_\beta(-\alpha))$.

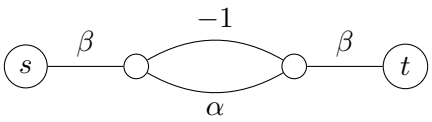


Figure 3.5: The graph J in the proof of Lemma 3.3.

From Remark 3.32 it follows that

$$\frac{\gamma - 1}{\gamma + 1} = \left(\frac{\beta - 1}{\beta + 1}\right)^2 \frac{-\alpha - 1}{-\alpha + 1}. \tag{3.13}$$

The idea to complete this proof is (Δ, β)-implementing α so that the complex number in (3.13) has norm larger than 1 (hence larger than $1/\sqrt{\Delta - 1}$ so Theorem 1.7 applies). By

Item 3 of Proposition 3.16, the norm is larger than 1 if and only if $\mathrm{Re}(\gamma) < 0$, which is what we are aiming for. We also want γ to be non-real. Note that γ is real if and only if $1 - 2/(\gamma+1) = (\gamma-1)/(\gamma+1)$ is real. Let $r \in (0,1)$ be the rational number in the statement of Lemma 3.38. Let $\varepsilon = |(\beta-1)/(\beta+1)|^2 r/2$, so ε is an algebraic number with $\varepsilon \in (0, 1/(\Delta-1))$ and $\varepsilon \leq r/4$. Let $\xi \in \mathbb{C}_{\mathbb{A}}$ such that

$$\frac{\xi-1}{\xi+1} = \frac{r(\beta-1)^2}{4(\beta+1)^2} i.$$

We have $|\xi-1|/|\xi+1| = \varepsilon/2$, so $\xi \in \mathcal{R}(\varepsilon/2)$. From Proposition 3.16, we obtain $\mathrm{Re}(\xi) \geq 0$ and

$$|\xi| \leq \frac{1+\varepsilon/2}{1-\varepsilon/2} \leq \frac{1+1/4}{1-1/4} = 5/3,$$

where we used that $0 < \varepsilon \leq 1/(\Delta-1) \leq 1/2$. Thus, we have

$$|\xi - 1| = |\xi+1|\frac{\varepsilon}{2} \leq (5/3+1)\frac{\varepsilon}{2} = \frac{4}{3}\varepsilon \leq \frac{1}{3}r.$$

Therefore, we can use Lemma 3.38 to (Δ, β)-implement $\alpha \in \mathbb{C}_{\mathbb{A}}$ with $|\xi - \alpha| < \varepsilon/8$. We have

$$|\alpha - 1| \leq |\alpha - \xi| + |\xi - 1| < (1/8 + 4/3)\varepsilon < 2\varepsilon < 1.$$

Hence, $\mathrm{Re}(\alpha) > 0$ and we find that

$$\left|\frac{\xi-1}{\xi+1} - \frac{\alpha-1}{\alpha+1}\right| = 2\left|\frac{\xi-\alpha}{(\xi+1)(\alpha+1)}\right| < 2|\xi-\alpha| < \varepsilon/4.$$

Let $a = (\alpha-1)/(\alpha+1)$, $b = (\beta-1)^2/(\beta+1)^2 r/2$ and $z = (\xi-1)/(\xi+1) = ib/2$. The situation is plotted in Figure 3.6.

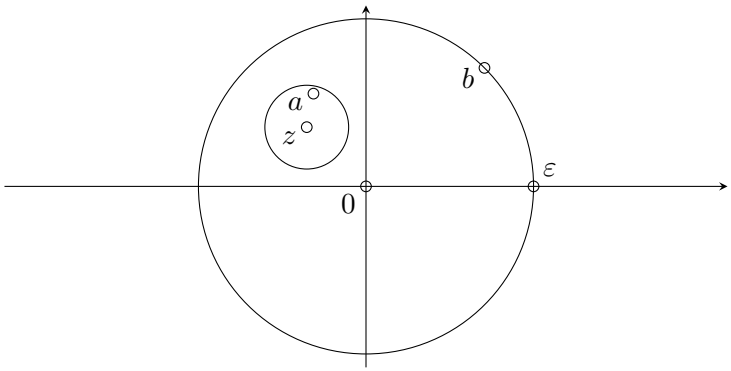


Figure 3.6: The quantities a, b, z in the proof of Lemma 3.3. We have $|b| = \varepsilon$, $z = ib/2$ and $|a - z| < \varepsilon/4$.

Let $\overline{xy} = \{\lambda(y-x) : \lambda \in \mathbb{R}\}$ for any $x, y \in \mathbb{C}$. Note that $\overline{0z}$ and $\overline{0b}$ are perpendicular, so 0 is the closest point of the line $\overline{0b}$ to z. Since $0 \notin B(z, \varepsilon/4)$, we conclude that $\overline{0b} \cap B(z, \varepsilon/4) = \emptyset$. In particular, a is not in the line $\overline{0b}$. Also note that $|a| < \varepsilon$ by the triangle inequality. Putting all this together with equation (3.13), we find that

$$\frac{\gamma-1}{\gamma+1} = -\frac{2}{r} b a^{-1} \notin \mathbb{R} \qquad \text{and} \qquad \left|\frac{\gamma-1}{\gamma+1}\right| = \frac{2}{r}\frac{\varepsilon}{|a|} > 1$$

as we wanted. We have shown how to (Δ, β)-implement $\gamma \in \mathbb{C}_{\mathbb{A}}$ with $\mathrm{Re}(\gamma) < 0$ and $\gamma \notin \mathbb{R}$. In particular, we have $|\gamma - 1|/|\gamma + 1| > 1/\sqrt{\Delta - 1}$. As a consequence of Theorem 1.7, the problems $\textsc{IsingNorm}(\Delta, \gamma, 1.01)$ and $\textsc{IsingArg}(\Delta, \gamma, \pi/3)$ are #P-hard. These problems reduce to $\textsc{IsingNorm}(\Delta, \beta, 1.01)$ and $\textsc{IsingArg}(\Delta, \beta, \pi/3)$ because we can (Δ, β)-implement γ, and the result follows. □

The rest of this section exploits Lemma 3.3 to obtain hardness for zeros of the partition function. Our approach uses a zero to implement -1 and conclude hardness with the help of Lemma 3.3.

Lemma 3.43. *Let Δ be an integer with $\Delta \geq 3$. Let $\beta \in \mathbb{C}_{\mathbb{A}} \setminus (\mathbb{R} \cup \{i, -i\})$. Suppose that there is a graph with maximum degree at most Δ having terminals s, t such that*

1. *the degree of s and t is at most $\Delta - 1$;*
2. *$Z_{\mathrm{Ising}}(G; \beta) = 0$;*
3. *$Z^{ij}_{st}(G; \beta) \neq 0$ for some $i, j \in \{0, 1\}$.*

Then $\textsc{IsingNorm}(\Delta, \beta, 1.01)$ and $\textsc{IsingArg}(\Delta, \beta, \pi/3)$ are #P-hard.

Proof. By symmetry of the spins 0 and 1 in the definition of Z_{Ising}, for any vertex v of G we have $Z^0_v(G; \beta) = Z^1_v(G; \beta)$. Let $i, j \in \{0, 1\}$ as in the statement. We obtain $0 = Z_{\mathrm{Ising}}(G; \beta) = 2Z^i_s(G; \beta)$ so

$$0 = Z^{i0}_{st}(G; \beta) + Z^{i1}_{st}(G; \beta). \tag{3.14}$$

Since either $Z^{i0}_{st}(G; \beta)$ or $Z^{i1}_{st}(G; \beta)$ is non-zero by hypothesis, both quantities are non-zero. Again, by symmetry of the spins 0 and 1, we have $Z^{00}_{st}(G; \beta) = Z^{11}_{st}(G; \beta)$ and $Z^{01}_{st}(G; \beta) = Z^{10}_{st}(G; \beta)$. Thus, by dividing by $Z^{01}_{st}(G; \beta)$ in (3.14) we find that

$$-1 = \frac{Z^{11}_{st}(G; \beta)}{Z^{01}_{st}(G; \beta)}.$$

We have shown that the graph G β-implements -1. Consider the graph H that is a copy of G with two extra vertices, s' and t', and two extra edges, (s, s') and (t, t'). By the properties of series compositions, see (3.4), the graph H β-implements $h_\beta(h_\beta(-1)) = -1$ for the terminals s' and t' (both of which have degree 1). Moreover, H has maximum degree at most Δ because G has maximum degree at most Δ and the vertices s and t have at most $\Delta - 1$ neighbours in G. We conclude that H (Δ, β)-implements -1, and hardness follows from Lemma 3.3. □

Corollary 3.44. *Let $\Delta = 3$. There is a $\beta \in \mathbb{C}_{\mathbb{A}} \setminus (\mathbb{R} \cup \{i, -i\})$ with $|\beta - 1|/|\beta + 1| < 1/\sqrt{\Delta - 1}$ such that $\textsc{IsingNorm}(\Delta, \beta, 1.01)$ and $\textsc{IsingArg}(\Delta, \beta, \pi/3)$ are #P-hard.*

Proof. Let us consider the graph G given in Figure 3.7 with distinguished vertices s and t.

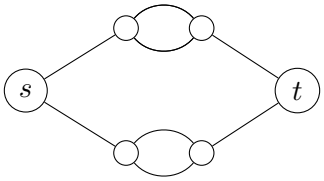


Figure 3.7: A graph that G with maximum degree 3 such that $Z_{\text{Ising}}(G;x)$ has a zero $\beta \in \mathcal{R}(1/\sqrt{2})$.

One can check that $Z^{01}_{st}(G;x) = (1+x^2+2x^3)^2$ and $Z^{11}_{st}(G;x) = x^2(2+x+x^3)^2$. We have $Z_{\text{Ising}}(G;x) = 2(1+6x^2+8x^3+2x^4+8x^5+6x^6+x^8)$. Using `Mathematica` we have determined that $Z_{\text{Ising}}(G;x)$ has a zero at $\beta \approx 0.396608 + 0.917988i$. Moreover, we have $|Z^{01}_{st}(G;\beta)| > 2$, so β and G satisfy the hypothesis of Lemma 3.43. We conclude that IsingNorm$(\Delta,\beta,1.01)$ and IsingArg$(\Delta,\beta,\pi/3)$ are #P-hard. Finally, we have $|\beta-1|/|\beta+1| < 1/\sqrt{2}$ since $|\beta-1|/|\beta+1| \approx 0.6572981$. □

We point out that one can use the approach that Buys developed for the independent set polynomial to find more zeroes inside the region $|\beta-1|/|\beta+1| \leq 1/\sqrt{\Delta-1}$ [25].

Let $\beta \in \mathbb{C} \setminus (\mathbb{R} \cup \{i,-i\})$. Lemma 3.43 uses the existence of a graph G with maximum degree at most Δ and $Z_{\text{Ising}}(G;\beta) = 0$ to demonstrate the hardness of IsingNorm$(\beta,\Delta,1.01)$ and IsingArg$(\beta,\Delta,\pi/3)$. However, Lemma 3.43 relies on the additional condition that $Z^{ij}_{st}(G;\beta) \neq 0$ for some $i,j \in \{0,1\}$ and two terminals s and t with degree at most $\Delta-1$. In the following conjecture, we conjecture that these additional conditions are not necessary.

Conjecture 3.4. *Let Δ be an integer with $\Delta \geq 3$ and let $\beta \in \mathbb{C}_{\mathbb{A}}$ with $\beta \notin \mathbb{R} \cup \{i,-i\}$. If there is a graph G with maximum degree at most Δ such that $Z_{\text{Ising}}(G;\beta) = 0$, then the problems* IsingNorm$(\beta,\Delta,1.01)$ *and* IsingArg$(\beta,\Delta,\pi/3)$ *are #P-hard.*

We make some progress on this conjecture in Corollary 3.45 (by changing the degree constraint from Δ to $\Delta-1$), but the full result seems to be out of reach for our implementation techniques.

Corollary 3.45. *Let Δ be an integer with $\Delta \geq 3$ and let $\beta \in \mathbb{C}_{\mathbb{A}} \setminus (\mathbb{R} \cup \{i,-i\})$. Suppose that there is a graph G of maximum degree at most $\Delta-1$ with $Z_{\text{Ising}}(G;\beta) = 0$. Then* IsingNorm$(\beta,\Delta,1.01)$ *and* IsingArg$(\beta,\Delta,\pi/3)$ *are #P-hard.*

Proof. Let $\mathcal{F} = \{G' : G' \text{ has maximum degree at most } \Delta-1 \text{ and } Z(G',\beta) = 0\}$, which is not empty by our hypothesis. We can choose $H \in \mathcal{F}$ with the minimum possible number of edges. Let $e = (s,t)$ be an edge of H. Let $H \setminus e$ be the graph obtained by deleting the edge e from H. We have

$$Z^{00}_{st}(H;\beta) = \beta Z^{00}_{st}(H \setminus e;\beta),$$
$$Z^{01}_{st}(H;\beta) = Z^{01}_{st}(H \setminus e;\beta).$$

Therefore, if $Z^{00}_{st}(H;\beta) = Z^{01}_{st}(H;\beta) = 0$, then $Z_{\text{Ising}}(H \setminus e;\beta) = 2Z^0_s(H \setminus e;\beta) = 2(Z^{00}_{st}(H;\beta) + Z^{01}_{st}(H;\beta)) = 0$, which contradicts the minimality of H. We conclude that either $Z^{00}_{st}(H;\beta) \neq 0$ or $Z^{01}_{st}(H;\beta) \neq 0$. Since s and t have degree at most $\Delta-1$, the result follows from Lemma 3.43. □

In [37], a parallel work to this chapter, a similar conjecture to Conjecture 3.4 has actually been shown for the independent set polynomial. A key element of the proof of [37] is establishing an analogous result to Lemma 3.3 for the independent set polynomial. In such a setting, a version of Lemma 3.3 turns out to be enough to prove that zeroes imply hardness: one can use an argument similar to the proof of Corollary 3.45, so a zero for the independent set polynomial implies that there is a tree that also achieves a zero and that implements -1 as a ratio. In the Ising case we are restricted by the fact that trees without pinnings do not implement any meaningful edge interactions due to the symmetry of the model and, thus, we can not use this trick to guarantee that there is a vertex in the graph H (see proof of Conjecture 3.4) such that H has two vertices with degree less than Δ.

3.6 Mobius-programs: proofs of Lemmas 3.30 and 3.31

In this section we prove Lemmas 3.30 and 3.31. These lemmas generalise the results on implementations for the independent set polynomial given in [15] to a more general setting so that they can be applied to other spin systems, including the Ising model. In fact, in a work published after we made public the results in this chapter, these results have been applied in the context of the Tutte polynomial [14]. Some of the definitions required in this section have been stated in Section 3.4.1, so we ask the reader to read Section 3.4.1 before this section. This section is organised as follows. In Section 3.6.1 we show how to generate approximations of any point around a program-approximable fixed point as a first step towards the proof of Lemma 3.30. In Section 3.6.2 we prove Lemma 3.30. Finally, in Section 3.6.3 we prove Lemma 3.31.

3.6.1 From program-approximable to densely program-approximable

In this section we generalise the results in [15, Section 7.2] on hardcore-programs to Mobius-programs. The main result of this section is Lemma 3.49, where we show that program-approximable fixed points are densely program-approximable under some hypothesis (see Section 3.4.1 for definitions). The main idea behind the results given in [15, Section 7] is that, locally around ω, hardcore-programs behave as *straight-line-programs*, which are much easier to study. This property is not specific to hardcore-programs, as illustrated in Lemma 3.47.

Definition 3.46. *Let $z \in \mathbb{C}$ with $z \neq 0$. A straight-line-program with operation*

$$(a_1, \dots, a_d) \mapsto z \sum_{j=1}^{d} a_j \tag{3.15}$$

is a sequence of assignments starting with $a_0 = 0, a_1 = 1$ and

$$a_k = z\left(a_{i_{k,1}} + \cdots + a_{i_{k,d}}\right), \quad \text{for } k = 2, 3, \dots,$$

where $i_{k,1}, \dots, i_{k,d} \in \{0, \dots, k-1\}$. We say that the straight-line-program generates $x \in \mathbb{C}$ if there exists integer $k \geq 0$ such that $a_k = x$.

Lemma 3.47 ([15, Lemma 7.9 for Mobius-programs])**.** *Let d be an integer with $d \geq 2$ and let g be a Mobius map. Let $\omega \in \mathbb{C}$ be a fixed point of $f(z) = g(z^d)$ with $\omega \neq 0$ and $g'(\omega^d), g''(\omega^d) \neq \infty$. Set $z := g'(\omega^d)\omega^{d-1}$. There exist reals $C_0 := C_0(g, d, \omega) > 1$ and $\delta_0 := \delta_0(g, d, \omega) > 0$ such that for any $a_1, \ldots, a_d \in \mathbb{C}$ with $|a_j| \leq \delta_0$ (for $j \in [d]$) we have*

$$g\left((\omega + a_1) \cdots (\omega + a_d)\right) = \omega + z\left(\sum\nolimits_{j=1}^{d} a_j\right) + \tau,$$

where $\tau \in \mathbb{C}$ with $|\tau| \leq C_0 \max_{j\in[d]} |a_j|^2$.

Proof. The proof is analogous to that of [15, Lemma 7.9]. The only difference is the determination of the constants C_0 and δ_0. Here these constants are obtained by a continuity argument whereas in [15, Lemma 7.9] C_0 and δ_0 are determined explicitly. We include this proof to illustrate this continuity argument. Let $b_1, \ldots, b_d \in \mathbb{C}$ with $|b_j| \leq 1$ for every $j \in [d]$. For $t \in \mathbb{R}$, we define

$$F(t) = g\left((\omega + tb_1) \cdots (\omega + tb_d)\right).$$

Note that $F(0) = g(\omega^d) = \omega$. To simplify our notation, for each $j \in [d]$, let $x_j(t) = \omega + tb_j$, and set $y(t) = x_1(t) \cdots x_d(t)$, so $F(t) = g(y(t))$. We have

$$F'(t) = g'\left(y\left(t\right)\right) \sum_{j=1}^{d} b_j \prod_{i=1,\, i\neq j}^{d} x_i(t).$$

In particular, we obtain

$$F'(0) = g'\left(\omega^d\right) \sum_{j=1}^{d} b_j \omega^{d-1} = z \sum_{j=1}^{d} b_j. \tag{3.16}$$

We have

$$F''(t) = g''\left(y\left(t\right)\right) \left(\sum\nolimits_{j=1}^{d} b_j \prod\nolimits_{\substack{i=1 \\ i\neq j}}^{d} x_i(t)\right)^2 + 2g'\left(y\left(t\right)\right) \sum_{1\leq j<i\leq d} b_j b_i \prod_{\substack{l=1 \\ l\neq i,j}}^{d} x_l(t). \tag{3.17}$$

Since $g'(\omega^d), g''(\omega^d) \neq \infty$ (by assumption) and $y(0) = \omega^d$, from the continuity of the maps y, g' and g'' we find that there is $\delta_0 := \delta_0(g, d, \omega) \in (0, 1)$ such that $g'(y(t))$ and $g''(y(t))$ are bounded when $|t| \leq \delta_0$. Note that $|x_j(t)| \leq |\omega| + 1$ when $|t| \leq \delta_0$. Therefore, (3.17) can be upper bounded when $|t| \leq \delta_0$ by a constant $C_0 := C_0(g, d, \omega) > 1$. By Taylor's formula we conclude that, for every $t \in \mathbb{R}$ with $|t| \leq \delta_0$,

$$\left|F(t) - F(0) - F'(t)t\right| \leq C_0 t^2. \tag{3.18}$$

Finally, let $a_1, \ldots, a_d$ with $|a_j| \leq \delta_0$. We choose $t = \max_{j\in[d]} |a_j|$. The result for $t = 0$ is equivalent to $F(0) = \omega$. Hence, we can assume that $t > 0$ and define $b_j = a_j/t$ for $j \in [d]$. The result then follows from $F(0) = \omega$, (3.16) and (3.18). □

Remark 3.48. *If the Mobius map g of Lemma 3.47 is given explicitly, then the constants $\delta_0(g, d, \omega)$ and $C_0(g, d, \omega)$ can be determined explicitly as it is done for $g(x) = 1/(1 + \lambda x)$ in the proof of [15, Lemma 7.9].*

As noted in [15, Section 7], straight-line-programs can generate evaluations of any polynomial $p(z)$ with positive coefficients, up to a factor z^n. This property of straight-line-programs is used in [15, Lemma 2.10] in conjunction with a density result on evaluations of polynomials to come up with hardcore-programs that generate approximations of any number near a fixed point of $f(x) = 1/(1 + \lambda x^d)$. Here we extend this result to Mobius-programs in Lemma 3.49. Apart from differences in notation, the proof is the same as that of [15, Lemma 2.10]; hence, we omit this proof and only highlight the notation differences. We also note that the difference between Lemmas 3.49 and 3.30 is that the latter gives an algorithm whereas the former only proves existence of these Mobius-programs.

Lemma 3.49 ([15, Lemma 2.10] for Mobius-programs)**.** *Let d be an integer with $d \geq 2$ and let g be a Mobius map. Let $f(x) := g(x^d)$ and let ω be a fixed point of f. Let us assume that the following assumptions hold.*

1. *ω is program-approximable for g and $a_0 \in \mathbb{C}$;*
2. *$\omega \neq 0$ and $g'(\omega^d), g''(\omega^d) \neq \infty$;*
3. *Let $z := f'(\omega)/d = g'(\omega^d)\omega^{d-1}$. We have $0 < |z| < 1$ and $z \notin \mathbb{R}$.*

Then, for any $\varepsilon, \kappa > 0$ there exists a radius $\rho \in (0, \kappa)$ such that the following holds. For every $x \in B(\omega, \rho)$ there is a Mobius-program for g starting at a_0 that generates a_k with $|x - a_k| \leq \varepsilon\rho$.

Proof. The proof is the same one as that of [15, Lemma 2.10] apart from a few differences in notation. Here we point out these differences in notation so that the reader can translate the proof to our setting if needed. First of all, in our version we have an arbitrary Mobius map g whereas [15, Lemma 2.10] sets $g(z) = 1/(1+\lambda z)$ for some activity $\lambda \in \mathbb{Q}_{\mathbb{C}} \setminus \mathbb{R}$ of the independent set polynomial. The particular choice of g does not affect the proof, so every instance of "hardcore-program" in [15, Lemma 2.10] can be effectively replaced by "Mobius-program for g and a_0", and every time that the proof invokes [15, Lemma 7.9] we can use the more general Lemma 3.47 instead.

The second main difference is that our statement adds a layer of generality in the choice of the fixed point $\omega \in \mathbb{C}$. In [15, Lemma 2.10] ω is chosen as the fixed point of $f(z) = 1/(1 + \lambda z^d)$ with the smallest norm. It turns out that such a fixed point satisfies the hypothesis of our statement. First, ω is program-approximable for $g(z) = 1/(1 + \lambda z)$ and $a_0 = \lambda$ (see [15, Lemma 2.7]). Secondly, we have ω and $g'(\omega^d), g''(\omega^d) \neq \infty$. Thirdly, $0 < |z| < 1$ and $z \notin \mathbb{R}$, see [15, Lemma 7.4] (we should point out that in [15] the authors set $z = \omega - 1$, which agrees with $z = g'(\omega^d)\omega^{d-1}$ for their choice of g). These are all the properties of ω needed to carry out the proof of [15, Lemma 2.10].

Finally, it is useful to note that if a hardcore-program generates a number a_k, then there is a tree of maximum degree at most $d + 1$ that implements λa_k. This explains why in the proof and statement of [15, Lemma 2.10] there is an extra factor λ when activities of the independent set polynomial are considered. Here we can omit this factor because we are not translating programs to gadgets. □

3.6.2 Proof of Lemma 3.30

In this section we translate the results given in [15, Section 7.3] to Mobius-programs. The main result of this section is Lemma 3.30, which gives an algorithmic version of Lemma 3.49. First, we need some technical results, Lemmas 3.50 and 3.51, which extend [15, Lemmas 7.10 and 7.11] to our more general setting.

Lemma 3.50 ([15, Lemma 7.10 for Mobius-programs])**.** *Let d be an integer with $d \geq 2$ and let g be a Mobius map. Let $\omega \in \mathbb{C}$ be a fixed point of $f(z) = g(z^d)$ with $\omega \neq 0$, $g'(\omega^d) \notin \{0, \infty\}$ and $g''(\omega^d) \neq \infty$. Set $z := g'(\omega^d)\omega^{d-1}$. There exist reals $C_1 := C_1(g, d, \omega) > 1$ and $\delta_1 := \delta_1(g, d, \omega) > 0$ such that for any $a_1, \ldots, a_d \in \mathbb{C}$ with $|a_j| \leq \delta_1$ (for $j \in [d]$) we have*

$$\Phi^{-1}(\omega + a_d) = \omega + \frac{a_d}{z} - \sum_{j=1}^{d-1} a_j + \tau,$$

where

$$\Phi(x) = g\left(x \prod_{j=1}^{d-1} (\omega + a_j)\right)$$

and $\tau \in \mathbb{C}$ with $|\tau| \leq C_0 \max_{j\in[d]} |a_j|^2$.

Proof. First, note that

$$\Phi^{-1}(x) = g^{-1}(x) \prod_{j=1}^{d-1} (\omega + a_j)^{-1}.$$

Let $b_1, \ldots, b_d \in \mathbb{C}$ with $|b_j| \leq 1$ for every $j \in [d]$. For $t \in (-|\omega|, |\omega|)$, note that $\omega + tb_j \neq 0$, so we can define

$$F(t) = g^{-1}(\omega + tb_d) \prod_{j=1}^{d-1} (\omega + tb_j)^{-1}.$$

We note that when g is particularised to $g(x) = 1/(1 + \lambda x)$, F coincides with the definition of F given in [15, Lemma 7.10]. Moreover, $F(t)$ agrees with $\Phi^{-1}(\omega + a_d)$ for $t = \max_{j\in\{1,\ldots,d\}}|a_j|$ and $b_j = a_j/t$. Note that $F(0) = g^{-1}(\omega)\omega^{-d+1} = \omega$. One can check that $F'(0) = b_d/z - \sum_{j=1}^{d-1} b_j$. The proof is now analogous to that of [15, Lemma 7.10], with the difference that the constants $C_1 := C_1(g, d, \omega) > 1$ and $\delta_1 := \delta_1(g, d, \omega) > 0$ are not explicitly determined but rather obtained by a continuity argument as in Lemma 3.47 that uses the hypotheses $\omega \neq 0$, $g'(\omega^d) \notin \{0, \infty\}$ and $g''(\omega^d) \neq \infty$. Hence, we do not repeat the rest of the proof here. □

Lemma 3.51 ([15, Lemma 7.11 for Mobius-programs])**.** *Let d be an integer with $d \geq 2$ and let g be a Mobius map. Let $\omega \in \mathbb{C}$ be a fixed point of $f(z) = g(z^d)$ with $g'(\omega^d), g''(\omega^d) \neq \infty$. Set $z := g'(\omega^d)\omega^{d-1}$. There exist reals $C_2 := C_2(g, d, \omega) > 1$ and $\delta_2 := \delta_2(g, d, \omega) > 0$ such that for any $a_1, \ldots, a_d \in \mathbb{C}$ with $|a_j| \leq \delta_2$ (for $j \in [d]$) we have*

$$\Phi'(\omega + a_d) = z + \tau,$$

where

$$\Phi(x) = g\left(x \prod_{j=1}^{d-1} (\omega + a_j)\right)$$

and $\tau \in \mathbb{C}$ *with* $|\tau| \leq C_0 \max_{j\in[d]} |a_j|$.

Proof. First, note that

$$\Phi'(x) = g'\left(x\prod_{j=1}^{d-1}(\omega + a_j)\right)\prod_{j=1}^{d-1}(\omega + a_j).$$

Let $b_1, \ldots, b_d \in \mathbb{C}$ with $|b_j| \leq 1$ for every $j \in [d]$. For $t \in \mathbb{R}$, we define

$$F(t) = g'\left(\prod_{j=1}^{d}(\omega + tb_j)\right)\prod_{j=1}^{d-1}(\omega + tb_j),$$

so $F(t)$ agrees with $\Phi'(\omega + a_d)$ for $t = \max_{j\in\{1,\ldots,d\}}|a_j|$ and $b_j = a_j/t$. Note that $F(0) = g'(\omega^d)\omega^{d-1} = z$. At this point the proof is analogous to that of Lemma 3.47, so we are not repeating it again. The only difference is that this time we have to bound $F'(t)$, instead of $F''(t)$, in a neighbourhood of $t = 0$, obtaining $\delta_2 := \delta_2(g, d, \omega) \in (0, 1)$ and $C_2 := C_2(g, d, \omega) > 1$ such that $|F'(t)| \leq C_2$ for all $t \in (-\delta_2, \delta_2)$. In [15, Lemma 7.11] the constants δ_2 and C_2 are made precise for the choice $g(x) = 1/(1 + \lambda x)$, whereas here we obtain them by continuity of $F'(t)$ and the fact that $g'(\omega^d), g''(\omega^d) \neq \infty$. □

Lemma 3.52 ([15, Lemma 7.12 for Mobius-programs])**.** *Let* d *be an integer with* $d \geq 2$ *and let* g *be a Mobius map. Let* $f(x) := g(x^d)$ *and let* ω *be a fixed point of* f*. Let us assume that the following assumptions hold.*

1. ω *is program-approximable for* g *and* $a_0 \in \mathbb{C}$*;*
2. $\omega \neq 0$ *and* $g'(\omega^d), g''(\omega^d) \notin \{0, \infty\}$*;*
3. *Let* $z := f'(\omega)/d = g'(\omega^d)\omega^{d-1}$*. We have* $0 < |z| < 1$ *and* $z \notin \mathbb{R}$*.*

Then there are Mobius-programs for g *starting at* a_0 *that generate* $\{\lambda_0, \lambda_1, \ldots, \lambda_t\} \subseteq \mathbb{C}$*, and a real* $r > 0$ *such that the following hold for all* $\hat{\omega} \in B(\omega, r)$*.*

1. *For* $i = 0$*,* $\lambda_0 \in B(\hat{\omega}, 2r)$*.*
2. *For* $i = 1, \ldots, t$*, the map* Φ_i *given by* $\Phi_i(x) = g(x\lambda_i\lambda_0^{d-2})$ *is contracting on the ball* $B(\hat{\omega}, 2r)$*.*
3. $B(\hat{\omega}, 2r) \subseteq \bigcup_{i=1}^{t}\Phi_i(B(\hat{\omega}, 2r))$*.*

Proof. The proof is exactly the same one as that of [15, Lemma 7.12] apart from the differences in notation mentioned in the proof of Lemma 3.49, and the fact that we use the more general Lemma 3.50 instead of [15, Lemma 7.10] and the more general Lemma 3.51 instead of [15, Lemma 7.11]. □

When one has maps $\Phi_1, \ldots, \Phi_t$ with the properties 2 and 3 of Lemma 3.52, there is an efficient algorithm to approximate numbers using sequential applications of the maps $\Phi_1, \ldots, \Phi_t$, see Lemma 3.53.

Lemma 3.53 ([15, Lemma 2.8])**.** *Let $z_0 \in \mathbb{C}_{\mathbb{A}}$, $r \in \mathbb{A}_{>0}$ and U be the ball $B(z_0, r)$. Further, suppose that $\Phi_1, \dots, \Phi_t$ are Mobius maps (with coefficients in $\mathbb{C}_{\mathbb{A}}$) that satisfy the following:*

1. *for each $i \in [t]$, Φ_i is contracting on the ball U,*
2. *$U \subseteq \bigcup_{i=1}^{t} \Phi_i(U)$.*

Then there is a polynomial-time algorithm which, on input (i) a starting point $x_0 \in U \cap \mathbb{C}_{\mathbb{A}}$, (ii) a target $x \in U \cap \mathbb{C}_{\mathbb{A}}$, and (iii) a rational $\varepsilon > 0$, outputs a number $\hat{x} \in U \cap \mathbb{C}_{\mathbb{A}}$ and a sequence $i_1, i_2, \dots, i_k \in [t]$ such that

$$\hat{x} = \Phi_{i_k}\left(\Phi_{i_{k-1}}\left(\cdots \Phi_{i_1}\left(x_0\right)\cdots\right)\right) \text{ and } |x - \hat{x}| \leq \varepsilon.$$

Even though [15, Lemma 2.8] is stated for particular maps Φ_i that arise in the context of the independent set polynomial, its proof is more general and works in the setting of Lemma 3.53. Now Lemma 3.30 follows from combining Lemmas 3.52 and 3.53.

Lemma 3.30 ([15, Proposition 2.6 for Mobius-programs])**.** *Let d be an integer with $d \geq 2$ and let g be a Mobius map with coefficients in $\mathbb{C}_{\mathbb{A}}$. Let $f(x) := g(x^d)$ and let ω be a fixed point of f. Let us assume that the following assumptions hold.*

1. *ω is program-approximable for g, d and $a_0 \in \mathbb{C}$;*
2. *$\omega \neq 0$, $g'(\omega^d) \notin \{0, \infty\}$ and $g''(\omega^d) \neq \infty$;*
3. *Let $z := f'(\omega)/d = g'(\omega^d)\omega^{d-1}$. We have $0 < |z| < 1$ and $z \notin \mathbb{R}$.*

Then ω is densely program-approximable in polynomial time for g, d and a_0.

Proof. The proof is the same as that of [15, Proposition 2.6], the main differences being that we invoke the more general Lemmas 3.52 and 3.53 instead of [15, Lemmas 7.12 and 2.8]. We also we stop the proof once we have obtained the desired program instead of translating the program to a gadget for the independent set polynomial. Finally, in the definition of densely program-approximable in polynomial time we ask the algorithm to compute k approximations $x_1, \dots, x_k$ of λ'. This can be done by running k versions of the algorithm given in the proof of [15, Proposition 2.6] and setting a different value for x_0 in each version when applying Lemma 3.53. In the proof of [15, Proposition 2.6] the value x_0 is a good approximation of the fixed point ω. These distinct values for x_0 are obtained by generating a better approximation of the fixed point ω each time. The generated elements $x_1, \dots, x_k$ will be of the form $\Phi_{i_j}\left(\Phi_{i_{j-1}}\left(\cdots \Phi_{i_1}\left(x_0\right)\cdots\right)\right)$, so all of them are distinct because the starting points for x_0 are distinct and the maps Φ_i are bijective. □

3.6.3 Proof of Lemma 3.31

In this section we prove Lemma 3.31, that is, we show how to generate approximations of any complex number with a Mobius-program. This generalises [15, Proposition 2.2] to Mobius-programs. Up to this point the results of [15] on hardcore-programs have been generalised to

Mobius-programs without much effort. In this section we have to refine the arguments given in the proof of [15, Proposition 2.2] to make it work for any Mobius map g, although the main idea stays the same: starting from a repelling fixed point and applying results of complex dynamics (see Section 3.2.4) to come up with an appropriate Mobius-program.

First, let us make some remarks about the proof of [15, Proposition 2.2]. This result shows how to efficiently implement approximations of any complex activity of the independent set polynomial via a hardcore-program. The proof is divided into three steps. First, the authors show how to generate approximations of any activity sufficiently large. Then they use the fact that $g(x) = 1/(1+\lambda x)$ tends to 0 when x diverges to generate approximations of any complex number near 0. Finally, they combine both results to generate an approximation of any complex number. Unfortunately, the second step breaks for arbitrary Mobius maps and, in particular, for the Mobius maps that we use when particularising these results to the Ising model. This motivates the work presented in this section.

This section is organised as follows. In Lemma 3.54 we show that if a repelling fixed point of $f(z) := g(z^d)$ is densely program-approximable in polynomial-time, then any complex point is densely program-approximable in polynomial-time. In particular, this includes the point 0 that escapes from the arguments given in [15], which rely heavily on the fact that complex holomorphic maps are locally Lipschitz. Instead of this local property, here we use the fact that rational maps are Lipschitz on the Riemann sphere with respect to the chordal metric (Lemma 3.11), which simplifies the proofs because we do not have to deal so carefully with the poles of the rational map. In Lemma 3.55 we show how to generate approximations of any complex number that is sufficiently large. Although Lemma 3.55 could follow from the technical proof given in [15, Proposition 2.2], we include a simpler proof that goes along the same lines as the proof of Lemma 3.54. Finally, we combine Lemmas 3.54 and 3.55 to prove Lemma 3.31.

Lemma 3.54. *Let d be an integer with coefficients in $\mathbb{C}_{\mathbb{A}}$. Let $\omega \in \mathbb{C}$ be a repelling fixed point of $f(z) := g(z^d)$ that is densely program-approximable in polynomial time for g and $a_0 \in \mathbb{C}_{\mathbb{A}}$. Let E_f be the exceptional set of the rational map f and let $\gamma \in \mathbb{C} \setminus E_f$. Then γ is densely program-approximable in polynomial time for g and a_0.*

Proof. Let $r_\omega > 0$ from the definition of densely program-approximable point for ω (Definition 3.29). Since ω is a repelling fixed point of f, by Lemma 3.12 it belongs to the Julia set of f and, thus, by Theorem 3.13, $\bigcup_{n=0}^{\infty} f^n(B(\omega, r_\omega)) = \widehat{\mathbb{C}} \setminus E_f$. Let N be the smallest non-negative integer such that $\gamma \in f^N(B(\omega, r_\omega))$. If $N = 0$, then the fact that γ is densely program-approximable in polynomial time for g and a_0 is trivial: let r_γ be any positive real number with $B(\gamma, r_\gamma) \subseteq B(\omega, r_\omega)$ and use the algorithm from the definition of densely program-approximable (in polynomial time) point for ω on the inputs $\lambda' \in B(\gamma, r_\gamma) \cap \mathbb{C}_{\mathbb{A}}$ and $\varepsilon > 0$ rational. In the rest of the proof we deal with the case $N \geq 1$.

Let $x \in B(\omega, r_\omega)$ such that $f^N(x) = \gamma \in \mathbb{C}$. Let

$$\mathcal{P} = \{z \in \mathbb{C} : z \text{ is a pole of } f^n \text{ for some } n \in [N]\}.$$

Note that $\mathcal{P}$ is a finite set so there is $r > 0$ such that

$$\overline{B}(x, r) \subseteq B(\omega, r_\omega) \qquad \text{and} \qquad \overline{B}(x, 2r) \cap \mathcal{P} \subseteq \{x\}. \tag{3.19}$$

We point out here that $x \in \mathcal{P}$ if and only if there is $n \in [N-1]$ such that $f^n(x) = \infty$. Since $f^N(\overline{B}(x, 2r))$ is a compact set of complex numbers (f^N is continuous on $\overline{B}(x, 2r)$ as a complex function due to the lack of poles), there is a rational constant $C > 0$ depending on γ such that

$$|f^N(z)| \leq C \quad \text{for every } z \in \overline{B}(x, 2r). \tag{3.20}$$

Since f^N is a rational function, $f^N(B(x, r))$ is an open set in $\widehat{\mathbb{C}}$ by the open mapping theorem (Proposition 3.8). Hence, there is a rational $r_\gamma > 0$ with $B(\gamma, r_\gamma) \subseteq f^N(B(x, r))$ (here we used that $\gamma \neq \infty$). This is the radius in the definition of densely program-approximable point for γ (Definition 3.29). By Lemma 3.11, there is a rational $L \geq 1$ such that f^N is Lipschitz with constant L in $\widehat{\mathbb{C}}$ with respect to the chordal metric.

Now we proceed to give the polynomial-time algorithm. Let k be a positive integer. Let $\lambda \in B(\gamma, r_\gamma) \cap \mathbb{C}_{\mathbb{A}}$ and $\varepsilon > 0$ rational be the inputs of the algorithm. We are going to compute k elements of Mobius-programs that approximate λ up to an error ε. We set

$$\varepsilon' = \frac{\varepsilon}{1 + C^2} \qquad \text{and} \qquad \varepsilon'' = \min\left\{\varepsilon'/L, r\right\}. \tag{3.21}$$

We can write $f^N(z) = P(z)/Q(z)$ where P and Q are polynomials with coefficients in $\mathbb{C}_{\mathbb{A}}$. Note that the equation $P(z)/Q(z) = \lambda$ in $z \in B(x, r) \cap \mathbb{C}_{\mathbb{A}}$ is equivalent to $P(z) = \lambda Q(z)$ in $z \in B(x, r) \cap \mathbb{C}_{\mathbb{A}}$ because Q has no zeros in $B(x, r)$. We can solve this polynomial equation numerically as described in the proof of [15, Proposition 2.2, case I] to compute $x' \in B(x, r) \cap \mathbb{C}_{\mathbb{A}}$ with $|x^* - x'| \leq \varepsilon''/2$ for some solution x^* of $P(z) = \lambda Q(z)$ with $x^* \in B(x, r) \cap \mathbb{C}_{\mathbb{A}}$, so $f^N(x^*) = \lambda$. Since $x' \in B(x, r) \cap \mathbb{C}_{\mathbb{A}} \subseteq B(\omega, r_\omega)$, we can use the algorithm of the definition of densely program-approximable in polynomial time for ω to compute $k + 1$ distinct elements $\hat{x}_1, \ldots, \hat{x}_{k+1}$ of a Mobius-program for g and a_0 with $|x' - \hat{x}_j| \leq \varepsilon''/2$. Let $\hat{x}$ be any of these $k + 1$ elements and let us analyse how close $f^N(\hat{x})$ is to λ. We have $|x^* - \hat{x}| \leq \varepsilon''$ by the triangle inequality. We claim that $|\lambda - f^N(\hat{x})| \leq \varepsilon$. In view of the Lipschitz property of f^N and (3.21), we have

$$d\left(\lambda, f^N\left(\hat{x}\right)\right) \leq L d\left(x^*, \hat{x}\right) = L \frac{|x^* - \hat{x}|}{\left(\left(1 + |x^*|^2\right)\left(1 + |\hat{x}|^2\right)\right)^{1/2}} \leq L\,|x^* - \hat{x}| \leq L\varepsilon'' \leq \varepsilon'.$$

Note that by the triangle inequality, $|x - \hat{x}| \leq |x - x'| + |x' - \hat{x}| \leq r + \varepsilon'' \leq 2r$. We can now use the upper bound (3.20) with $z = \hat{x}$ and $z = x^*$ to conclude that $|f^N(x^*)| = |\lambda| \leq C$ and

$$\left|\lambda - f^N\left(\hat{x}\right)\right| = \left(\left(1 + |f^N\left(\hat{x}\right)|^2\right)\left(1 + |\lambda|^2\right)\right)^{1/2} d\left(\lambda, f^N\left(\hat{x}\right)\right) \leq \left(1 + C^2\right) d\left(\lambda, f^N\left(\hat{x}\right)\right) = \varepsilon.$$

The algorithm chooses k numbers in $\{f^N\left(\hat{x}_1\right), \ldots, f^N\left(\hat{x}_{k+1}\right)\}$ in conjunction with its representation as a sequence of tuples (which comes from the representation of $\hat{x}$ and the corresponding applications of f). Recall that N does not depend on the inputs ε and λ, so computing $f^N\left(\hat{x}\right)$ from $\hat{x}$ only adds a constant factor to the running time. In order to conclude the proof, we have

to ensure that $f^1(\hat{x}), f^2(\hat{x}), \ldots, f^N(\hat{x})$ is a sequence of complex numbers, that is, ∞ does not appear in the sequence. It is enough to show that this is the case for at least k of the $k+1$ outputs $f^N(\hat{x}_1), \ldots, f^N(\hat{x}_{k+1})$, since we only wanted k outputs to begin with. This follows from (3.19) and the fact that at most one of the numbers $\hat{x}_1, \ldots, \hat{x}_{k+1}$ is equal to x. □

Lemma 3.55 ([15, Proposition 2.2, case I])**.** *Let d be an integer with $d \geq 2$ and let g be a Mobius map with coefficients in $\mathbb{C}_{\mathbb{A}}$. Let $\omega \in \mathbb{C}$ be a repelling fixed point of $f(z) := g(z^d)$ that is densely program-approximable for g and $a_0 \in \mathbb{C}_{\mathbb{A}}$. Let E_f be the exceptional set of the rational map f. If $\infty \notin E_f$, then there exists a rational number $M > 1$ such that the following holds.*

There is a polynomial-time algorithm such that, on input $\lambda \in \mathbb{C}_{\mathbb{A}}$ with $|\lambda| > M$ and rational $\varepsilon > 0$, computes an element a_k of a Mobius-program for g starting at a_0 with $|\lambda - a_k| \leq \varepsilon$.

Proof. This lemma can be proven following the argument given in the first case of the proof of [15, Proposition 2.2] for the independent set polynomial. Here we give a simpler proof that works even when g is just a rational function. The original proof is significantly more technical because the authors first get close to a pole and then apply one more iteration of f to get near the desired point λ. To do this, they have to make sure that the poles of $f^1, \ldots, f^N$ are excluded from the all the domains considered and that all the applications of f are locally Lipschitz.

Our proof follows the same structure as that of Lemma 3.54 with $\gamma = \infty$, but it requires a slightly different analysis because in the proof x is a pole of f^N. Let $r_\omega > 0$ from the definition of densely program-approximable fixed point for ω. Since ω is a repelling fixed point of f, by Lemma 3.12 it belongs to the Julia set of f and, thus, by Theorem 3.13, $\bigcup_{n=0}^{\infty} f^n(B(\omega, r_\omega)) = \widehat{C} \setminus E_f$. Since $\infty \notin E_f$, we can consider the smallest non-negative integer N such that $\infty \in f^N(B(\omega, r_\omega))$. Note that $N \geq 1$ because $\infty \notin B(\omega, r_\omega)$.

Let $x \in B(\omega, r_\omega)$ such that $f^N(x) = \infty$. Let

$$\mathcal{P} = \{z \in \mathbb{C} : z \text{ is a pole of } f^n \text{ for some } n \in [N]\}$$

and

$$\mathcal{Z} = \{z \in \mathbb{C} : z \text{ is a zero of } f^n \text{ for some } n \in [N]\}.$$

Note that $\mathcal{P}$ and $\mathcal{Z}$ are finite sets and $x \in \mathcal{P}$ because $f^N(x) = \infty$. Let δ be the minimum distance between any two distinct numbers in $\mathcal{P} \cup \mathcal{Z}$. There is $0 < r < \delta/4$ such that

$$\overline{B}(x, r) \subseteq B(\omega, r_\omega) \qquad \text{and} \qquad \overline{B}(x, 2r) \cap (\mathcal{P} \cup \mathcal{Z}) = \{x\}. \tag{3.22}$$

Since f^N is a rational function, $f^N(B(x, r))$ is an open set in $\widehat{\mathbb{C}}$, see Section 3.2.4. Hence, by the topology of the Riemann sphere and the fact that $\infty \in f^N(B(x, r))$, there is $M > 1$ with $U = \{z \in \mathbb{C} : |z| > M\} \subseteq f^N(B(x, r))$. This is the constant given in the statement. We can write $f^N(z) = P(z)/Q(z)$ where P and Q are polynomials with coefficients in $\mathbb{C}_{\mathbb{A}}$ that do not share any root, so the set of roots of P is $\mathcal{Z}$ and the set of roots of Q is $\mathcal{P}$. We are going to bound $|P(z)|$ and $|Q(z)|$ in $B(x, r)$. First, let us bound $|P(z)|$. Let k be the multiplicity of the

pole x of f^N. For any $z \in \overline{B}(x, 2r)$ and $\zeta \in \mathcal{Z} \cup \mathcal{P}$ with $\zeta \neq x$, in view of $r < \delta/4$ and (3.22), we have

$$|z - \zeta| \leq |x - \zeta| + |z - x| \leq |x - \zeta| + 2r \leq 2r + \max_{\zeta' \in \mathcal{Z} \cup \mathcal{P}} \left|x - \zeta'\right|$$

and

$$|z - \zeta| \geq |x - \zeta| - |x - z| \geq \delta - 2r \geq \delta/2.$$

Hence, there are real numbers $C_0, C_1 > 0$ such that, for any $z \in \overline{B}(x, 2r)$,

$$C_0(\delta/2)^{\deg P} \leq |P(z)| \leq C_1 \quad \text{and} \quad C_0(\delta/2)^{(\deg Q)-k} |x - z|^k \leq |Q(z)| \leq C_1 |x - z|^k . \tag{3.23}$$

Combining the bounds in (3.23) and setting $D_0 = C_0(\delta/2)^{\deg P}/C_1$ and $D_1 = C_1(\delta/2)^{k-(\deg Q)}/C_0$ we find that, for any $z \in \overline{B}(x, 2r)$,

$$D_0 |x - z|^{-k} \leq \left|f^N(z)\right| \leq D_1 |x - z|^{-k} . \tag{3.24}$$

Note that D_0 and D_1 are positive. The bounds (3.24) play an important role in our algorithm. We also need Lemma 3.11, which gives a rational $L \geq 1$ such that f^N is Lipschitz with constant L in $\widehat{\mathbb{C}}$.

Now we proceed to give the polynomial-time algorithm. Let $\lambda \in \mathbb{C}_{\mathbb{A}}$ with $|\lambda| > M$ and $\varepsilon > 0$ rational be the inputs. We set $\tau = \frac{D_1}{D_0} 2^k |\lambda|$ and

$$\varepsilon' = \frac{\varepsilon}{((1 + |\tau|^2)(1 + |\lambda|^2))^{1/2}} \qquad \text{and} \qquad \varepsilon'' = \min\left\{\varepsilon'/L, r, \frac{1}{2}(D_0/|\lambda|)^{1/k}\right\}. \tag{3.25}$$

Note that $\text{size}(\varepsilon'') = \text{poly}(\text{size}(\varepsilon), \text{size}(\lambda))$ since L, D_0, D_1, r, k are constants that do not depend on the inputs. The equation $P(z)/Q(z) = \lambda$ in $z \in B(x, r) \cap \mathbb{C}_{\mathbb{A}}$ is equivalent to $P(z) = \lambda Q(z)$ in $z \in B(x, r) \cap \mathbb{C}_{\mathbb{A}}$ because Q has no zeros in $B(x, r)$ other than x. We can solve this polynomial equation numerically as described in the proof of [15, Proposition 2.2, case I] to compute $x' \in B(x, r) \cap \mathbb{C}_{\mathbb{A}}$ with $|x^* - x'| \leq \varepsilon''/2$ for some solution x^* of $P(z) = \lambda Q(z)$ with $x^* \in B(x, r) \cap \mathbb{C}_{\mathbb{A}}$, so $f^N(x^*) = \lambda$. Since $x' \in B(x, r) \cap \mathbb{C}_{\mathbb{A}} \subseteq B(\omega, r_\omega)$, we can use the algorithm of the definition of densely program-approximable in polynomial time for ω to compute 2 distinct elements $\hat{x}_1, \hat{x}_2$ of a Mobius-program for g and a_0 with $|x' - \hat{x}_j| \leq \varepsilon''/2$ for any $j \in \{1, 2\}$. By the triangle inequality, we have $|x^* - \hat{x}_j| \leq \varepsilon''$ and $|x - \hat{x}_j| \leq |x - x^*| + |x^* - \hat{x}_j| \leq r + \varepsilon'' \leq 2r$, where we used (3.25). In light of the choice of r in (3.22), for any $z \in \overline{B}(x, 2r)$, we have $f^N(z) = \infty$ if and only if $z = x$. Hence, we can check if $\hat{x}_j$ is x by evaluating $f^N(\hat{x}_j)$ and checking if the result is ∞ or not. Since $\hat{x}_1$ and $\hat{x}_2$ are distinct, at least one of the two is not x, so we can pick $\hat{x} \in \{\hat{x}_1, \hat{x}_2\}$ with $\hat{x} \neq x$. We work with $\hat{x}$ in the rest of the proof. We claim that $|\lambda - f^N(\hat{x})| \leq \varepsilon$. Recall that $x^*, \hat{x} \in \overline{B}(x, 2r)$. In view of the bounds (3.24) and (3.25) we have

$$\frac{1}{2}|x - x^*| \geq \frac{1}{2}\left(\frac{D_0}{|f^N(x^*)|}\right)^{1/k} = \frac{1}{2}\left(\frac{D_0}{|\lambda|}\right)^{1/k} \geq \varepsilon'',$$

so $|x - \hat{x}| \geq |x - x^*| - |x^* - \hat{x}| \geq |x - x^*| - \varepsilon'' \geq |x - x^*|/2$ and

$$\left|f^N(\hat{x})\right| \leq \frac{D_1}{|x - \hat{x}|^k} \leq D_1 \left(\frac{2}{|x - x^*|}\right)^k \leq \frac{D_1}{D_0} 2^k |\lambda| = |\tau|. \tag{3.26}$$

In view of the Lipschitz property of f^N and (3.25), we have

$$d\left(\lambda, f^N\left(\hat{x}\right)\right) \leq Ld\left(x^*, \hat{x}\right) = L\frac{|x^* - \hat{x}|}{\left(\left(1+|x^*|^2\right)\left(1+|\hat{x}|^2\right)\right)^{1/2}} \leq L\left|x^* - \hat{x}\right| \leq L\varepsilon'' \leq \varepsilon',$$

which in combination with (3.26) yields

$$\left|\lambda - f^N\left(\hat{x}\right)\right| = \left(\left(1+\left|f^N\left(\hat{x}\right)\right|^2\right)\left(1+|\lambda|^2\right)\right)^{1/2} d\left(\lambda, f^N\left(\hat{x}\right)\right) \leq \left(\left(1+|\tau|^2\right)\left(1+|\lambda|^2\right)\right)^{1/2}\varepsilon' = \varepsilon$$

as we wanted to prove. The algorithm outputs $f^N\left(\hat{x}\right)$ in conjunction with its representation as a sequence of tuples (which comes from the representation of $\hat{x}$ and the corresponding applications of f). In order to conclude the proof, we have to guarantee that the sequence $f^1(\hat{x}), f^2(\hat{x}), \ldots, f^N(\hat{x})$ does not contain the point ∞. In light of (3.22), $\hat{x}$ is a pole of f^n for some $n \in [N]$ if and only if $\hat{x} = x$. But we chose $\hat{x} \in \{\hat{x}_1, \hat{x}_2\}$ with $\hat{x} \neq x$, concluding the proof. □

Now we can combine Lemmas 3.54 and 3.55 to prove Lemma 3.31. The proof follows the same idea as that of [15, Proposition 2.2, case I] with the difference that we can not use the particular shape of g to simplify some steps. Hence, we have again to use the Lipschitz property of rational functions on the Riemann sphere (Lemma 3.11).

Lemma 3.31 ([15, Proposition 2.2 for Mobius-programs])**.** *Let d be an integer with $d \geq 2$ and let g be a Mobius map with coefficients in $\mathbb{C}_{\mathbb{A}}$ such that $g(\infty) \in \mathbb{C}$. Let $\omega \in \mathbb{C}$ be a repelling fixed point of $f(z) := g(z^d)$ that is densely program-approximable in polynomial time for g and $a_0 \in \mathbb{C}_{\mathbb{A}}$. Let E_f be the exceptional set of the rational map f. If $0, \infty \notin E_f$, then the following holds.*

There is a polynomial-time algorithm such that, on input $\lambda \in \mathbb{C}_{\mathbb{A}}$ and rational $\varepsilon > 0$, computes an element a_k of a Mobius-program for g starting at a_0 with $|\lambda - a_k| \leq \varepsilon$.

Proof. By Lemma 3.11 there is a rational $L \geq 1$ such that f^N is Lipschitz with constant L in $\widehat{\mathbb{C}}$. Note that g only has one complex pole because g is a Mobius map with $g(\infty) \in \mathbb{C}$. Let $p \in \mathbb{C}$ be the pole of g, so $g(p) = \infty$. We can write g as $g(z) = a/(z-p) + b$ for some $a, b \in \mathbb{C}_{\mathbb{A}}$ with $a \neq 0$. Let $r_0 \in (0,1)$ be rational number as in Lemma 3.54 for $\gamma = 0$. Let $M > 1$ be a rational number as in Lemma 3.55. There is a rational $r > 0$ such that

$$|g(z)| > 2M \qquad \text{for all } z \in B(p, r). \tag{3.27}$$

Since $g(\infty) = b \in \mathbb{C}$ and p is the only pole of g, we find that g is bounded on $\mathbb{C} \setminus B(p, r/2)$. Let C be a positive real number with

$$|g(z)| \leq C \qquad \text{for all } z \in \mathbb{C} \setminus B(p, r/2). \tag{3.28}$$

We can now specify the algorithm announced in the statement. Let $\lambda \in \mathbb{C}_{\mathbb{A}}$ and $\varepsilon > 0$ rational be its inputs. Our algorithm distinguishes two cases depending on $\lambda \in B(p, r_p)$.

1. $|\lambda| > M$. Then we can use the algorithm of Lemma 3.55 to compute an element a_k of a Mobius-program for g starting at a_0 with $|\lambda - a_k| \leq \varepsilon$.

2. $|\lambda| \leq M$. Let $x^* = g^{-1}(\lambda)$. We have $x^* \notin B(p, r)$ because $|g(x^*)| = |\lambda| < 2M$, see (3.27). Let
$$\varepsilon' = \min\left\{\frac{\varepsilon}{L(1+C^2)}, r/2\right\} \tag{3.29}$$
Our algorithm computes x^* and distinguish two more cases depending on x^*. In each of the two cases the algorithm is going to compute an element $\hat{x}$ of a Mobius-program for g starting at a_0 with $|x^* - \hat{x}| \leq \varepsilon'$.

   - $|x^*| > M$. Our algorithm uses the algorithm of Lemma 3.55 with inputs x^* and ε' to compute an element $\hat{x}$ of a Mobius-program for g starting at a_0 with $|x^* - \hat{x}| \leq \varepsilon'$, and returns a_k.
   - $|x^*| \leq M$. Note that $2M/r_0 > M$. Our algorithm first uses the algorithm of Lemma 3.55 with inputs $\lambda = 2M/r_0$ and $\varepsilon = 1$ to compute an element λ_4 of a Mobius-program for g starting at a_0 with $|2M/r_0 - \lambda_4| \leq 1$, so $|\lambda_4| > M/r_0$. This step takes constant time since all the quantities involved are constants stored in our algorithm. The idea for this part of the proof is borrowed from [15, Proposition 2.2, case III]. Here λ_4 plays the same role as the activity λ_4 implemented in [15, Proposition 2.2, case III], hence the choice of the name. We have $|x^*/\lambda_4| < r_0$, so we can use the algorithm of Lemma 3.54 for $\gamma = 0$ with inputs x^*/λ_4 and $\varepsilon'/|\lambda_4|$ to compute an element $\hat{y}$ of a Mobius-program for g starting at a_0 with $|x^*/\lambda_4 - \hat{y}| \leq \varepsilon'/|\lambda_4|$. We set $\hat{x} = \lambda_4\hat{y}$ and note that $|x^* - \hat{x}| \leq \varepsilon'$.

   From the definition of ε' (3.29) and the triangle inequality we have
$$|p - \hat{x}| \geq |p - x^*| - |x^* - \hat{x}| \geq r - \varepsilon' \geq r/2.$$
   Hence, (3.28) yields
$$|g(\hat{x})| \leq C. \tag{3.30}$$
   In view of the Lipschitz property of f^N we have
$$d(\lambda, g(\hat{x})) \leq Ld\left(g^{-1}(\lambda), \hat{x}\right) = L\frac{|x^* - \hat{x}|}{\left((1+|x^*|^2)(1+|\hat{x}|^2)\right)^{1/2}} \leq L|x^* - \hat{x}| \leq L\varepsilon' \leq \frac{\varepsilon}{L(1+C^2)}$$
   which in combination with (3.30) and $|\lambda| \leq M \leq C$ yields
$$|\lambda - g(\hat{x})| = \left(\left(1+|g(\hat{x})|^2\right)\left(1+|\lambda|^2\right)\right)^{1/2} d(\lambda, g(\hat{x})) \leq L(1+C^2)d(\lambda, g(\hat{x})) \leq \varepsilon.$$
   Our algorithm returns the representation of $g(\hat{x})$ as an element of a Mobius-program. □

Remark 3.56. *The hypothesis $g(\infty) \in \mathbb{C}$ can be removed from Lemma 3.31. This would require us to study the case $g(\infty) = \infty$ in the proof of Lemma 3.31. Note that $g(z) = \infty$ if and only if g is of the form $az + b$ for $a, b \in \mathbb{Q}$ with $a \neq 0$. This case is not relevant for this work, hence why we left it out of the statement. Also, for convenience, [15] restricted attention to complex numbers whose real and imaginary parts are rational, but it suffices for them to be algebraic.*

Chapter 4

Fast sampling of satisfying assignments from random k-SAT

◦ This chapter is based on the following publication:
Andreas Galanis, Leslie Ann Goldberg, Heng Guo, and Andrés Herrera-Poyatos. Fast sampling of satisfying assignments from random k-sat. *arXiv preprint*, 2022. arXiv:2206.15308
◦ This chapter also includes results on the geometry of the space of satisfying assignments of random k-CNF formulas, that have been developed in conjunction with Zongchen Chen, Nitya Mani and Ankur Moitra. The proofs of these geometry results presented here are my own. An extended version of [48] containing these geometry results has been submitted to Random Structures and Algorithms.

Organisation of this chapter

In this chapter we introduce our almost-uniform sampler for satisfying assignments of random k-CNF formulas and prove its correctness. This chapter is organised as follows. First, in Section 4.1 we describe our algorithm and provide an outline of our proof. Given the amount of notation and background needed to state and explain our algorithms and main technical lemmas, the preliminary material is presented across the proof outline, as opposed to previous chapters. Our proofs are then split into 9 sections, proving our main theorem on the correctness of our sampling algorithm in Section 4.7 and our results on the geometry of the space of satisfying assignments of random k-CNF formulas in Section 4.8. Since the titles and content of these sections are technical and require some notation and definitions before introducing them, we provide a more detailed organisation of this chapter at the end of the proof outline, see Section 4.1.5.

4.1 Proof outline and preliminaries

Our nearly linear-time sampling algorithm is based on running a Markov chain; this is a standard technique in approximate counting, where typically one runs a Markov chain on the whole state space that converges to the desired distribution. The twist in k-SAT is that the state space of the Markov chain needs to be carefully selected in order to avoid certain bottleneck phenomena that impede fast convergence. This approach has been recently applied to bounded-degree k-CNF formulae [43, 75, 44, 74] building on the work of Moitra [93] and using the Markov chain known as single-site Glauber dynamics. The main difficulties in all of these works are that the usual

distribution properties that are typically used to obtain fast algorithms (such as correlation decay and spatial mixing) fail on the set of all SAT solutions, and in fact even ensuring a connected state space is a major problem. Working around this is one of the main challenges for us too, and in the random k-SAT setting it is further aggravated by the fact that a linear number of variables have degrees much higher than average. In fact, w.h.p., a good portion of vertices have degrees depending on n, with the maximum degree of the formula scaling as $(\log n)/(\log\log n)$; this can be shown by analysing a bins and balls experiments where variables identify with bins and you throw $k\lfloor \alpha n \rfloor$ balls (one for each literal of the random formula), see [92, Chapter 5] for details.

This poses several new challenges for the Markov chain approach to work in our setting. First of all, we have to ensure that the set of satisfying assignments that our Markov chain considers has good connectivity properties. We address this problem in Section 4.1.1 of this proof outline, where we find a suitable subset of marked variables where we can run the Glauber dynamics; this part is inspired by Moitra's "marking" approach, though here we need to add an extra layer of marking to facilitate later the analysis of the Markov chain. Second and more importantly, state-of-the-art arguments for bounding the mixing time of the single-site Glauber dynamics on k-CNF formulae, such as [75, 43] break under the presence of high-degree variables. We focus on this in Section 4.1.2, where we outline a novel argument that analyses the mixing time of the uniform-block Glauber dynamics using recent advances in spectral independence [5, 79, 7, 28]. This is the first application of the spectral-independence framework for k-CNF formulae, where the absence of correlation decay limits the application of standard techniques (based on self-avoiding walk trees [7, 28]). A reader unfamiliar with spectral independence is encouraged to read Section 4.1.2.1 before continuing reading this proof outline. To obtain our spectral-independence bounds we need to combine the probabilistic structure of satisfying assignments with the local sparsity properties of the random formula. The third challenge in our approach is simulating the individual steps of the uniform-block Glauber dynamics since they involve updating a linear number of variables, making the computation of the transition probabilities more challenging. To this end, we need to initialise our block Glauber dynamics to random values (instead of an arbitrary assignment that is typically used as initialisation), and show that the formula breaks into small tree-like connected components that allows us to do the relevant computations throughout the algorithm's execution (cf. Section 4.1.3). Based on these pieces, the full algorithm is presented in Section 4.1.4.

The fact that the formula breaks into small tree-like connected components when marked variables are assigned random values will also allow to analyse the geometry of the space of satisfying assignment of the random formula, and we will delve into this connection in Section 4.1.3.

4.1.1 Marking variables in the random k-SAT model

In order to ensure good connectivity properties which are essential for fast convergence of the relevant Markov chain, our algorithm runs Glauber dynamics on a large subset $\mathcal{V}_{\mathrm{m}}$ of so-called "marked" variables of the random formula, leaving the rest of the variables unassigned. The variables in $\mathcal{V}_{\mathrm{m}}$ are chosen in a way that ensures that their marginals are near $1/2$, which is important for ensuring rapid mixing. Moitra [93] introduced a random "marking" procedure to identify such a subset of variables in the bounded-degree case. The presence of high-degree variables impedes a direct application of this technique in the random-formula setting, but in [49] the authors show that by temporarily removing a small linear number of "bad" clauses that contain high-degree variables, one can also achieve marginals near $1/2$ for an appropriate set of variables in the random k-SAT model. Here, we further refine these arguments, as we need more control over the high-degree variables of the formula in order to conclude rapid mixing of the Glauber dynamics. Recall that the degree of a variable v is the number of occurrences of literals involving the variable v in Φ and that the maximum degree of the formula Φ is the maximum degree among its variables. The following important definitions will be used throughout the paper. We usually use $\mathcal{V}$ to denote the set of variables and $\mathcal{C}$ to denote the set of clauses of a k-CNF formula Φ. For any $c \in \mathcal{C}$ we denote by $\mathrm{var}(c)$ the set of variables appearing in c, and for any $S \subseteq \mathcal{C}$ we denote $\mathrm{var}(S) = \bigcup_{c \in S} \mathrm{var}(c)$.

Definition 4.1 (high-degree, Δ_r)**.** *Let $r \in (0,1)$ and let $k \geq 3$ be an integer. Let $\Phi = (\mathcal{V}, \mathcal{C})$ be a k-CNF formula. We say that a variable $v \in \mathcal{V}$ is* high-degree *if the degree of v is at least $\Delta_r := \lceil 2^{rk} \rceil$.*

We refer to Section 4.2 for details on our procedure to determine the bad variables/clauses of the formula Φ. Roughly, bad variables consist of high-degree variables (as in Definition 4.1), plus those variables that appear in a clause with at least two other bad variables (recursively); bad clauses are those clauses that contain at least three bad variables. We use $\mathcal{V}_{\mathrm{bad}}(r)$ and $\mathcal{C}_{\mathrm{bad}}(r)$ to denote the sets of bad variables and clauses. We use $\mathcal{V}_{\mathrm{good}}(r) = \mathcal{V} \setminus \mathcal{V}_{\mathrm{bad}}(r)$ to denote the set of *good variables*, and $\mathcal{C}_{\mathrm{good}}(r) = \mathcal{C} \setminus \mathcal{C}_{\mathrm{bad}}(r)$ to denote the set of *good clauses*. The following proposition, proved in Section 4.2, summarises the main properties of the above sets.

Proposition 4.2. *Let $\Phi = (\mathcal{V}, \mathcal{C})$ be a k-CNF formula. For any $c \in \mathcal{C}_{\mathrm{good}}(r)$, we have $|\mathrm{var}(c) \cap \mathcal{V}_{\mathrm{bad}}(r)| \leq 2$, and for any $c \in \mathcal{C}_{\mathrm{bad}}(r)$, we have $|\mathrm{var}(c) \cap \mathcal{V}_{\mathrm{good}}(r)| = 0$. Moreover, every good variable has degree less than Δ_r. There is a procedure to determine $\mathcal{C}_{\mathrm{bad}}$ that runs in time $O(n + mk)$, where n is the number of variables of Φ and m is the number of clauses of Φ.*

It turns out that, w.h.p. over the choice of Φ, most clauses (and variables) in the random formula Φ are good, see Lemma 4.15 for a precise statement. At this stage, it would be natural to try to rework the Markov chain approach of [43]. To do this, we would split the set of good variables into *marked variables* and *control variables* in such a way that marked variables have marginals close to $1/2$. To ensure this bound on these marginal probabilities, it turns out that it is enough to find a marking such that each good clause has a high enough number of marked

variables. Then we run the Glauber dynamics on the set of marked variables. However, as we explain in Section 4.1.2, the state-of-the-art techniques used to analyse the mixing time of the single-site Glauber dynamics on bounded-degree formulae do not generalise to the random k-SAT setting; the main reason for this is that they fail to capture the effect that the high-degree variables have on the marginal probabilities of other variables. Therefore, we need to develop an alternative approach that is robust against the presence of high-degree variables. Our main contribution is an argument to apply the spectral independence framework [28, 29] to the random k-SAT model that leads to nearly linear sampling algorithms. To do this, it is important to introduce a third type of good variables, which we call the *auxiliary variables*. This motivates the following definition of marking.

Definition 4.3 (ρ-distributed, $(r, r_{\mathrm{m}}, r_{\mathrm{a}}, r_{\mathrm{c}})$-marking, r_0, r_1, δ)**.** *Let $r \in (0,1)$. Let $\Phi = (\mathcal{V}, \mathcal{C})$ be a k-CNF formula and let V be a subset of $\mathcal{V}_{\mathrm{good}}(r)$. We say that V is* ρ-distributed *if for each $c \in \mathcal{C}_{\mathrm{good}}(r)$ we have $|\mathrm{var}(c) \cap V| \geq \rho(k-3)$. An* $(r, r_{\mathrm{m}}, r_{\mathrm{a}}, r_{\mathrm{c}})$-marking *of Φ is a partition $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{a}}, \mathcal{V}_{\mathrm{c}})$ of the variables of Φ such that*

1. *the set of good variables $\mathcal{V}_{\mathrm{m}}$ is r_{m}-distributed;*
2. *the set of good variables $\mathcal{V}_{\mathrm{a}}$ is r_{a}-distributed.*
3. *$\mathcal{V}_{\mathrm{c}}$ contains all the bad variables and the set $\mathcal{V}_{\mathrm{c}} \setminus \mathcal{V}_{\mathrm{bad}}(r)$ is r_{c}-distributed;*

The variables in $\mathcal{V}_{\mathrm{m}}$ are called marked *variables, the variables in $\mathcal{V}_{\mathrm{a}}$ are called* auxiliary *variables, and the variables in $\mathcal{V}_{\mathrm{c}}$ are called* control *variables.*

In our sampling algorithm we work with $r = r_0 - \delta$ for $r_0 := 0.117841$ and $\delta := 0.00001$, and work with an $(r, r_0, r_0, 2r_0)$-marking. In our connectivity results (Theorems 1.10 and 1.12) we choose $r = r_1 - \delta$ for $r_1 := 0.227092$ and work with an $(r, r_1, 0, r_1)$-marking in order to achieve the larger density threshold.

In Section 4.3 we show that random k-CNF formulae have $(r_0 - \delta, r_0, r_0, 2r_0)$-markings when the density α is below the threshold $2^{(r_0-\delta)k}/k^3$, and that the marginals of good variables are close to $1/2$; this is where the value of r_0 becomes important in the argument. We also show that random k-CNF formulae have $(r_1 - \delta, r_1, 0, r_1)$-markings when the density α is below the threshold $2^{(r_1-\delta)k}/k^3$. We state this result for r_0 in Proposition 4.5 below; first we give some relevant definitions.

Definition 4.4 (Ω^*, μ_A, Ω, Φ^Λ, $\mathcal{C}^\Lambda$, $\mathcal{V}^\Lambda$, Ω^Λ)**.** *Let $\Phi = (\mathcal{V}, \mathcal{C})$ be a k-CNF formula. Let Ω^* be the set of all assignments $\mathcal{V} \to \{\mathsf{F}, \mathsf{T}\}$. Given any subset $A \subseteq \Omega^*$, let μ_A be the uniform distribution on A. Let Ω be the set of satisfying assignments of Φ. For any partial assignment Λ we denote by Φ^Λ the formula obtained by simplifying Φ under Λ, i.e., removing the clauses which are already satisfied by Λ, and removing false literals from the remaining clauses. We denote by $\mathcal{C}^\Lambda$ and $\mathcal{V}^\Lambda$ the sets of clauses and variables of Φ^Λ. Moreover, we denote by Ω^Λ the set of satisfying assignments of Φ^Λ.*

Recall that we say that an event $\mathcal{E}$ regarding the choice of the random formula Φ holds *with high probability* (abbreviated w.h.p.) if $\Pr(\mathcal{E}) = 1 - o(1)$ as $n \to \infty$, see Section 1.1.3 for the definition of random formula used in this probability distribution.

Proposition 4.5. *There is an integer k_0 such that for any $k \geq k_0$ and any density α with $\alpha \leq 2^{(r_0-\delta)k}/k^3$ the following holds w.h.p. over the choice of the random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. There exists an $(r_0 - \delta, r_0, r_0, 2r_0)$-marking $(\mathcal{V}_\mathrm{m}, \mathcal{V}_\mathrm{a}, \mathcal{V}_\mathrm{c})$ of Φ. Moreover, for any such marking, for any $v \in \mathcal{V}_{\mathrm{good}}(r_0 - \delta)$, any $V \subseteq \mathcal{V}_\mathrm{m} \cup \mathcal{V}_\mathrm{a}$ with $v \notin V$, and any $\Lambda \colon V \to \{\mathsf{F}, \mathsf{T}\}$, we have*

$$\max\left\{\Pr_{\mu_{\Omega\Lambda}}(v \mapsto \mathsf{F}), \Pr_{\mu_{\Omega\Lambda}}(v \mapsto \mathsf{T})\right\} \leq \frac{1}{2}\exp\left(\frac{1}{k2^{r_0 k}}\right).$$

Proof. This follows directly by combining Lemmas 4.21 and 4.23, which are stated and proved in Section 4.3. □

We note that the density threshold of Theorem 1.8 is $2^{0.039k}$, which is significantly smaller than the threshold $2^{(r_0-\delta)k}/k^3$ in Proposition 4.5. The bottleneck for the threshold Theorem 1.8 comes from our mixing time results, see Section 4.1.2.

The bound given in Proposition 4.5 on the marginal probabilities of the marked and auxiliary variables is exploited several times in this work, and we will explain some of these applications in this proof outline. We remark that the bound on the marginals of good variables holds for *any* pinning of *any* subset of marked and auxiliary variables, which will be relevant in the spectral independence argument.

Definition 4.6 ($\mu|_V$)**.** *Let $\mathcal{V}$ be a finite set and let $\Omega \subseteq \{\mathsf{F}, \mathsf{T}\}^{\mathcal{V}}$. Let μ be a distribution over Ω. For a set $V \subseteq \mathcal{V}$, we denote by $\mu|_V$ the marginal distribution of μ on V.*

Proposition 4.5 implies that the distribution $\mu_\Omega|_{\mathcal{V}_\mathrm{m} \cup \mathcal{V}_\mathrm{a}}$ is very close to the uniform distribution over all assignments $\mathcal{V}_\mathrm{m} \cup \mathcal{V}_\mathrm{a} \to \{\mathsf{F}, \mathsf{T}\}$. This concept is formalised in the following definition.

Definition 4.7 (ε-uniform)**.** *Let V be a set of variables and μ be a probability distribution over the assignments $V \to \{\mathsf{F}, \mathsf{T}\}$. Let $\Lambda \colon S \to \{\mathsf{F}, \mathsf{T}\}$ be an assignment of some subset of variables $S \subseteq V$. We denote by $\Pr_\mu(\Lambda)$ the probability under μ of the event that the variables in S are assigned values according to Λ, and by $\Pr_\mu(\cdot|\Lambda)$ the corresponding conditional distribution of μ.*

For $\varepsilon \in (0, 1)$, we say that the distribution μ is ε-uniform *if for any variable $v \in V$ and any partial assignment $\Lambda \colon V \setminus \{v\} \to \{\mathsf{F}, \mathsf{T}\}$, we have*

$$\max\{\Pr_\mu(v \mapsto \mathsf{F} | \Lambda), \Pr_\mu(v \mapsto \mathsf{T} | \Lambda)\} \leq \frac{1}{2}e^{\varepsilon}.$$

From Proposition 4.5, it follows that the distribution $\mu_\Omega|_{\mathcal{V}_\mathrm{m}}$ is ε-uniform for $\varepsilon = (2^{-r_0 k}/k)$, so for any $\Lambda \colon \mathcal{V}_\mathrm{m} \to \{\mathsf{F}, \mathsf{T}\}$, the probability that the assignment of the marked variables is Λ is at least $(1 - e^{\varepsilon}/2)^{|\mathcal{V}_\mathrm{m}|}$. The ε-uniform property also (trivially) guarantees that the space of assignments $\Lambda \colon \mathcal{V}_\mathrm{m} \to \{\mathsf{F}, \mathsf{T}\}$ with $\Pr_{\mu_\Omega}(\Lambda) > 0$ is connected via single-variable updates, so we can indeed consider the Glauber dynamics over $\mathcal{V}_\mathrm{m}$. This leads to the main challenge of this work: does this chain mix rapidly?

4.1.2 Mixing time of the Glauber dynamics on the marked variables

Recently, there has been significant progress in showing that the single-variable Glauber dynamics on appropriately chosen subsets of variables mixes quickly for k-CNF formulae with bounded degree [43, 74]. These approaches carefully execute a union bound over paths of clauses connecting marked variables in order to bound the coupling time between two copies of the chain. However, these union bound arguments break under the presence of high-degree variables that are present in random k-SAT; this is because the number of paths connecting marked variables is very sensitive to the max degree of the formula and in particular grows too fast in our setting. We give a more detailed discussion in Section 4.6.1.

Instead, we apply the spectral independence framework to show rapid mixing of a uniform-block Glauber dynamics, which we review briefly below. Applications of spectral independence usually exploit decay of correlations to show that the spectral independence condition holds, see [7, 28, 17] for examples. As we have mentioned in the introduction, correlation decay fails to hold for densities exponential in k in the random k-SAT model [95] and therefore, we have to develop a different approach to conclude that the spectral-independence condition holds in our setting. This is our main contribution in this work; we show that the marginal distribution on the marked variables, i.e., $\mu_\Omega|_{\mathcal{V}_\mathrm{m}}$, is $(\varepsilon \log n)$-spectrally independent for some $\varepsilon > 0$ that can be made arbitrarily small for sufficiently large k. Our argument builds on the coupling idea of Moitra [93] (as refined in [49] for random k-SAT) and relates the spectral independence condition to the expected number of failed clauses in this coupling process. This allows us to exploit the local sparsity properties of the random k-SAT model to analyse the mixing time of the Glauber dynamics.

A caveat here is that the spectral independence of $\mu_\Omega|_{\mathcal{V}_\mathrm{m}}$ is not enough on its own to conclude fast mixing of the single-site Glauber dynamics. The most direct way to work around this is to analyse instead the so-called ρ-uniform-block Glauber dynamics that updates ρ vertices at a time for some ρ that scales linearly in n; the main missing ingredient there is to show that the modified chain can be implemented efficiently which we discuss in Section 4.1.3. We next give a quick overview of the relevant ingredients of the spectral-independence literature that we will need.

4.1.2.1 The ρ-uniform-block Glauber dynamics, spectral independence, and the mixing time

Let V be a finite set of size M and μ be a distribution over the assignments $V \to \{\mathsf{F}, \mathsf{T}\}$. Let Ω be the set of assignments $V \to \{\mathsf{F}, \mathsf{T}\}$ with positive probability under μ. For an integer $\rho \in \{1, 2, \ldots, |V|\}$, the ρ-uniform-block Glauber dynamics for μ is a Markov chain X_t where $X_0 \in \Omega$ is an arbitrary configuration and, for $t \geq 1$, X_t is obtained from X_{t-1} by first picking a subset $S \subseteq V$ of size ρ uniformly at random, letting Λ_t be the restriction of X_t to $V \setminus S$, and updating the configuration on S according to the probability distribution $\mu(\cdot|\Lambda_t)$. This chain satisfies the detailed balance equation for μ. Hence, when the chain is irreducible, for $\varepsilon > 0$, we

can consider its mixing time $T_{\mathrm{mix}}(\rho,\varepsilon) = \max_{\sigma\in\Omega} \min\{t : d_{\mathrm{TV}}(X_t,\mu) \le \varepsilon \mid X_0 = \sigma\}$. We say that μ is *b-marginally bounded* if for all $v \in V$, $S \subseteq V \setminus \{v\}$, $\Lambda\colon S \to \{\mathsf{F},\mathsf{T}\}$ with $\Pr_\mu(\Lambda) > 0$, and $\omega \in \{\mathsf{F},\mathsf{T}\}$, it either holds that $\Pr_\mu(v \mapsto \omega|\Lambda) = 0$ or $\Pr_\mu(v \mapsto \omega|\Lambda) \ge b$. Spectral independence results have recently been used in the b-marginally bounded setting to obtain fast mixing time of the uniform-block Glauber dynamics [19, 29]. For $S \subset V$, $\Lambda\colon S \to \{\mathsf{F},\mathsf{T}\}$ with $\Pr_\mu(\Lambda) > 0$, and $u, v \in V \setminus S$ and $0 < \Pr_\mu(u \mapsto \mathsf{T}|\Lambda) < 1$, the *influence* of u on v (under μ and Λ) is defined as

$$\mathcal{I}^\Lambda(u \to v) = \Pr_\mu\left(v \mapsto \mathsf{T} | u \mapsto \mathsf{T}, \Lambda\right) - \Pr_\mu\left(v \mapsto \mathsf{T} | u \mapsto \mathsf{F}, \Lambda\right). \tag{4.1}$$

The *influence matrix conditioned on* Λ is the (two-dimensional) matrix whose entries consist of $\mathcal{I}^\Lambda(u \to v)$ over all relevant u and v. We denote by $\mathcal{I}^\Lambda$ the matrix and by $\lambda_1(\mathcal{I}^\Lambda)$ its largest absolute value of its eigenvalues. For a real $\eta > 0$, we say that μ is *η-spectrally independent* if for all $S \subset V$ and $\Lambda\colon S \to \{\mathsf{F},\mathsf{T}\}$ with $\Pr_\mu(\Lambda) > 0$ we have $\lambda_1(\mathcal{I}^\Lambda) \le \eta$. From the results of [29], one can conclude the following bound for the mixing time of the uniform-block Glauber dynamics, see Section 4.10 for details.

Lemma 4.8. *The following holds for any reals $b, \eta > 0$, any $\kappa \in (0,1)$ and any integer M with $M \ge \frac{2}{\kappa}(4\eta/b^2 + 1)$. Let V be a set of size M, let μ be a distribution over the assignments $V \to \{\mathsf{F},\mathsf{T}\}$, let $\Omega = \{\Lambda\colon V \to \{\mathsf{F},\mathsf{T}\} : \mu(\Lambda) > 0\}$ and let $\mu_{\min} = \min_{\Lambda\in\Omega} \mu(\Lambda)$. If μ is b-marginally bounded and η-spectrally independent, then, for $\rho = \lceil \kappa M \rceil$ and $C_\rho = (2/\kappa)^{4\eta/b^2+1}$, we have*

$$T_{\mathrm{mix}}(\rho,\varepsilon) \le \left\lceil C_\rho \frac{M}{\rho}\left(\log\log\frac{1}{\mu_{\min}} + \log\frac{1}{2\varepsilon^2}\right)\right\rceil.$$

We are going to consider the uniform-block Glauber dynamics on the marked variables of Φ, so $V = \mathcal{V}_{\mathrm{m}}$, and the set of states coincides with the set of assignments $\mathcal{V}_{\mathrm{m}} \to \{\mathsf{F},\mathsf{T}\}$ as all of them have positive probability. In this setting, the target distribution is $\mu_\Omega|_{\mathcal{V}_{\mathrm{m}}}$. The distribution $\mu_\Omega|_{\mathcal{V}_{\mathrm{m}}}$ is $(1/e)$-marginally-bounded as a straightforward consequence of the fact that it is $(1/k)$-uniform, see Remark 4.49 for details. Hence, in order to conclude rapid mixing it remains to establish spectral independence. For this, we are going to use the well-known fact (see for instance [28]) that, for $S \subset V$ and $\Lambda\colon S \to \{\mathsf{F},\mathsf{T}\}$, we have

$$\lambda_1(\mathcal{I}^\Lambda) \le \max_{u\in V\setminus S} \sum_{v\in V\setminus S} |\mathcal{I}^\Lambda(u \to v)|. \tag{4.2}$$

4.1.2.2 Spectral independence in the random k-SAT model

In this section we state our spectral independence results in the random k-SAT model. The results stated in this section are proved in Section 4.6. Our main technical result is the following.

Lemma 4.9. *There is an integer $k_0 \ge 3$ such that for any integer $k \ge k_0$ and any density α with $\alpha \le 2^{r_0 k/3}/k^3$ the following holds. W.h.p. over the choice of the random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$, for any $(r_0 - \delta, r_0, r_0, 2r_0)$-marking $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{a}}, \mathcal{V}_{\mathrm{c}})$ of Φ, the distribution $\mu_\Omega|_{\mathcal{V}_{\mathrm{m}}}$ is $(2^{-(r_0-\delta)k}\log n)$-spectrally independent.*

We are going to describe some of the ideas behind the proof of Lemma 4.9. First, we highlight the fact that, due to the presence of high-degree variables (which form logarithmically-sized connected components), current techniques seem unable to conclude η-spectral independence with $\eta = O(1)$. This has also been the case in recent work on 2-spin systems on random graphs [17], where instead correlation decay is exploited to prove η-spectral independence for some $\eta = o(\log n)$. Here, our η-spectral independence bound for $\eta = o_k(\log n)$ will be based on an appropriate coupling. Note, in light of Lemma 4.8, $\eta = O(\log n)$ is good enough for proving polynomial mixing time of the uniform-block Glauber dynamics, but we need the improved bound of Lemma 4.9 in order to conclude the following fast mixing-time result from Lemma 4.8 (as illustrated Section 4.6).

Lemma 4.10. *There is a function $k_0(\theta) = \Theta(\log(1/\theta))$ such that, for any $\theta \in (0,1)$, for any integer $k \geq k_0(\theta)$ and any density α with $\alpha \leq 2^{0.039k}$ the following holds. W.h.p. over the choice of the random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$, for any $(r_0 - \delta, r_0, r_0, 2r_0)$-marking $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{a}}, \mathcal{V}_{\mathrm{c}})$ of Φ and for $\rho = \lceil 2^{-k-1}|\mathcal{V}_{\mathrm{m}}| \rceil$, the ρ-uniform-block Glauber dynamics for updating the marked variables has mixing time $T_{\mathrm{mix}}(\rho, \varepsilon/2) \leq T := \lceil 2^{2k+3} n^{\theta} \log \frac{2n}{\varepsilon^2} \rceil$.*

Lemma 4.10 is stated for the block size $\rho = \lceil 2^{-k-1}|\mathcal{V}_{\mathrm{m}}| \rceil$, but it could be proved more generally when $\rho = c|\mathcal{V}_{\mathrm{m}}|$ and $c \in (0,1)$. The fact that $\rho \leq |\mathcal{V}_{\mathrm{m}}|/2^k$ in the statement will be relevant in implementing efficiently the dynamics, discussed in Section 4.1.3.

We remark that the more restrictive density threshold $\alpha \leq 2^{r_0 k/3}/k^3$ in the statement of Lemma 4.9 arises in the union bound given in the proof of this lemma, and that for large enough k we have $2^{0.039k} \leq 2^{r_0 k/3}/k^3$, the former being the density threshold given in Lemma 4.10 and Theorem 1.8.

Our approach to prove η-spectral independence significantly differs from those that in two-spin systems, where it is enough to study sum of influences over trees (thanks to the tree of self-avoiding walks) and exploit decay of correlations in this setting (very roughly, the further away two vertices are in the tree, the smaller the influence that one vertex has in the other). Here we relate influences to the structure of the dependency graph G_Φ by running a coupling process on the auxiliary variables, and we state this connection in the upcoming Lemma 4.40. First we define more formally the dependency graph G_Φ.

Definition 4.11 (G_Φ)**.** *Let $\Phi = (\mathcal{V}, \mathcal{C})$ be a k-CNF formula. We define the graph G_Φ as follows. The vertex set of G_Φ is $\mathcal{C}$ and two clauses c_1 and c_2 are adjacent if and only if $\mathrm{var}(c_1) \cap \mathrm{var}(c_2) \neq \emptyset$. A set $C \subseteq \mathcal{C}$ is connected if C is connected in the graph G_Φ. We say that two variables u and v are connected in Φ if there is a path $c_1, c_2, \ldots, c_\ell$ in G_Φ with $u \in \mathrm{var}(c_1)$ and $v \in \mathrm{var}(c_\ell)$.*

Let $u \in \mathcal{V}_{\mathrm{m}}$, $S \subset \mathcal{V}_{\mathrm{m}}$ and $\Lambda \colon S \to \{\mathsf{F}, \mathsf{T}\}$. The aim of the coupling process is bounding the sum $\sum_{v \in \mathcal{V}_{\mathrm{m}} \setminus (S \cup \{u\})} |\mathcal{I}^{\Lambda}(u \to v)|$ in terms of the expected size of a connected set of *failed* clauses, where the expectation is over the choices made in the coupling process. We refer to Section 4.6 for a definition of failed clauses, as it is not relevant in this discussion. Here we give a brief overview of how the coupling process on the auxiliary variables works. First, we start

with two assignments $X = \Lambda \cup (u \mapsto \mathsf{T})$ and $Y = \Lambda \cup (u \mapsto \mathsf{F})$, where $\Lambda \cup (u \mapsto \omega)$ denotes the assignment defined on $S \cup \{u\}$ that agrees with Λ on S and sends u to ω. The process progressively extends X and Y on some auxiliary variables $v_1, v_2, \ldots$ following the optimal coupling between the marginals $\Pr_{\mu_\Omega}(v \mapsto \cdot|X)$ and $\Pr_{\mu_\Omega}(v \mapsto \cdot|Y)$, see Section 4.6 for the definition of optimal coupling. The main property of this process is that with high probability over the choices made, at some point the graphs G_{Φ^X} and G_{Φ^Y} factorise in small connected components in spite of the presence of bad variables and, on top of that, Φ^X and Φ^Y share most of these connected components. Then we can bound influences between marked variables by analysing the connected components where Φ^X and Φ^Y differ, which turn out to be $\mathrm{poly}(k) \log n$ in size after enough steps of the process.

One of the key ideas behind our analysis is exploiting the fact that, in the random k-SAT model, w.h.p. over the choice of the random formula Φ, any logarithmic-sized set of clauses Z that is connected in G_Φ has constant tree-excess, that is, the number of edges connecting a pair of clauses in Z is $|Z| + O(1)$. This saves a factor of $\Delta_{r_0-\delta}$ in the spectral independence bound by ensuring that there is a large independent set of clauses in the set of failed clauses. We also obtain improved analysis by restricting the coupling process to auxiliary variables. This enables us to get exponentially small bounds (in k) on the influences between marked variables, which leads to our $(2^{-(r_0-\delta)k} \log n)$-spectral independence result.

4.1.3 Analysis of the connected components of Φ^Λ

In this section we deal with the third challenge mentioned at the beginning of Section 4.1: can we determine the transition probabilities of the Glauber dynamics so that we can actually simulate this Markov chain? In fact, simulating the single-site Glauber dynamics on the marked variables was one of the main challenges even in the bounded-degree case. In that case this was resolved using a method that is restricted to the bounded-degree setting (and whose bottleneck is the analysis of a rejection sampling procedure). A different procedure is required for the random k-SAT setting.

One of the key ideas to simulate this chain is starting the chain on an assignment $X_0 \colon \mathcal{V}_\mathrm{m} \to \{\mathsf{F}, \mathsf{T}\}$ drawn from the uniform distribution over all assignments of $\mathcal{V}_\mathrm{m}$. Since the distribution $\mu_\Omega|_{\mathcal{V}_\mathrm{m}}$ is $(1/k)$-uniform (Proposition 4.5), the transition probabilities of the Glauber dynamics are close to uniform. This allows us to show that the probability distribution of the assignment X_t that is output by the uniform-block Glauber dynamics after t steps is also $(1/k)$-uniform (Corollary 4.24), which will be important in what follows.

In order to run the ρ-uniform-block Glauber dynamics we need to be able to sample from the distribution μ_{Ω^Λ} for any set $S \subseteq \mathcal{V}_\mathrm{m}$ with $|S| = \rho$ and any assignment $\Lambda \colon \mathcal{V}_\mathrm{m} \setminus S \to \{\mathsf{F}, \mathsf{T}\}$ that arises. Unless we can restrict Λ, sampling from μ_{Ω^Λ} could potentially be as hard as sampling from μ_Ω. Fortunately for us, the assignment Λ is not completely arbitrary; Λ is determined by the random choice of S and the current state of the Glauber dynamics (which follows a $(1/k)$-uniform distribution as discussed above). We show that we can efficiently sample from μ_{Ω^Λ}

w.h.p. over the choice of Λ. An important observation is that we can efficiently sample from μ_{Ω^Λ} when the connected components of G_{Φ^Λ} are logarithmic in size, for example, by applying brute force. This raises the following question: does G_{Φ^Λ} break into small connected components w.h.p. over the choice of Λ? Lemma 4.12 gives a positive answer when $0 \le \rho \le |V|/2^k$. Here the reader can see V as the set of marked variables. The proof of Lemma 4.12 exploits sparsity properties of logarithmic-sized connected sets of clauses in random formulae in conjunction with the fact that μ is $(1/k)$-uniform. Lemma 4.12 is stated with an added layer of generality, as we will also apply it to analyse the geometry of the space of satisfying assignments of Φ with $r = r_1 - \delta$. In our sampling algorithm setting we consider $r = r_0 - \delta$. Recall that $r_0 = 0.117841$, $r_1 = 0.227092$ and $\delta = 0.00001$. The restriction $r \in (2\delta, 1/(2\log 2)]$ in the statement of Lemma 4.12 is not optimal, but it is enough for our purposes.

Lemma 4.12. *Let $r \in (2\delta, 1/(2\log 2)]$. There is an integer $k_0 \ge 3$ such that, for any integer $k \ge k_0$, any density $\alpha \le 2^{(r-2\delta)k}$, and any real number b with $a := 2k^4 < b$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$.*

Let L be an integer satisfying $a \log n \le L \le b \log n$. Let V be a set of good variables of Φ that is $(r+\delta)$-distributed (Definition 4.3), let μ be a $(1/k)$-uniform distribution over the assignments $V \to \{\mathsf{F}, \mathsf{T}\}$, and let ρ be an integer with $0 \le \rho \le |V|/2^k$. Consider the following experiment. First, draw $S \subseteq V$ from the uniform distribution τ over subsets of V with size ρ. Then, sample an assignment Λ from $\mu|_{V \setminus S}$. Denote by $\mathcal{F}$ the event that that there is a connected set of clauses Y of Φ with $|Y| \ge L$ such that all clauses in Y are unsatisfied by Λ. Then $\Pr_{S \sim \tau}\left(\Pr_{\Lambda \sim \mu|_{V\setminus S}}(\mathcal{F}) \le 2^{-\delta k L}\right) \ge 1 - 2^{-\delta k L}$.

Proof sketch. The proof is in Section 4.4. For the sake of exposition, we first sketch the proof in the case $\rho = 0$, where the conclusion in the statement reads $\Pr_{\Lambda \sim \mu|_V}(\mathcal{F}) \le 2^{-\delta k L}$. At the end of this proof sketch we explain how we extend the proof to any ρ with $0 \le \rho \le |V|/2^k$.

The first step is exploiting local sparsity properties of random k-CNF formulae to find many variables from V in any sufficiently large connected set of clauses. Our sparsity results hold for connected sets of clauses with size at least $2k^4 \log n$, and let us conclude the following result (stated as Lemma 4.28 in Section 4.4): w.h.p. over the choice of Φ, for every connected set of clauses $Z \subseteq \mathcal{C}$ we have

$$\text{if } \ 2k^4 \log(n) \le |Z| \le b \log(n), \quad \text{then } \ |\mathrm{var}(Z) \cap V| \ge rk|Z|. \tag{4.3}$$

The proof of Lemma 4.28 counts the variables from V in Z by using the fact that Z does not contain many bad clauses (Lemma 4.15, which gives the restriction on r) and the fact that there are not many edges joining clauses in Z. In fact, for such a set Z, we show that the number of edges is of order $|Z| + O(1)$, that is, Z has constant tree-excess (Lemma 4.26). We also need the following result on random k-CNF formulae. For each clause $c \in \mathcal{C}$, let $\mathcal{Z}(c, L) = \{Z \subseteq \mathcal{C} : c \in Z, Z \text{ is connected in } G_\Phi, |Z| = L\}$. Then, w.h.p. over the choice of Φ, [49, Lemma 40] shows that, as long as $L \ge \log n$,

$$\text{for any clause } c \in \mathcal{C} \text{ we have } \ |\mathcal{Z}(c, L)| \le (9k^2\alpha)^L. \tag{4.4}$$

Once we have established (4.3) and (4.4), the proof exploits the fact that μ is close to the uniform distribution. First, we introduce some notation. Let L be an integer with $a \log n \leq L \leq b \log n$. Let $S = \emptyset$ as we are dealing with the case $\rho = 0$. For $c \in \mathcal{C}$ and $Z \in \mathcal{Z}(c, L)$, we denote by $\mathcal{E}_1(Z, S)$ the event that none of the clauses of Z are satisfied by assignment Λ (Definition 4.4), where Λ is drawn from $\mu|_{V \setminus S}$, see Definition 4.6. We keep track of S in the notation here as this is relevant in the general case. The first observation is that the event $\mathcal{F}$ from the statement satisfies $\mathcal{F} = \bigcup_{c \in \mathcal{C}, Z \in \mathcal{Z}(c,L)} \mathcal{E}_1(Z, S)$. We then claim that for any $c \in \mathcal{C}$ and $Z \in \mathcal{Z}(c, L)$ we have

$$\Pr_{\Lambda \sim \mu|_{V \setminus S}} (\mathcal{E}_1(Z, S)) \leq \frac{2^{-\delta k L}}{|\mathcal{C}| \cdot |\mathcal{Z}(c, L)|}, \tag{4.5}$$

so the result would follow from a union bound over c and Z. Let us give some insight on how we prove (4.5). Let $c \in \mathcal{C}$ and $Z \in \mathcal{Z}(c, L)$. The main idea is that, if all clauses in Z are unsatisfied by Λ then, when we sampled $\Lambda \sim \mu|_{V \setminus S}$, for each variable v in $\mathrm{var}(Z) \cap (V \setminus S)$ we picked the value that does not satisfy the clauses of Z containing v. Thus, we can bound the probability that all clauses in Z are unsatisfied as a product, over the variables in $\mathrm{var}(Z) \cap (V \setminus S)$, of probabilities, each factor corresponding to the probability that a variable is assigned a certain value (under some careful conditioning, see the proof in Section 4.4 for details). Since the distribution μ is $(1/k)$-uniform, each one of these factors can be bounded by $\exp(1/k)/2$, obtaining

$$\Pr_{\Lambda \sim \mu|_{V \setminus S}} (\mathcal{E}_1(Z, S)) \leq \left(\frac{1}{2} \exp\left(\frac{1}{k} \right) \right)^{|\mathrm{var}(Z) \cap (V \setminus S)|}. \tag{4.6}$$

In (4.3) we gave a lower bound on $|\mathrm{var}(Z) \cap V|$, which can be applied in conjunction with (4.4) to conclude, after some calculations, that the bound given in (4.5) holds.

The case $\rho > 0$ is more technical and one has to be more careful in these calculations. We show that (4.5) holds when S does not contain many variables in $\mathrm{var}(Z) \cap V$. A slightly different argument is needed when going from (4.6) to (4.5); here we have to bound $|\mathrm{var}(Z) \cap (V \setminus S)|$ instead of $|\mathrm{var}(Z) \cap V|$. It turns out that, as long as the bound $|\mathrm{var}(Z) \cap V \cap S| \leq |\mathrm{var}(Z) \cap V|/k$ holds, the calculations to go from (4.6) to (4.5) also hold in this setting. Finally, we show that the probability that $|\mathrm{var}(Z) \cap V \cap S| \leq |\mathrm{var}(Z) \cap V|/k$ occurs when picking S is at least $1 - 2^{\delta k L}$. The proof of this fact is purely combinatorial, and requires the hypothesis $\rho \leq |V|/2^k$, see Section 4.4 for details. □

Once we have established Lemma 4.12, we can use it to implement the ρ-uniform-block Glauber dynamics on the marked variables for $0 < \rho \leq |\mathcal{V}_{\mathrm{m}}|$ and complete our sampling algorithm, which we explicitly state in Section 4.1.4.

Before concluding this section, we mention how we apply Lemma 4.12 to analyse the geometry of the space of satisfying assignments of Φ in order to conclude the $O(\log n)$-connectivity and $O(\log n)$-looseness results given in Theorems 1.10 and 1.12. First, we need the following definition.

Definition 4.13 (H_Φ)**.** *Let $\Phi = (\mathcal{V}, \mathcal{C})$ be a k-CNF formula. We define the graph H_Φ as follows. The vertex set of H_Φ is $\mathcal{V}$ and two variables v_1 and v_2 are adjacent in H_Φ if there is a clause $c \in \mathcal{C}$ with $v_1, v_2 \in \mathrm{var}(c)$.*

We apply Lemma 4.12 with $r = r_1 - \delta$ and a density $\alpha \leq 2^{(r_1-3\delta)k}/k^3$. For an $(r, r_1, 0, r_1)$-marking $(\mathcal{V}_\mathrm{m}, \emptyset, \mathcal{V}_\mathrm{c})$ of Φ, we let $V = \mathcal{V}_\mathrm{m}$ and $\mu = \mu_\Omega|_{\mathcal{V}_\mathrm{m}}$. In this setting, for $\rho = 0$, Lemma 4.12 allows us to conclude that, w.h.p. over the choice of $\Lambda \sim \mu_\Omega|_{\mathcal{V}_\mathrm{m}}$, the graph G_{Φ^Λ} consists of connected components with size at most $O(\log n)$. Thus, the connected components of H_{Φ^Λ} have size at most $O(\log n)$ as each clause contains at most k variables. This leads to the main idea behind the proof of Theorem 1.10: we can construct $O(\log n)$-paths between satisfying assignments by progressively updating the variables in each one of the connected components of H_{Φ^Λ}. As an example, let $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_t$ be these connected components and let σ_1 and σ_2 be two satisfying assignments that agree with Λ on $\mathcal{V}_\mathrm{m}$. Then we can find an $O(\log n)$-path $\sigma_1 = \zeta_0 \leftrightarrow \zeta_1 \leftrightarrow \cdots \leftrightarrow \zeta_t = \sigma_2$ as follows: the assignment ζ_j is the satisfying assignment that agrees with Λ, agrees with σ_1 on the variables in $\mathcal{V} \setminus \left(\bigcup_{i=1}^{j} \mathcal{E}_j\right)$ and agrees with σ_2 on the variables in $\bigcup_{i=1}^{j} \mathcal{E}_j$. The case when σ_1 and σ_2 differ on some marked variables builds on the same idea though it is more technical and requires applying Lemma 4.12 with $\rho = 1$. We refer to Section 4.8.1 for this argument and the proof of Theorem 1.10.

The fact that the connected components of H_{Φ^Λ} are $O(\log n)$ in size with high probability over $\Lambda \sim \mu_\Omega|_{\mathcal{V}_\mathrm{m}}$ is also related to the looseness of the formula Φ. Let $v \in \mathcal{V} \setminus \mathcal{V}_\mathrm{m}$. For any satisfying assignment σ that agrees with Λ on the marked variables, we can construct a satisfying assignment τ with $\tau(v) \neq \sigma(v)$ and $\|\sigma - \tau\|_1 = O(\log n)$ by updating the variables in the connected component of v in H_{Φ^Λ}, provided that there is a way to satisfy this connected component when giving v the value $\tau(v)$. In Section 4.8.2 we formalise this idea and give all the details of this argument to prove Theorem 1.12.

4.1.4 The sampling algorithm

To complete this proof outline, we explicitly describe Algorithm 1, our algorithm for sampling satisfying assignments of k-CNF formulae. The algorithm uses a method Sample(Φ^Λ, S) to sample an assignment $\tau \colon S \to \{\mathsf{F}, \mathsf{T}\}$ from the distribution $\mu_{\Omega^\Lambda}|_S$. This method exploits the fact that logarithmic-sized connected set of clauses have constant tree-excess, which does not hold in the bounded-degree case. This tree-like property enables us to efficiently sample satisfying assignments on the connected components of Φ^Λ by a standard dynamic programming argument, see Section 4.5. Lemma 4.14 is our main result on Sample(Φ^Λ, S).

Lemma 4.14. *There is an integer $k_0 \geq 3$ such that, for any integers $k \geq k_0$, $b \geq 2k^4$ and any density $\alpha > 0$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. Let V be a subset of variables and let $\Lambda \colon V \to \{\mathsf{F}, \mathsf{T}\}$ be a partial assignment such that all the connected components in G_{Φ^Λ} have size at most $b \log(n)$. Then, there is an algorithm that, for any $S \subseteq \mathcal{V} \setminus V$, samples an assignment from $\mu_{\Omega^\Lambda}|_S$ in time $O(|S| \log n)$.*

The method Sample(Φ^Λ, S) is used in Algorithm 1 to implement each step of the ρ-uniform-block Glauber dynamics on the marked variables. It is also used to extend the assignment of marked variables computed by the Glauber dynamics to a satisfying assignment of Φ. As a

design choice, this method returns *error* when the connected components of G_{Φ^Λ} have size larger than $2k^4(1+\xi)\log(n)$. We remark that the probability that Sample(Φ^Λ, S) returns *error* is very small when running the Glauber dynamics thanks to Lemma 4.12. We can now introduce Algorithm 1, which has two parameters $\theta \in (0,1)$ and $\xi \geq 1$ as in Theorem 1.8.

Algorithm 1 The approximate sampling algorithm for satisfying assignments of random k-CNF formulae.

Input: A k-CNF formula $\Phi = (\mathcal{V}, \mathcal{C})$ with n variables

1: Compute the sets of bad/good variables and bad/good clauses for Φ as in Proposition 4.2.
2: Let $\varepsilon = n^{-\xi}$. Compute an $(r_0 - \delta, r_0, r_0, 2r_0)$-marking $(\mathcal{V}_\mathrm{m}, \mathcal{V}_\mathrm{a}, \mathcal{V}_\mathrm{c})$ for Φ as in Proposition 4.5 (see Lemma 4.21 for the algorithm, use $p = \varepsilon/4$). This succeeds with probability at least $1 - \varepsilon/4$. If this does not succeed, the algorithm returns *error*.
3: For each $v \in \mathcal{V}_\mathrm{m}$, sample $X_0(v) \in \{\mathsf{F}, \mathsf{T}\}$ uniformly at random.
4: **for** t from 1 to $T := \lceil 2^{2k+3} n^\theta \log \frac{2n}{\varepsilon^2} \rceil$ **do**
5: Choose uniformly at random a set of marked variables $S \subseteq \mathcal{V}_\mathrm{m}$ with size $\rho := \lceil 2^{-k-1} |\mathcal{V}_\mathrm{m}| \rceil$.
6: Let Λ_t be the assignment X_{t-1} restricted to $\mathcal{V}_\mathrm{m} \setminus S$.
7: $Y \leftarrow$ Sample(Φ^{Λ_t}, S).
8: $X_t \leftarrow \Lambda_t \cup Y$.
9: **end for**
10: $Y \leftarrow$ Sample$(\Phi^{X_T}, \mathcal{V}_\mathrm{a} \cup \mathcal{V}_\mathrm{c})$.
11: **return** $X_T \cup Y$.

We remark here that Algorithm 1 only works for large enough k, and this hypothesis will be used several times in our arguments. The quantity T defined in this algorithm corresponds to the mixing time of the ρ-uniform-block Glauber dynamics given in Lemma 4.10.

4.1.5 Organisation of the rest of this chapter

The rest of this chapter is organised as follows. In Section 4.2 we introduce a procedure for determining bad clauses of a k-CNF formula. In Section 4.3 we prove Proposition 4.5 on markings of random formulae. In Section 4.4 we prove our technical result on the connected components of Φ^Λ, Lemma 4.12. In Section 4.5 we give the method Sample and prove Lemma 4.14. In Section 4.6 we prove the results on spectral independence stated in Section 4.1.2 of this proof outline. In Section 4.7 we complete the proof of Theorem 1.8 by combining our mixing time results (Lemma 4.10), our algorithm to sample from small connected components (Lemma 4.14) and our result on the size of the connected components of Φ^Λ (Lemma 4.12). In Section 4.8 we prove Theorems 1.10 and 1.12 on the geometry of the space of satisfying assignments of Φ. Finally, in Sections 4.9 and 4.10 we prove three lemmas that are independent of the rest of this work. More precisely, in Section 4.10 we bound the number of bad clauses in a random k-CNF formula, both globally and for sufficiently large connected subsets of clauses. Finally, in Section 4.10 we prove Lemma 4.8 on the uniform-block Glauber dynamics, which follows from

combining some result of [29].

4.2 High-degree and bad variables in random CNF formulae

As we noted in Section 1.5 and in our proof outline, one of the keys to sampling satisfying assignments in the unbounded-degree setting is to "sacrifice" a few variables per clause (treating them separately in the sampling algorithm) and to (temporarily) remove a small linear number of clauses that contain these. The point of this is to ensure that the remaining ("good") clauses have mostly low-degree variables (at most two bad ones) and also that the rest of the clauses (the "bad" ones) form small connected components that interact with the good clauses in a manageable way.

Recall that, for $r \in (0,1)$, high-degree variables were introduced in Definition 4.1 as those variables with at least $\Delta_r := \lceil 2^{kr} \rceil$ occurrences in the formula. In this work we consider two possible values for r here, $r = r_0 - \delta$ and $r = r_1 - \delta$, where $r_0 = 0.117841$, $r_1 = 0.227092$ and $\delta = 0.00001$. The values r_0 and r_1 arise as solutions of an optimisation problem in Section 4.3 when we establish the markings that we use in our proofs. The marking used in our algorithmic results requires the more restrictive definition of high-degree variable with $r = r_0 - \delta$ than the marking used in our connectivity results with $r = r_1 - \delta$. Subtracting δ will make our calculations easier without affecting our results.

By standard arguments about random graphs, one can determine that, w.h.p. over the choice of Φ, the number of high-degree variables of Φ is bounded. We want to identify the clauses of Φ that have at most 2 high-degree variables, since clauses with a lot of high-degree variables will interfere with our sampling algorithms. This motivates the following construction. The *bad variables* and *bad clauses* of Φ are identified by running the process given in Algorithm 2. Here $\mathcal{V}_{\text{bad}}(r)$ denotes the set of bad variables and $\mathcal{C}_{\text{bad}}(r)$ denotes the set of bad clauses.

Algorithm 2 Computing bad variables and bad clauses for $r \in (0,1)$

Input: A k-CNF formula $\Phi = (\mathcal{V}, \mathcal{C})$

1: $\mathcal{V}_0(r) \leftarrow$ the set of high-degree variables, i.e., variables with at least $\Delta_r = \lceil 2^{rk} \rceil$ occurrences in Φ.
2: $\mathcal{C}_0(r) \leftarrow$ the set of clauses with at least 3 variables in $\mathcal{V}_0(r)$
3: $i \leftarrow 0$
4: **while** $i = 0$ or $\mathcal{V}_i(r) \neq \mathcal{V}_{i-1}(r)$ **do**
5: $\quad i \leftarrow i + 1$
6: $\quad \mathcal{V}_i(r) \leftarrow \mathcal{V}_{i-1}(r) \cup \text{var}(\mathcal{C}_{i-1}(r))$
7: $\quad \mathcal{C}_i(r) \leftarrow \{c \in \mathcal{C} : |\text{var}(c) \cap \mathcal{V}_i(r)| \geq 3\}$
8: **end while**
9: $\mathcal{C}_{\text{bad}}(r) \leftarrow \mathcal{C}_i(r)$ and $\mathcal{V}_{\text{bad}} \leftarrow \mathcal{V}_i(r)$
10: **return** $\mathcal{V}_{\text{bad}}(r), \mathcal{C}_{\text{bad}}(r)$

We define the *good clauses* of Φ as $\mathcal{C}_{\text{good}}(r) = \mathcal{C} \setminus \mathcal{C}_{\text{bad}}(r)$ and the *good variables* of Φ as $\mathcal{V}_{\text{good}}(r) = \mathcal{C} \setminus \mathcal{V}_{\text{bad}}(r)$. The sets $\mathcal{V}_{\text{good}}(r), \mathcal{V}_{\text{bad}}(r), \mathcal{C}_{\text{good}}(r), \mathcal{C}_{\text{bad}}(r)$ depend on the parameter $r \in (0,1)$. The value of r here will be $r_0 - \delta$ except in Section 4.8 where we prove our connectivity results for $r = r_1 - \delta$, and in some of the marking results in Section 4.3. We will use the observations given in Proposition 4.2 several times in this work.

Proposition 4.2. *Let $\Phi = (\mathcal{V}, \mathcal{C})$ be a k-CNF formula. For any $c \in \mathcal{C}_{\text{good}}(r)$, we have $|\text{var}(c) \cap \mathcal{V}_{\text{bad}}(r)| \leq 2$, and for any $c \in \mathcal{C}_{\text{bad}}(r)$, we have $|\text{var}(c) \cap \mathcal{V}_{\text{good}}(r)| = 0$. Moreover, every good variable has degree less than Δ_r. There is a procedure to determine $\mathcal{C}_{\text{bad}}$ that runs in time $O(n + mk)$, where n is the number of variables of Φ and m is the number of clauses of Φ.*

Proof. In this proof we briefly explain the implementation of Algorithm 2. First, for each clause c we keep track of the number of bad variables in $\text{var}(c)$, denoted $\text{bad}(c)$. We also have a stack of bad variables $S_{\mathcal{V}}$ that are yet to be processed by the algorithm. At the start of the algorithm, we set $S_{\mathcal{V}} \leftarrow \mathcal{V}_0$. While $S_{\mathcal{V}}$ is non-empty, we take the variable v on the top of the stack and increase $\text{bad}(c')$ by 1 for those clauses c' where v appears. If any of these updates gives $\text{bad}(c') \geq 3$, we add $\text{var}(c')$ to the stack $S_{\mathcal{V}}$, set the variables in $\text{var}(c')$ as bad and set the clause c' as bad. At the end of this process, $S_{\mathcal{V}}$ is empty and we have found all the bad variables and bad clauses of Φ. As every variable is added to the stack at most once and the list $\text{bad}(\cdot)$ is updated at most mk times (once per literal in Φ), the running time is $O(n + mk)$. □

In our work we need a variation of result of [49] that controls the number of bad clauses in connected subgraphs of G_Φ. We state this result in Lemma 4.15 and prove it in Section 4.9.

Lemma 4.15 (Modified version of [49, Lemma 8.16])**.** *Let $r \in (0, 1/(2\log 2)]$. There is a positive integer k_0 such that for any integer $k \geq k_0$, $\Delta_r = \lceil 2^{rk} \rceil$, and any density α with $\alpha \leq \Delta_r / k^3$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. For every connected set of clauses Y in G_Φ such that $|\text{var}(Y)| \geq 2k^4 \log n$, we have $|Y \cap \mathcal{C}_{\text{bad}}(r)| \leq |Y|/k$.*

We also need a bound on the number of bad clauses of Φ, which is also proved in Section 4.9.

Lemma 4.16 (Modified version of [49, Lemma 8.12])**.** *Let $r \in (0, 1/(2\log 2)]$. There is a positive integer k_0 such that for any integer $k \geq k_0$, $\Delta_r = \lceil 2^{rk} \rceil$, and any density α with $\alpha \leq \Delta_r / k^3$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. We have $|\mathcal{C}_{\text{bad}}(r)| \leq 2(\alpha/\Delta_r) n / 2^{k^{10}}$ and $|\mathcal{V}_{\text{bad}}(r)| \leq 2(k+1)(\alpha/\Delta_r) n / 2^{k^{10}}$.*

Lemmas 4.15 and 4.16 guarantee that, w.h.p. over the choice of Φ, bad clauses are a minority among all the clauses of Φ. This will be used to show that bad clauses do not affect significantly the behaviour of our sampling algorithm. We point out that the definitions of $\mathcal{V}_{\text{good}}(r), \mathcal{V}_{\text{bad}}(r), \mathcal{C}_{\text{good}}(r)$ and $\mathcal{C}_{\text{bad}}(r)$ given in [49] have $r = 1/300$ and, in Algorithm 2, use the condition $|\text{var}(c) \cap \mathcal{V}_i(r)| \geq k/10$ instead of $|\text{var}(c) \cap \mathcal{V}_i(r)| \geq 3$

Hence, our definitions of good clauses and good variables are more restrictive. However, it turns out that, with minor changes, the proof of Lemma 4.15 given in [49] can be extended to our setting. These changes are explained in Section 4.9.

4.3 Identifying a set of "marked" variables with good marginals

A property that is useful for sampling satisfying assignments is having a high proportion of variables in each good clause such that the marginals of these variables are fairly close to 1/2. That is, having variables which are roughly equally likely to be true or false in a random satisfying assignment. The marginals of high-degree variables do vary. However, even in the random k-SAT model it turns out that there are enough variables with marginals near 1/2. Following the basic approach of Moitra [93], we partition the good variables of a random k-CNF formula into types. Here we have three types of variables (instead of two): marked, auxiliary and control variables. The high-level goal is to do this in such a way that each clause has a good proportion of each one of these types of variables. We call this construction a marking, see Definition 4.3 of the proof outline for the precise definition. For such a marking, we will show that as long as the control variables are left unassigned/unpinned, the marginals of the marked and auxiliary variables are all near 1/2 as a consequence the Lovász local lemma [42]. We first set up the notation and results that we need.

It is not difficult to show that in the random k-SAT model, w.h.p. over the choice of the formula Φ, two distinct clauses share at most 2 variables (see Lemma 4.17). Previous work on counting/sampling satisfying assignments of bounded degree formulae had to analyse subsets of disjoint clauses in order to deal with the fact that small sets of clauses might share most of their variables. The restriction to disjoint subsets imposes further restrictions on the maximum degree of the formula and on the density of the formula in the random k-SAT model setting. Here we manage to exploit Lemma 4.17 to avoid these restrictions.

Lemma 4.17. *For any $k \geq 3$ and any density $\alpha > 0$ (possibly depending on k), the following holds w.h.p. over the choice of the random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. We have $|\mathrm{var}(c)| \geq k - 1$ and $|\mathrm{var}(c) \cap \mathrm{var}(c')| \leq 2$ for all $c, c' \in \mathcal{C}$ with $c \neq c'$.*

Proof. First, let us prove that, for $k \geq 3$, w.h.p. over the choice of Φ, $|\mathrm{var}(c)| \geq k-1$ for all $c \in \mathcal{C}$. Let us denote by $\mathcal{R}_c$ the event that a clause c has at least two repetitions among its variables, that is, $|\mathrm{var}(c)| \leq k - 2$. We claim that $\Pr(\mathcal{R}_c) \leq q(k)/n^2$, where $q = \binom{k}{3} + k(k-1)(k-2)(k-3)/4$. To prove this statement we note that the probability that a variable appears at least 3 times in c is at most $\binom{k}{3} n^{k-2}/n^k$, and the probability that two distinct variables are repeated in c is at most $p(k)n(n-1)n^{k-4}/n^k$ for $p(k) = k(k-1)(k-2)(k-3)/4$. Hence, by adding up both cases, we find that $\Pr(\mathcal{R}_c) \leq q(k)/n^2$, and $\Pr(\bigcup_{c \in \mathcal{C}} \mathcal{R}_c) \leq q(k)m/n^2 \leq q(k)\alpha/n = O(1/n)$, so the result follows.

Let $c, c' \in \mathcal{C}$ with $c \neq c'$. We study $|\mathrm{var}(c) \cap \mathrm{var}(c')|$,

$$\Pr\left(\left|\mathrm{var}(c) \cap \mathrm{var}(c')\right| \geq 3\right) \leq \frac{n(n-1)(n-2)n^{2(k-3)}(k(k-1)(k-2))^2}{n^{2k}} \leq \frac{k^6}{n^3}.$$

Therefore, the probability that there is a pair of clauses c, c' with $|\mathrm{var}(c) \cap \mathrm{var}(c')| \geq 3$ is bounded from above by $\frac{m(m-1)}{2}\frac{k^6}{n^3} \leq \frac{\alpha^2}{2}\frac{k^6}{n} = O\left(\frac{1}{n}\right)$, which finishes the proof. □

We will use the asymmetric version of the Lovász local lemma (LLL), proved by Lovász and originally published in [110]. Before stating this result, let us introduce some notation. Let $\mathcal{P}$ be a finite collection of mutually independent random variables. Let B an event that is a function of the random variables in $\mathcal{P}$. Let $\mathcal{A}$ be a collection of events that are a function of the random variables in $\mathcal{P}$. We define $\Gamma(B)$ as the set of events $A \in \mathcal{A}$ such that $A \neq B$ and A and B are not independent. In this setting, $\Pr_P(B)$ is the probability that the event B holds when sampling all the random variables in $\mathcal{P}$.

Theorem 4.18 (Asymmetric Lovász local lemma, [64, Theorems 1.1 and 2.1])**.** *Let $\mathcal{P}$ be a finite collection of mutually independent random variables. Let $\mathcal{A}$ be a collection of events that are a function of the random variables in $\mathcal{P}$. If there exists a function $x : \mathcal{A} \to (0,1)$ such that, for all $A \in \mathcal{A}$, we have*

$$\Pr_P(A) \leq x(A) \prod_{N \in \Gamma(A)} (1 - x(N)),$$

then $\Pr_P\left(\bigcap_{A \in \mathcal{A}} \overline{A}\right) > 0$. Furthermore, for any event B that is a function of the random variables in $\mathcal{P}$, we have

$$\Pr_P\left(B \,\middle|\, \bigcap_{A \in \mathcal{A}} \overline{A}\right) \leq \Pr_P(B) \prod_{A \in \Gamma(B)} (1 - x(A))^{-1}.$$

We are going to apply the LLL in Lemma 4.21 to find an $(r_0 - \delta, r_0, r_0, 2r_0)$-marking of Φ (Definition 4.3), w.h.p. over the choice of the random formula, for some appropriate $r_0 \in (0,1)$. Before proving Lemma 4.21, let us highlight how strong the properties of a marking are. First, the fact that a set of marked variables is ρ-distributed (Definition 4.3) will allow us to find, w.h.p. over the choice of Φ, a good amount of marked variables in any set of clauses, even if the set includes bad clauses, see Lemma 4.28 for a precise statement. This result is an essential ingredient in our proofs. Secondly, as long as the control variables are left unassigned, the marginals of the marked and auxiliary variables will be near $1/2$ as a consequence of the LLL, as we show later in this section (Lemma 4.23). We remark that, in the definition of ρ-distributed set of variables, we ask for $|\mathrm{var}(c) \cap V| \geq \rho(k-3)$ instead of $|\mathrm{var}(c) \cap V| \geq \rho k$ to account for the fact that w.h.p. a good clause has at most a repeated variable (Lemma 4.17) and at most two bad variables (Proposition 4.2), which will come up in the proofs presented in this section. First, we need the following definition.

Definition 4.19 ($\Phi_{\mathrm{good}}(r)$, $\Phi_{\mathrm{bad}}(r)$)**.** *Let $r \in (0,1)$. Let $\Phi = (\mathcal{V}, \mathcal{C})$ be a k-CNF formula. Let $\Phi_{\mathrm{good}}(r) = (\mathcal{V}_{\mathrm{good}}(r), \mathcal{C}_{\mathrm{good}}(r))$ be the CNF formula obtained by taking the good clauses of Φ and ignoring the bad variables appearing in them. Let $\Phi_{\mathrm{bad}}(r)$ be the k-CNF formula with variables $\mathcal{V}_{\mathrm{bad}}(r)$ and clauses $\mathcal{C}_{\mathrm{bad}}(r)$.*

Note that in $G_{\Phi_{\mathrm{good}}(r)}$ two clauses c_1 and c_2 in $\mathcal{C}_{\mathrm{good}}$ are adjacent if and only if $\mathrm{var}(c_1) \cap \mathrm{var}(c_2) \cap \mathcal{V}_{\mathrm{good}} \neq \emptyset$. By definition of good variables, the maximum degree in $G_{\Phi_{\mathrm{good}}(r)}$ is at most $k(\Delta_r - 1)$, which will be important when applying the LLL. We also need the following version of Chernoff's bounds.

Lemma 4.20 (Chernoff's bounds - [98, Theorem 2.1 and Corollary 4.1])**.** *Let $n \in \mathbb{N}$, $p \in [0,1]$, and let $X_1, \ldots, X_n$ be n independent random variables with $X_j \in \{0,1\}$ and $\Pr(X_j = 1) = p$ for all $j = 1, \ldots, n$. Let $X = \sum_{j=1}^{n} X_j$. Then, for any $t \in (p,1)$ and any $s \in (0,p)$, we have $\Pr(X \geq tn) \leq e^{-D(t,p)n}$ and $\Pr(X \leq sn) \leq e^{-D(s,p)n}$, where, for reals $x, y \in (0,1)$, $D(x,y) := x\log(x/y) + (1-x)\log((1-x)/(1-y))$ is the Kullback-Leibler divergence.*

We can now state the main result of this section. The Lovász local lemma ideas in the proof of Lemma 4.21 are standard in the literature since the work of Moitra [93] but the quantities involved are adapted to our setting.

Lemma 4.21. *There is a positive integer k_0 such that for any $k \geq k_0$ and any density α with $\alpha \leq 2^{(r_0-\delta)k}/k^3$ the following holds w.h.p. over the choice of the random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$:*

1. *there exists a partial assignment of bad variables that satisfies all bad clauses;*
2. *there exists an $(r_0 - \delta, r_0, r_0, 2r_0)$-marking of Φ. Furthermore, for any $p \in (0,1)$, such an $(r_0 - \delta, r_0, r_0, 2r_0)$-marking can be computed with probability at least $1-p$ in time $O(n\log(1/p))$.*

Proof. In this proof we set $r = r_0 - \delta$. We note that for any $k \geq 4$ our density $\alpha \leq 2^{(r_0-\delta)k}/k^3$ is below the threshold $c_k > 1.3836 \cdot 2^k/k$ established in [46, Theorem 1.3]. For densities below this threshold, w.h.p. over the choice of Φ, there is a satisfying assignment for Φ. When Φ is satisfiable, we claim that there is an assignment of the bad variables that satisfies all bad clauses. Indeed, all the variables in bad clauses are bad (Proposition 4.2) and, thus, the restriction of a satisfying assignment to $\mathcal{V}_{\text{bad}}(r)$ must satisfy all the bad clauses. In the rest of this proof we show that assertion 2 also holds.

In view of Lemma 4.17, we may assume that $|\text{var}(c)| \geq k-1$ for all $c \in \mathcal{C}$. Let us find the $(r, r_0, r_0, 2r_0)$-marking $(\mathcal{V}_{\text{m}}, \mathcal{V}_{\text{a}}, \mathcal{V}_{\text{c}})$. If all clauses are bad, then we set $\mathcal{V}_{\text{c}} = \mathcal{V}$, $\mathcal{V}_{\text{m}} = \emptyset$ and $\mathcal{V}_{\text{a}} = \emptyset$. This is trivially an $(r, r_0, r_0, 2r_0)$-marking for Φ. In the rest of the proof we assume that there are good variables. We study the following probability space. For each good variable v, we set v as "marked" with probability $\beta \in (0, 1/2)$, "auxiliary" with probability β and "control" with probability $1 - 2\beta$. This decision is made independently for each good variable. Each bad variable is set as "control". Let $\mathcal{P}$ be the set $\{P_v : v \in \mathcal{V}_{\text{good}}(r)\}$, where P_v is the random choice made in this experiment for v. Let $\mathcal{V}_{\text{m}}$ be the set of marked variables, let $\mathcal{V}_{\text{a}}$ be the set of auxiliary variables, and let $\mathcal{V}_{\text{c}}$ be the set of control variables obtained by running this experiment. For each clause $c \in \mathcal{C}_{\text{good}}(r)$, let A_c be the event that c has less than $r_0(k-3)$ marked variables or less than $r_0(k-3)$ auxiliary variables or less than $2r_0(k-3)$ good control variables. We are going to apply the LLL on the formula $\Phi_{\text{good}}(r)$ so as to show that $\Pr(\bigcap_{c \in \mathcal{C}_{\text{good}}(r)} \overline{A_c}) > 0$. For each $c \in \mathcal{C}_{\text{good}}(r)$, in view of Proposition 4.2 and the fact that $|\text{var}(c)| \geq k-1$, we have $|\text{var}(c) \cap \mathcal{V}_{\text{good}}(r)| \geq k-3$. Hence, we can apply the Chernoff bound given in Lemma 4.20 with $n = |\text{var}(c) \cap \mathcal{V}_{\text{good}}(r)|$, $p = \beta$ and $s = r_0$ to obtain, for any choice $V \in \{\mathcal{V}_{\text{m}}, \mathcal{V}_{\text{a}}\}$,

$$\Pr_P(|\text{var}(c) \cap V| < r_0(k-3)) \leq e^{-D(r_0,\beta)(k-3)}.$$

When $V = \mathcal{V}_{\mathrm{c}} \setminus \mathcal{V}_{\mathrm{bad}}$, $n = |\mathrm{var}(c) \cap \mathcal{V}_{\mathrm{good}}(r)|$, $p = 1 - 2\beta$ and $s = 2r_0$ we obtain

$$\Pr_P\left(|\mathrm{var}(c) \cap V| < 2r_0(k-3)\right) \leq e^{-D(2r_0, 1-2\beta)(k-3)}.$$

We have chosen r_0 to be as large as possible under the restrictions that $D(r_0, \beta) \geq r_0 \log 2$ and $D(2r_0, 1-2\beta) \geq r_0 \log 2$. The values $\beta = 0.571027$ and $r_0 = 0.117841$ satisfy these restrictions. We conclude that

$$\Pr_P\left(A_c\right) \leq 2 \cdot e^{-D(r_0,\beta)(k-3)} + e^{-D(2r_0,1-2\beta)(k-3)} \leq 3 \cdot 2^{-r_0(k-3)}.$$

Let $\Delta' = 2^{r_0(k-3)}/(3e^2k)$ and let $x(A_c) = 1/(k\Delta')$ for all $c \in \mathcal{C}_{\mathrm{good}}(r)$. We check that x satisfies the condition of the LLL for $\mathcal{P}$ and $\mathcal{A} = \{A_c : c \in \mathcal{C}_{\mathrm{good}}(r)\}$. For $k \geq 43$, $1/(k\Delta') \in (0,1)$ and thus $x(A_c) \in (0,1)$ for all $c \in \mathcal{C}_{\mathrm{good}}(r)$. We note that $\Gamma(A_c) = \{A_{c'} : c' \in \mathcal{C}_{\mathrm{good}}(r), c' \neq c, \mathrm{var}(c') \cap \mathrm{var}(c) \cap \mathcal{V}_{\mathrm{good}}(r) \neq \emptyset\}$. The graph $G_{\Phi_{\mathrm{good}}(r)}$, given in Definition 4.11, has maximum degree at most $k(\Delta_r - 1)$, so $|\Gamma(A_c)| \leq k(\Delta_r - 1) \leq k\Delta'$, where the latter inequality holds for large enough k as $\Delta_r = \lceil 2^{rk} \rceil$ and $r = r_0 - \delta$. Therefore, we have

$$x(A_c) \prod_{N \in \Gamma(A_c)} (1 - x(N)) \geq \frac{1}{k\Delta'} \left(1 - \frac{1}{k\Delta'}\right)^{k\Delta'} \geq \frac{1}{e^2 k\Delta'} = 3 \cdot 2^{-r_0(k-3)}, \tag{4.7}$$

where we used $(1 - 1/z)^z \geq e^{-2}$ for all $z \geq 2$ in the second inequality. Thus,

$$x(A_c) \prod_{N \in \Gamma(A_c)} (1 - x(N)) \geq 3 \cdot 2^{-r_0(k-3)} \geq \Pr_P\left(A_c\right).$$

We conclude that, by the LLL, $\Pr_P\left(\bigcap_{c \in \mathcal{C}_{\mathrm{good}}(r)} \overline{A_c}\right) > 0$, so there exists a partition $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{a}}, \mathcal{V}_{\mathrm{c}})$ of the variables of Φ such that $\mathcal{V}_{\mathrm{bad}}(r) \subseteq \mathcal{V}_{\mathrm{c}}$ and each good clause contains at least $r_0(k-3)$ marked variables, $r_0(k-3)$ auxiliary variables and $2r_0(k-3)$ good control variables. That is, $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{a}}, \mathcal{V}_{\mathrm{c}})$ satisfies Definition 4.3 for $r = r_0 - \delta$, $r_{\mathrm{m}} = r_0$, $r_{\mathrm{a}} = r_0$, and $r_{\mathrm{c}} = 2r_0$. Moreover, with probability at least $1 - \delta$, this partition can be computed in $4n\alpha\Delta' k \log(1/\delta)$ steps with the algorithm of Moser and Tardos [97]. □

We now give the marking result that we use in our connectivity results, which holds for densities at most $2^{(r_1-\delta)k}/k^3$, where $r_1 = 0.227092$. The larger density threshold comes from the fact that the marking result is less strong – we do not require auxiliary variables nor a high number of good control variables in every clause.

Lemma 4.22. *There is a positive integer k_0 such that for any $k \geq k_0$ and any density α with $\alpha \leq 2^{(r_1-\delta)k}/k^3$ the following holds w.h.p. over the choice of the random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$:*

1. *there exists a partial assignment of bad variables that satisfies all bad clauses;*
2. *there exists an $(r_1 - \delta, r_1, 0, r_1)$-marking of Φ. Furthermore, for any $p \in (0,1)$, such an $(r_1 - \delta, r_1, 0, r_1)$-marking can be computed with probability at least $1 - p$ in time $O(n \log(1/p))$.*

Proof. The proof is analogous to that of Lemma 4.21. Here we explain the main differences. First, we set $r = r_1 - \delta$ instead of $r = r_0 - \delta$. The second difference is that we study the following probability space: each good variable v is set as "marked" with probability β and "control" with probability $1-\beta$. We let A_c be the event that c has less than $r_1(k-3)$ marked variables or less than $r_1(k-3)$ good control variables. A Chernoff bound as in the proof of Lemma 4.21 gives

$$\Pr_P(A_c) \leq e^{-D(r_1,\beta)(k-3)} + e^{-D(r_1,1-\beta)(k-3)} \leq 2 \cdot 2^{-r_1(k-3)},$$

where we chose r_1 as large as possible so that $D(r_1,\beta) \geq r_1 \log 2$ and $D(r_1,1-\beta) \geq r_1 \log 2$. The choices $\beta = 1/2$ and $r_1 = 0.227092$ satisfy these restrictions. We let $\Delta' = 2^{r_1(k-3)}/(3e^2k)$ and let $x(A_c) = 1/(k\Delta')$ for all $c \in \mathcal{C}_{\text{good}}(r)$. It remains to check that we can apply the asymmetric LLL on the formula $\Phi_{\text{good}}(r)$ to conclude that $\Pr(\bigcap_{c\in\mathcal{C}_{\text{good}}(r)} \overline{A_c}) > 0$. This was done in equation (4.7) in Lemma 4.21. We note that the bound given in (4.7) also holds in our current setting if we replace r_0 by r_1. We find that $x(A_c)\prod_{N\in\Gamma(A_c)}(1-x(N)) \geq 3 \cdot 2^{-r_1(k-3)} \geq \Pr_P(A_c)$ and, thus, there exists a partition $(\mathcal{V}_{\text{m}}, \mathcal{V}_{\text{a}}, \mathcal{V}_{\text{c}})$ of the variables of Φ such that $\mathcal{V}_{\text{bad}}(r) \subseteq \mathcal{V}_{\text{c}}$, $\mathcal{V}_{\text{a}} = \emptyset$, and each good clause contains at least $r_1(k-3)$ marked variables and at least $r_1(k-3)$ good control variables. □

In the remaining of this section we bound the marginals of μ_Ω (recall that μ_Ω is the uniform distribution over the satisfying assignments of the formula Φ, Definition 4.4) on any marked and auxiliary variable. In fact, we prove the stronger result that the marginal distribution of μ_Ω on $\mathcal{V}_{\text{m}} \cup \mathcal{V}_{\text{a}}$ is ε-uniform, i.e., very close to the uniform distribution, see Definition 4.7. We give a bound for each one of the markings established in Lemmas 4.21 and 4.22. Here we write $\Lambda_1 \cup \Lambda_2$ for the combined assignment of Λ_1 and Λ_2.

Lemma 4.23. *Let $\Phi = (\mathcal{V}, \mathcal{C})$ be a satisfiable k-CNF formula. The following claims hold.*

1. *Let $r = r_0 - \delta$ and let $(\mathcal{V}_{\text{m}}, \mathcal{V}_{\text{a}}, \mathcal{V}_{\text{c}})$ be a $(r, r_0, r_0, 2r_0)$-marking of Φ. Then for any satisfying assignment Λ_{bad} of $\Phi_{\text{bad}}(r)$, any assignment $\Lambda\colon S \to \{\mathsf{F},\mathsf{T}\}$ where $S \subseteq \mathcal{V}_{\text{m}} \cup \mathcal{V}_{\text{a}}$, and any $v \in \mathcal{V}_{\text{good}}(r) \setminus S$ we have*

$$\max\left\{\Pr_{\mu_\Omega}(v \mapsto \mathsf{F} |\, \Lambda \cup \Lambda_{\text{bad}}), \Pr_{\mu_\Omega}(v \mapsto \mathsf{T} |\, \Lambda \cup \Lambda_{\text{bad}})\right\} \leq \frac{1}{2}\exp\left(\frac{1}{k2^{r_0 k}}\right).$$

   In particular, the distribution $\mu_\Omega|_{\mathcal{V}_{\text{m}}\cup\mathcal{V}_{\text{a}}}$ is $(2^{-r_0k}/k)$-uniform.

2. *Let $r = r_1 - \delta$ and let $(\mathcal{V}_{\text{m}}, \emptyset, \mathcal{V}_{\text{c}})$ be a $(r, r_1, \emptyset, r_1)$-marking of Φ. Then, for any satisfying assignment Λ_{bad} of $\Phi_{\text{bad}}(r)$, any assignment $\Lambda\colon S \to \{\mathsf{F},\mathsf{T}\}$ where $S \subseteq \mathcal{V}_{\text{m}}$, and any $v \in \mathcal{V}_{\text{good}}(r) \setminus S$ we have*

$$\max\left\{\Pr_{\mu_\Omega}(v \mapsto \mathsf{F} |\, \Lambda \cup \Lambda_{\text{bad}}), \Pr_{\mu_\Omega}(v \mapsto \mathsf{T} |\, \Lambda \cup \Lambda_{\text{bad}})\right\} \leq \frac{1}{2}\exp\left(\frac{1}{k}\right).$$

   In particular, the distribution $\mu_\Omega|_{\mathcal{V}_{\text{m}}}$ is $(1/k)$-uniform.

Proof. We prove each one of the claims separately. The proofs are analogous so for the second claim we only highlight the differences in the proof.

1. Here $r = r_0 - \delta$. Let Λ_{bad} be an assignment of bad variables that satisfies all bad clauses. Let $S \subseteq \mathcal{V}_{\mathrm{m}} \cup \mathcal{V}_{\mathrm{a}}$, let Λ be an assignment of S to $\{\mathsf{F}, \mathsf{T}\}$, and let $v \in \mathcal{V}_{\mathrm{good}}(r) \setminus S$. We note that $\Pr_{\mu_{\Omega^\tau}}(\cdot) = \Pr_{\mu_\Omega}(\cdot\,|\tau)$ for any assignment τ of some variables. In light of this observation, we are going to prove that

$$\max\left\{\Pr_{\mu_{\Omega^{\Lambda\cup\Lambda_{\mathrm{bad}}}}}(v \mapsto \mathsf{F}), \Pr_{\mu_{\Omega^{\Lambda\cup\Lambda_{\mathrm{bad}}}}}(v \mapsto \mathsf{T})\right\} \leq \frac{1}{2}\exp\left(\frac{1}{k2^{r_0 k}}\right). \tag{4.8}$$

We apply the LLL to the formula $\Phi' := \Phi^{\Lambda\cup\Lambda_{\mathrm{bad}}}$ as follows. Let $\mathcal{V}'$ and $\mathcal{C}'$ be the sets of variables and clauses of Φ'. Note that, $\mathcal{V}' \subseteq \mathcal{V}_{\mathrm{good}}(r)$, $\mathcal{C}' \subseteq \mathcal{C}_{\mathrm{good}}(r)$ and $G_{\Phi'}$ is a subgraph of $G_{\Phi_{\mathrm{good}}(r)}$ as all bad variables have been assigned a value and all bad clauses have been satisfied. We set $P_v = \sigma(v)$ for all $v \in \mathcal{V}'$, where $\sigma\colon \mathcal{V}' \to \{\mathsf{F}, \mathsf{T}\}$ is chosen uniformly at random from the set of assignments $\mathcal{V}' \to \{\mathsf{F}, \mathsf{T}\}$, and $\mathcal{P} = \{P_v : v \in \mathcal{V}'\}$. We define the set $\mathcal{A}$ as the set containing for all $c \in \mathcal{C}'$ the event $A_c =$ "the clause c is not satisfied by the random assignment σ". By the definition of $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{a}}, \mathcal{V}_{\mathrm{c}})$, there are at least $2r_0(k-3)$ good control variables in c. Since good control variables are not assigned a value by $\Lambda \cup \Lambda_{\mathrm{bad}}$ and, thus, they are in $\mathcal{V}'$, we have $\Pr_P(A_c) \leq 2^{-2r_0(k-3)}$. Recall that $\Delta_r = \lceil 2^{(r_0-\delta)k}\rceil$ (Definition 4.1). Let $\Delta' = 2^{2r_0(k-3)}/(e^2k)$ and let $x(A_c) = \frac{1}{k\Delta_0}$ for all $c \in \mathcal{C}'$. Let us show that x satisfies the LLL condition in this setting. In view of $\Gamma(A_c) = \{A_{c'} : c' \in \mathcal{C}', c' \neq c, \mathrm{var}(c) \cap \mathrm{var}(c') \cap \mathcal{V}' \neq \emptyset\}$, which can be identified with a subset of the neighbours of c in $G_{\Phi_{\mathrm{good}}(r)}$, and $|\Gamma(A_c)| \leq k\Delta_r \leq k\Delta'$ for large enough k, we find that

$$x(A_c)\prod_{N\in\Gamma(A_c)}(1-x(N)) \geq \frac{1}{k\Delta'}\left(1-\frac{1}{k\Delta'}\right)^{k\Delta'} \geq \frac{1}{e^2k\Delta'} = 2^{-2r_0(k-3)} \geq \Pr_P(A_c),$$

where we used $(1-1/z)^z \geq e^{-2}$ for all $z \geq 2$. Let $A = \{v \mapsto \mathsf{T}\} := \{\sigma\colon \mathcal{V}' \to \{\mathsf{F}, \mathsf{T}\}$ with $\sigma(v) = \mathsf{T}\}$. In Φ', we have $\Gamma(A) = \{A_c : c \in \mathcal{C}', v \in \mathrm{var}(c)\}$, so $|\Gamma(A)| < \Delta_r$. By the LLL, we obtain

$$\Pr_P\left(v \mapsto \mathsf{T}\,\middle|\,\bigcap\nolimits_{c\in\mathcal{C}'}\overline{A_c}\right) \leq \frac{1}{2}\prod_{N\in\Gamma(A)}\left(1-x(N)\right)^{-1} \leq \frac{1}{2}\left(1-\frac{1}{k\Delta'}\right)^{-(\Delta_r-1)}.$$

For $x > 1$, we have $(1-1/x)^{-1} = 1 + 1/(x-1) \leq \exp(1/(x-1))$. We find that

$$\Pr_P\left(v \mapsto \mathsf{T}\,\middle|\,\bigcap\nolimits_{c\in\mathcal{C}'}\overline{A_c}\right) \leq \frac{1}{2}\exp\left(\frac{\Delta_r - 1}{k\Delta' - 1}\right) \leq \frac{1}{2}\exp\left(\frac{1}{k2^{r_0k}}\right),$$

where in the latter inequality we used $(p-j)/(q-j) \leq p/q$ for all $0 < j < p \leq q$ and the fact that $\Delta_r = \lceil 2^{(r_0-\delta)k}\rceil \leq 2^{-r_0k}\cdot 2^{2r_0(k-3)}/(e^2k) = 2^{-r_0k}\Delta'$ for large enough k. We note that $\Pr_{\mu_{\Omega^{\Lambda\cup\Lambda_{\mathrm{bad}}}}}(\cdot) = \Pr_P\left(\cdot\,|\bigcap_{c\in\mathcal{C}'}\overline{A_c}\right)$, which completes the proof of one of the upper bounds of (4.8). The other upper bound is proved analogously by applying the LLL with $A = \{v \mapsto \mathsf{F}\}$. Finally, we conclude that the distribution $\mu_\Omega|_{\mathcal{V}_{\mathrm{m}}\cup\mathcal{V}_{\mathrm{a}}}$ is $(2^{-r_0k}/k)$-uniform by the arbitrary choice of Λ_{bad} and the law of total probability, see Definition 4.7.

2. The proof is analogous. The only changes are $r = r_1 - \delta$, $\Delta' = 2^{r_1(k-3)}/(e^2 k)$, and the fact that, since each good clause has at least $r_1(k-3)$ good control variables, we have $\Pr_P(A_c) \leq 2^{-r_1(k-3)}$. This time we have $x(A_c)\prod_{N\in\Gamma(A_c)}(1-x(N)) \geq \frac{1}{e^2 k\Delta'} \geq \Pr_P(A_c)$, which justifies our choice of Δ'. Thus, we can apply the LLL, and the conclusion this time becomes
$$\Pr_P\left(v \mapsto \mathsf{T} \,\middle|\, \bigcap\nolimits_{c\in\mathcal{C}'} \overline{A_c}\right) \leq \frac{1}{2}\exp\left(\frac{\Delta_r - 1}{k\Delta' - 1}\right) \leq \frac{1}{2}\exp\left(\frac{1}{k}\right),$$
where in the latter inequality we used $(p-j)/(q-j) \leq p/q$ for all $0 < j < p \leq q$ and the fact that $\Delta_r = \lceil 2^{(r_1-\delta)k}\rceil \leq 2^{r_1(k-3)}/(e^2 k) = \Delta'$ for large enough k. □

The $(1/k)$-uniform property proved in Lemma 4.23 is remarkably strong: as long as the control variables are left unassigned, the rest of the variables have marginals close to $1/2$, even if some of the marked and auxiliary variables are pinned / have already been assigned a value. This property is used several times in this work and will allow us to prove that, for any pinning of some marked variables, the influences between marked variables are bounded. In the following corollary we extend Lemma 4.23 to the distributions computed by the Glauber dynamics on the marked variables.

Corollary 4.24. *Let $r = r_0 - \delta$. Let $\Phi = (\mathcal{V}, \mathcal{C})$ be a satisfiable k-CNF formula that has an $(r, r_0, r_0, 2r_0)$-marking $(\mathcal{V}_\mathrm{m}, \mathcal{V}_\mathrm{a}, \mathcal{V}_\mathrm{c})$. Let ρ be an integer with $1 \leq \rho < |\mathcal{V}_\mathrm{m}|$. Let t be a non-negative integer and let X_t be the (random) assignment obtained after running the ρ-uniform-block Glauber dynamics on the marked variables for t steps, starting on an assignment X_0 that is chosen uniformly at random. Then the probability distribution of X_t is $(2^{-r_0 k}/k)$-uniform.*

Proof. Let $\varepsilon = (2^{-r_0 k}/k)$. Let $V_1, V_2, \ldots$, be a possible choice of sets of marked variables to be updated when running the ρ-uniform-block Glauber dynamics. We are going to prove that, conditioning on this choice of sets of variables, the probability distribution of X_t is ε-uniform. Note that by the law of total probability and the fact that the choice of $V_1, V_2, \ldots$ is arbitrary, this is enough to conclude the result. We carry out the proof by induction on t. Let π_t be the probability distribution of X_t. As π_0 is the uniform distribution over assignments on $\mathcal{V}_\mathrm{m}$, the claim holds for $t = 0$. Let us now assume that π_{t-1} is ε-uniform and let us prove that this is also the case for π_t. To show the desired uniformity of π_t (cf. Definition 4.7), consider arbitrary $v \in \mathcal{V}_\mathrm{m}$ and $\Lambda\colon \mathcal{V}_\mathrm{m} \setminus \{v\} \to \{\mathsf{F}, \mathsf{T}\}$, we need to bound $\Pr_{\pi_t}(v \mapsto \mathsf{F} | \Lambda)$ and $\Pr_{\pi_t}(v \mapsto \mathsf{T} | \Lambda)$. We distinguish two cases:

- Case $v \in V_t$. By definition of the Glauber dynamics, the values of X_t on V_t are obtained by sampling from the distribution μ_Ω conditioned on the restriction of X_{t-1} to $\mathcal{V}_\mathrm{m} \setminus V_t$. Thus, we have $\Pr_{\pi_t}(v \mapsto \mathsf{F} | \Lambda) = \Pr_{\mu_{\Omega^\Lambda}}(v \mapsto \mathsf{F})$ since the conditioning involving Λ sets all the marked variables other than v. As $\mu_\Omega|_{\mathcal{V}_\mathrm{m}\cup\mathcal{V}_\mathrm{a}}$ is ε-uniform by Lemma 4.23, we conclude that $\Pr_{\pi_t}(v \mapsto \mathsf{F} | \Lambda) = \Pr_{\mu_{\Omega^\Lambda}}(v \mapsto \mathsf{F}) \leq \frac{1}{2}\exp(\varepsilon)$. The same bound holds for $v \mapsto \mathsf{T}$.
- Case $v \notin V_t$. If v is not updated in steps 1 through t, then $\Pr_{\pi_t}(v \mapsto \mathsf{F} | \Lambda) = \Pr_{\pi_0}(v \mapsto \mathsf{F}) = 1/2$. Otherwise, let j be the largest integer with $j < t$ such that $v \in V_j$. Let Λ_j be the

restriction of Λ to $\mathcal{V}_{\mathrm{m}} \setminus \bigcup_{i\in\{j+1,j+2,\ldots,t\}} V_i$. By the induction hypothesis, $\Pr_{\pi_t}(v \mapsto \mathsf{F} | \Lambda) = \Pr_{\pi_j}(v \mapsto \mathsf{F} | \Lambda_j) \leq (1/2)\exp(\varepsilon)$. The same bound holds for $v \mapsto \mathsf{T}$.

As both cases are exhaustive, the proof is concluded. □

Previous work on counting/sampling satisfying assignments of k-CNF formulae does not require the use of auxiliary variables, so the marking used is of the form $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{c}})$. Here auxiliary variables play an essential role in bounding the influences between marked variables as we illustrated in Section 4.1. In order for this approach to be successful, we have to show that a large proportion of the variables are marked. We conclude this section with the following bound on the size of $\mathcal{V}_{\mathrm{m}}$.

Corollary 4.25. *Let $r \in (0, 1/(2\log 2))$. There is an integer k_0 such that for any $k \geq k_0$ and any density α with $\alpha \leq \Delta_r/k^3$ the following holds w.h.p. over the choice of the random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. For any $\rho \in (0,1)$ and any set of good variables V that is ρ-distributed we have $|V| \geq (\rho - \delta)(k\alpha/\Delta_r)n$.*

Proof. W.h.p. over the choice of Φ, by Lemma 4.16 we have $|\mathcal{C}_{\mathrm{bad}}(r)| \leq 2(\alpha/\Delta_r)n/2^{k^{10}} \leq \alpha n/4^k$, so $|\mathcal{C}_{\mathrm{good}}(r)| \geq |\mathcal{C}| - \alpha n/4^k \geq \alpha n - 1 - \alpha n/4^k = \alpha n(1 - 1/4^k) - 1$. Since V is ρ-distributed, counting repetitions, there are at least $\rho(k-3)|\mathcal{C}_{\mathrm{good}}(r)|$ occurrences of the variables of V in the good clauses of Φ. Each good variable occurs in at most Δ_r good clauses, so we find that

$$|V| \geq \frac{\rho(k-3)|\mathcal{C}_{\mathrm{good}}(r)|}{\Delta_r} \geq \frac{\rho(k-3)}{\Delta_r}\left(\alpha n\left(1 - \frac{1}{4^k}\right) - 1\right) \geq \frac{\rho(k-4)}{\Delta_r}(\alpha n - 1),$$

which is at least $(\rho - \delta)(k\alpha/\Delta_r)n$ for large enough k. □

4.4 Analysis of the connected components of Φ^Λ

In this section we prove Lemma 4.12, which bounds the size of the connected components of Φ^Λ, where Λ is drawn from a $(1/k)$-uniform distribution over an $(r+\delta)$-distributed set of good variables. In order to carry out this proof, we have to understand the structure of logarithmic-sized sets of clauses of the random k-CNF formula Φ. Section 4.4.1 is devoted to this purpose. In Section 4.4.2 we apply the results of Section 4.4.1 to obtain a lower bound of the number of marked/auxiliary variables in logarithmic-sized sets of clauses. Finally, in Section 4.4.3 we complete the proof of Lemma 4.12.

4.4.1 Logarithmic-sized sets of clauses in the random k-SAT model

A connected graph $H = (V, E)$ has *tree-excess* $c \in \mathbb{Z}_{\geq 0}$ if $|E| = c + |V| - 1$. It turns out that, w.h.p. over the choice of Φ, small connected sets of clauses of Φ have tree-excess bounded by a quantity that only depends on k and the density α. This property is established in Lemma 4.26 and is essential to our proofs.

Lemma 4.26. *Let $k \geq 3$ be an integer. Let $b > 0$ and $\alpha > 0$ be real numbers. W.h.p. over the choice of the random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$, every connected subset of clauses with size at most $b \log(n)$ has tree-excess at most $c := \max\{1, 2b \log(ek^2\alpha)\}$.*

Proof. Let n be the number of variables and m be the number of clauses of Φ, so $m/n \leq \alpha$. Note that the probability that two clauses of Φ are not disjoint is at most k^2/n. Let $\ell \in \{1, 2, \ldots, \lfloor b \log(n) \rfloor\}$. We upper bound the probability that there is a connected subset of clauses of size ℓ with tree-excess at least $c + 1$ by

$$\binom{m}{\ell} \ell^{\ell-2} \binom{\ell(\ell-1)/2}{c+1} \left(\frac{k^2}{n}\right)^{\ell+c}, \tag{4.9}$$

where the factors appearing are the following ones:

- $\binom{m}{\ell}$ is the number of subsets of clauses of size ℓ;
- $\ell^{\ell-2}$ is the number of trees on ℓ labelled vertices;
- $\binom{\ell(\ell-1)/2}{c+1}$ is the number of ways to pick $c + 1$ pairs of distinct clauses of a set of size ℓ;
- $\left(k^2/n\right)^{\ell+c}$ is an upper bound of the probability that all the edges chosen in the two previous items appear in the graph G_Φ.

We are going to show that the probability given in (4.9) is $O(n^{-c/4})$, where the hidden constant only depends on k. If this holds, by a union bound over $\ell \in \{1, 2, \ldots, \lfloor b \log(n) \rfloor\}$, we would find that the probability that there is a connected subset of clauses of Φ with size at most $b \log(n)$ and tree-excess at least $c + 1$ is $O(b \log(n) n^{-c/4}) = o(1)$. This would complete the proof. Using the inequality $\binom{p}{q} \leq (ep/q)^q$ and $m/n \leq \alpha$ we can bound (4.9) by

$$\begin{aligned}
\left(\frac{em}{\ell}\right)^{\ell} \ell^{\ell-2} \left(\frac{e\ell(\ell-1)/2}{c+1}\right)^{c+1} \left(\frac{k^2}{n}\right)^{\ell+c} &\leq \left(\frac{em}{\ell}\right)^{\ell} \ell^{\ell-2} \left(\frac{e\ell^2/2}{c+1}\right)^{c+1} \left(\frac{k^2}{n}\right)^{\ell+c} \\
&= \left(\frac{e}{2c+2}\right)^{c+1} \left(\frac{emk^2}{n}\right)^{\ell} \left(\frac{k^2\ell^2}{n}\right)^{c} \\
&\leq \left(\frac{e}{2c+2}\right)^{c+1} \left(ek^2\alpha\right)^{\ell} \left(\frac{k^2\ell^2}{n}\right)^{c}.
\end{aligned} \tag{4.10}$$

Now we distinguish two cases:

- Case when $ek^2\alpha \leq 1$. We have $c = 1$ by definition. Thus, (4.10) can be further bounded by
$$\left(\frac{e}{2c+2}\right)^{c+1} \left(\frac{k^2\ell^2}{n}\right)^{c} = O\left(\frac{(\log n)^2}{n}\right) = O\left(n^{-c/4}\right)$$
as we wanted.

- Case when $ek^2\alpha > 1$. Then, as $\ell \leq b \log n$ and $b \log(ek^2\alpha) \leq c/2$ by definition, we have
$$\left(ek^2\alpha\right)^{\ell} \leq \left(ek^2\alpha\right)^{b \log n} = n^{b \log(ek^2\alpha)} \leq n^{c/2}.$$

We conclude that (4.10) can be further bounded by

$$\left(\frac{e}{2c+2}\right)^{c+1}\left(\frac{k^2\ell^2}{\sqrt{n}}\right)^{c} = \left(\frac{e}{2c+2}\right)^{c+1}\left(\frac{k^4\ell^4}{n}\right)^{c/2} = O\left(n^{-c/4}\right)$$

as we wanted, where we used $c > 0$. □

Recall that in Lemma 4.15 we established that, in sets of clauses that have at least $2k^4 \log n$ variables, the number of bad clauses of Φ is not too large. We aim to apply Lemma 4.15 to logarithmic-sized sets of clauses. In general, $|Y|$ might be significantly larger than $|\mathrm{var}(Y)|$, so it is not clear how to apply Lemma 4.15. However, in the random k-CNF formula setting the following holds.

Lemma 4.27. *Let $k \geq 3$ be an integer and let $a > 0$ and $\alpha > 0$ be real numbers. W.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$, for every set of clauses Y with $|Y| \geq a \log n$, we have $|\mathrm{var}(Y)| \geq a \log n$.*

Proof. Let $\ell := \lceil a \log n \rceil - 1$ and let $m = \lfloor \alpha n \rfloor$. We prove the equivalent statement that, w.h.p. over the choice of Φ, for every set of clauses Y with $|\mathrm{var}(Y)| \leq \ell$, we have $|Y| \leq \ell$. We note that if there is a set of clauses Y with $|\mathrm{var}(Y)| \leq \ell$ and $|Y| > \ell$, then for any subset Y' of Y with $|Y'| = \ell + 1$ we have $|\mathrm{var}(Y')| \leq |\mathrm{var}(Y)| \leq \ell$. Hence, it suffices to prove that there is no set Y of clauses with $|\mathrm{var}(Y)| \leq \ell$ and $|Y| = \ell + 1$. We can assume n is large enough so that $\ell \leq e \cdot n$.

Let $\mathcal{E}$ be the event that there is a set of clauses Y of size $\ell + 1$ and a set of variables X of size ℓ such that all clauses in Y have all variables in X. Then by a union bound

$$\Pr(\mathcal{E}) \leq \binom{m}{\ell+1}\binom{n}{\ell}\left(\frac{\ell}{n}\right)^{(\ell+1)k},$$

where the first factor is the number of sets Y, the second factor is the number of sets X and the third factor is the probability that all variables in the clauses of Y are in X. From the well-known bound $\binom{p}{q} \leq (ep/q)^q$, we obtain

$$\begin{aligned}\Pr(\mathcal{E}) &\leq \left(\frac{em}{\ell+1}\right)^{\ell+1}\left(\frac{en}{\ell}\right)^{\ell}\left(\frac{\ell}{n}\right)^{(\ell+1)k} \leq \left(\frac{em}{\ell}\right)^{\ell+1}\left(\frac{en}{\ell}\right)^{\ell+1}\left(\frac{\ell}{n}\right)^{(\ell+1)k} \\ &\leq \left(\frac{e\alpha n}{\ell}\right)^{\ell+1}\left(\frac{en}{\ell}\right)^{\ell+1}\left(\frac{\ell}{n}\right)^{(\ell+1)k} = \left(e^2\alpha\frac{\ell^{k-2}}{n^{k-2}}\right)^{\ell+1},\end{aligned}$$

which is $O(\log(n)/n)$ because $k \geq 3$ and $\ell = O(\log n)$. □

4.4.2 Number of marked variables in logarithmic-sized sets of clauses

Our results on random k-CNF formulae can now be combined to give a lower bound on the number of marked / auxiliary variables in logarithmic-sized sets of clauses. We prove this result in a more general setting by considering a set of good variables V that is r'-distributed for the formula Φ. The reader can think of V as the set of marked variables or the set of auxiliary variables for one of the markings established in Section 4.3.

Lemma 4.28. *Let $r \in (0, 1/(2\log 2)]$, $r' \in (0,1)$ and $\hat{\delta} \in (0, r)$. There is a positive integer k_0 such that, for any integer $k \geq k_0$, any density $\alpha \leq \Delta_r/k^3$ and any real number b with $2k^4 < b$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. Let V be a set of good variables that is r'-distributed. Then, for every set of clauses Y that is connected in G_Φ such that $2k^4 \log(n) \leq |Y| \leq b\log(n)$, we have $|\mathrm{var}(Y) \cap V| \geq (r' - \hat{\delta})k|Y|$.*

Proof. Let $a = 2k^4$. We apply Lemma 4.15 to find that there is k_1 such that for $k \geq k_1$, w.h.p. over the choice of Φ, for every set of clauses Y that is connected in G_Φ,

$$\text{if } \ |\mathrm{var}(Y)| \geq a\log(n), \text{ then } |Y \cap \mathcal{C}_{\mathrm{bad}}(r)| \leq |Y|/k. \tag{4.11}$$

We apply Lemma 4.27 with $a = 2k^4$ to find that, w.h.p. over the choice of Φ, for every set of clauses Y, we have

$$\text{if } \ |Y| \geq a\log(n), \text{then } |\mathrm{var}(Y)| \geq a\log(n). \tag{4.12}$$

Finally, for any $b > 0$, we apply Lemma 4.26, obtaining that, w.h.p. over the choice of Φ, for every set of clauses Y that is connected in G_Φ,

$$\text{if } |Y| \leq b\log n, \text{ then } Y \text{ has tree-excess at most } c = \max\{1, 2b\log(ek^2\alpha)\} = O(1). \tag{4.13}$$

Let Y be a set of clauses that is connected in G_Φ such that $a\log(n) \leq |Y| \leq b\log(n)$. Then, by (4.12) and (4.11), we have $|Y \cap \mathcal{C}_{\mathrm{good}}(r)| \geq |Y|(1 - 1/k)$. By definition of r'-distributed (Definition 4.3), each good clause has at least $r'(k-3)$ variables in V. As there are at most $|Y| - 1 + c$ edges in G_Φ joining clauses in Y, see (4.13), and two distinct clauses only share at most two variables by Lemma 4.17, we have

$$\begin{aligned} |\mathrm{var}(Y) \cap V| &\geq r'(k-3)\left(1 - \frac{1}{k}\right)|Y| - 2(|Y| + c - 1) \\ &\geq (r'(k-4) - 2)|Y| - 2(c-1). \end{aligned}$$

There is $k_0 \geq k_1$ such that for $k \geq k_0$, we find that, for any set of clauses Y that is connected in G_Φ and has $a\log(n) \leq |Y| \leq b\log(n)$, $|\mathrm{var}(Y) \cap V| \geq (r' - \hat{\delta}/2)k|Y| - 2(c-1)$. Therefore, using $2(c-1) = O(1)$, for large enough n we conclude that $|\mathrm{var}(Y) \cap V| \geq (r' - \hat{\delta})k|Y|$ and the result follows. □

4.4.3 Proof of Lemma 4.12

We use the following result of [49] on the number of connected sets of clauses in G_Φ.

Lemma 4.29 ([49, Lemma 8.6])**.** *Let $\alpha > 0$. W.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$, for any clause c, the number of connected sets of clauses in G_Φ with size $\ell \geq \log n$ containing c is at most $(9k^2\alpha)^\ell$.*

We can now complete the proof of Lemma 4.12. Recall that we will apply this result with $r = r_0 - \delta$ or $r = r_1 - \delta$, where $\delta = 0.00001$.

Lemma 4.12. *Let $r \in (2\delta, 1/(2\log 2)]$. There is an integer $k_0 \geq 3$ such that, for any integer $k \geq k_0$, any density $\alpha \leq 2^{(r-2\delta)k}$, and any real number b with $a := 2k^4 < b$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$.*

Let L be an integer satisfying $a \log n \leq L \leq b \log n$. Let V be a set of good variables of Φ that is $(r+\delta)$-distributed (Definition 4.3), let μ be a $(1/k)$-uniform distribution over the assignments $V \to \{\mathsf{F}, \mathsf{T}\}$, and let ρ be an integer with $0 \leq \rho \leq |V|/2^k$. Consider the following experiment. First, draw $S \subseteq V$ from the uniform distribution τ over subsets of V with size ρ. Then, sample an assignment Λ from $\mu|_{V \setminus S}$. Denote by $\mathcal{F}$ the event that that there is a connected set of clauses Y of Φ with $|Y| \geq L$ such that all clauses in Y are unsatisfied by Λ. Then $\Pr_{S \sim \tau}\left(\Pr_{\Lambda \sim \mu|_{V \setminus S}}(\mathcal{F}) \leq 2^{-\delta k L}\right) \geq 1 - 2^{-\delta k L}$.

Proof. We apply Lemma 4.28 with our choices of b and with $\hat{\delta} = \delta$ to conclude that, w.h.p. over the choice of Φ, for every connected set of clauses $Z \subseteq \mathcal{C}$ we have

$$\text{if } \ a\log(n) \leq |Z| \leq b\log(n), \quad \text{then } \ |\mathrm{var}(Z) \cap V| \geq rk|Z|. \tag{4.14}$$

We also need the following result on random k-CNF formulae. For each clause $c \in \mathcal{C}$, let

$$\mathcal{Z}(c, L) = \{Z \subseteq \mathcal{C} : c \in Z, Z \text{ is connected in } G_\Phi, |Z| = L\}.$$

Then, w.h.p. over the choice of Φ, Lemma 4.29 shows that, as long as $L \geq \log n$,

$$\text{for any clause } c \in \mathcal{C} \text{ we have } \ |\mathcal{Z}(c, L)| \leq (9k^2\alpha)^L. \tag{4.15}$$

The facts that we have just established using Lemma 4.28 and Lemma 4.29 are all the properties of random formulae that we need in this proof. The hypothesis $\alpha \leq \Delta_r$ is used when calling Lemma 4.15 in the proof of Lemma 4.28.

Let L be an integer with $a \log n \leq L \leq b \log n$. First, we are going to fix $S \subseteq V$ with $|S| = \rho$ and study the event $\mathcal{F}$ described in the statement. For $c \in \mathcal{C}$ and $Z \in \mathcal{Z}(c, L)$, we denote by $\mathcal{E}_1(Z, S)$ the event that $Z \subseteq \mathcal{C}^\Lambda$, where Λ is drawn from $\mu|_{V \setminus S}$, see Definition 4.6. Recall that $Z \subseteq \mathcal{C}^\Lambda$ means that none of the clauses in Z are satisfied by the assignment Λ (Definition 4.4). We note that $\mathcal{F} = \bigcup_{c \in \mathcal{C}, Z \in \mathcal{Z}(c,L)} \mathcal{E}_1(Z, S)$. We are going to show that, for large enough n,

$$\Pr_{S \sim \tau}\left(\Pr_{\Lambda \sim \mu|_{V \setminus S}}\left(\bigcup_{c \in \mathcal{C}, Z \in \mathcal{Z}(c,L)} \mathcal{E}_1(Z, S)\right) > 2^{-\delta k L} \right) \leq 2^{-\delta k L}, \tag{4.16}$$

which is equivalent to the result stated in this lemma. Using the union bound

$$\Pr_{\Lambda \sim \mu|_{V \setminus S}}\left(\bigcup_{c \in \mathcal{C}, Z \in \mathcal{Z}(c,L)} \mathcal{E}_1(Z, S)\right) \leq \sum_{c \in \mathcal{C}} \Pr_{\Lambda \sim \mu|_{V \setminus S}}\left(\bigcup_{Z \in \mathcal{Z}(c,L)} \mathcal{E}_1(Z, S)\right),$$

we note that if $\Pr_{\Lambda \sim \mu|_{V \setminus S}}\left(\bigcup_{c \in \mathcal{C}, Z \in \mathcal{Z}(c,L)} \mathcal{E}_1(Z, S)\right) > 2^{-\delta k L}$, then there is a clause $c \in \mathcal{C}$ with $\Pr_{\Lambda \sim \mu|_{V \setminus S}}\left(\bigcup_{Z \in \mathcal{Z}(c,L)} \mathcal{E}_1(Z, S)\right) > 2^{-\delta k L}/|\mathcal{C}|$. Repeating the same argument, now with a union bound on $Z \in \mathcal{Z}(c, L)$, if there is $c \in \mathcal{C}$ with $\Pr_{\Lambda \sim \mu|_{V \setminus S}}\left(\bigcup_{Z \in \mathcal{Z}(c,L)} \mathcal{E}_1(Z, S)\right) > 2^{-\delta k L}/|\mathcal{C}|$,

then there is $Z \in \mathcal{Z}(c,L)$ such that $\Pr_{\Lambda\sim\mu|_{V\setminus S}}(\mathcal{E}_1(Z,S)) > 2^{-\delta kL}/(|\mathcal{C}|\cdot|\mathcal{Z}(c,L)|)$. We have shown that the event $\left[\Pr_{\Lambda\sim\mu|_{V\setminus S}}\left(\bigcup_{c\in\mathcal{C},Z\in\mathcal{Z}(c,L)}\mathcal{E}_1(Z,S)\right) > 2^{-\delta kL}\right]$ is contained in the event $\left[\exists c\in\mathcal{C}, Z\in\mathcal{Z}(c,L) : \Pr_{\Lambda\sim\mu|_{V\setminus S}}(\mathcal{E}_1(Z,S)) > 2^{-\delta kL}(|\mathcal{C}|\cdot|\mathcal{Z}(c,L)|)\right]$. Therefore, the left-hand side of (4.16) can be upper bounded by

$$\begin{aligned}\Pr_{S\sim\tau}\left(\exists c\in\mathcal{C}, Z\in\mathcal{Z}(c,L) : \Pr_{\Lambda\sim\mu|_{V\setminus S}}(\mathcal{E}_1(Z,S)) > \frac{2^{-\delta kL}}{|\mathcal{C}|\cdot|\mathcal{Z}(c,L)|}\right) \leq \\ \sum_{c\in\mathcal{C},Z\in\mathcal{Z}(c,L)} \Pr_{S\sim\tau}\left(\Pr_{\Lambda\sim\mu|_{V\setminus S}}(\mathcal{E}_1(Z,S)) > \frac{2^{-\delta kL}}{|\mathcal{C}|\cdot|\mathcal{Z}(c,L)|}\right).\end{aligned} \tag{4.17}$$

We are going to show that, for any $c\in\mathcal{C}$ and $Z\in\mathcal{Z}(c,L)$,

$$\Pr_{S\sim\tau}\left(\Pr_{\Lambda\sim\mu|_{V\setminus S}}(\mathcal{E}_1(Z,S)) > \frac{2^{-\delta kL}}{|\mathcal{C}|\cdot|\mathcal{Z}(c,L)|}\right) \leq \left(2ek\cdot 2^{-rk}\right)^L. \tag{4.18}$$

Before proving (4.18), let us complete the proof assuming that this inequality holds. In light of (4.15), we have $|\mathcal{Z}(c,L)| \leq (9k^2 2^{(r-2\delta)k})^L$. We use the following observation,

$$\text{for } k > 1/(\delta\log 2) \text{ and for large enough } n, \quad |\mathcal{C}| \leq n\alpha \leq n^{\delta k^5\log 2} \leq 2^{(\delta/2)kL}. \tag{4.19}$$

Combining (4.17), (4.18) and (4.19), we conclude that, for large enough k, the left-hand size of (4.16) is bounded above by

$$\sum_{c\in\mathcal{C},Z\in\mathcal{Z}(c,L)}\left(2ek\cdot 2^{-rk}\right)^L \leq n\alpha\cdot\left(9k^2 2^{(r-2\delta)k}\right)^L\cdot\left(2ek\cdot 2^{-rk}\right)^L = n\alpha\left(18ek^3 2^{-2\delta k}\right)^L \leq 2^{-\delta kL},$$

which completes the proof of (4.16), and hence the proof of the lemma, subject to (4.18).

To prove (4.18), we are going to find many S for which $\Pr_{\Lambda\sim\mu|_{V\setminus S}}(\mathcal{E}_1(Z,S)) \leq 2^{-\delta kL}/(|\mathcal{C}|\cdot|\mathcal{Z}(c,L)|)$ holds. With this in mind, we introduce an event that may occur when sampling S:

$$\begin{aligned}\mathcal{E}_2(Z) := &\text{“the random set } S\subseteq V \text{ that we select contains fewer} \\ &\text{than } \ell := \lceil|\mathrm{var}(Z)\cap V|/k\rceil \text{ variables in } \mathrm{var}(Z)\cap V\text{”}.\end{aligned} \tag{4.20}$$

We will show (in equation (4.24)) that the event $\mathcal{E}_2(Z)$ holds for most choices of S. Before proving this claim, let us assume that $\mathcal{E}_2(Z)$ holds for S and let us prove that $\Pr_{\Lambda\sim\mu|_{V\setminus S}}(\mathcal{E}_1(Z,S)) \leq 2^{-\delta kL}/(|\mathcal{C}|\cdot|\mathcal{Z}(c,L)|)$. If there are $c_1,c_2\in Z$ and $v\in\mathrm{var}(c_1)\cap\mathrm{var}(c_2)\cap(V\setminus S)$ such that $c_1\neq c_2$ and the literal of v in c_1 is the negation of the literal of v in c_2, then at least one of c_1 and c_2 is satisfied by the assignment $\Lambda\colon V\setminus S\to\{\mathsf{F},\mathsf{T}\}$. In this case we have $\Pr_{\Lambda\sim\mu|_{V\setminus S}}(\mathcal{E}_1(Z,S)) = 0$. Let us now consider the complementary case:

$$\begin{aligned}&\text{for all } c_1,c_2\in Z \text{ with } c_1\neq c_2 \text{ and } v\in\mathrm{var}(c_1)\cap\mathrm{var}(c_2)\cap(V\setminus S), \\ &\text{the literal of } v \text{ in } c_1 \text{ is the same as the literal of } v \text{ in } c_2.\end{aligned} \tag{4.21}$$

In this setting, we call $\omega(v)$ the value of v that does not satisfy the clauses in Z that contain v. Note that $\omega(v)$ is well-defined by assumption (4.21). Let $u_1,u_2,\ldots,u_t$ be the list of variables in

$(\mathrm{var}(Z) \cap V) \setminus S$. We denote by $\mathcal{W}_j$ the event that u_j is assigned the value $\omega(u_j)$ by Λ when sampling $\Lambda \sim \mu|_{V \setminus S}$. Then, by definition of $\mathcal{W}_j$, we have

$$\Pr_{\Lambda \sim \mu|_{V\setminus S}}(\mathcal{E}_1(Z,S)) = \prod_{j=1}^{t} \Pr_{\Lambda \sim \mu|_{V\setminus S}}\left(\mathcal{W}_j \,\middle|\, \bigcap_{i=1}^{j-1} \mathcal{W}_i\right).$$

As μ is $(1/k)$-uniform, we find that $\Pr_{\Lambda\sim\mu|_{V\setminus S}}(\mathcal{W}_j| \bigcap_{i=1}^{j-1}\mathcal{W}_i) \leq (1/2)\exp(1/k)$ for all $j \in \{1,2,\ldots,t\}$. We conclude that

$$\Pr_{\Lambda \sim \mu|_{V\setminus S}}(\mathcal{E}_1(Z,S)) \leq \left(\frac{1}{2}\exp\left(\frac{1}{k}\right)\right)^t.$$

From (4.14) and the fact that $\mathcal{E}_2(Z)$ holds for S, we have

$$t = |\mathrm{var}(Z)\cap(V\setminus S)| \geq |\mathrm{var}(Z)\cap V| - \lceil |\mathrm{var}(Z)\cap V|/k\rceil \geq |\mathrm{var}(Z)\cap V|(1-1/k)-1 \geq rL(k-1)-1.$$

It follows that

$$\begin{aligned}\Pr_{\Lambda \sim \mu|_{V\setminus S}}(\mathcal{E}_1(Z,S)) &\leq \left(\frac{1}{2}\exp\left(\frac{1}{k}\right)\right)^{r(k-1)L-1}\\ &\leq 2\left(2\cdot 2^{-rk}\exp\left(\frac{r(k-1)}{k}\right)\right)^{L}\\ &\leq \left(4e\cdot 2^{-rk}\right)^{L},\end{aligned}$$

where we used that $1/2 \leq (1/2)\exp(1/k) < 1$ in the second and third inequality. For large enough k, we find that

$$\left(4e\cdot 2^{-rk}\right)^L = \left(\frac{9\cdot 4ek^2\cdot\alpha\cdot 2^{-rk}}{9k^2\alpha}\right)^L \leq \left(\frac{9\cdot 4ek^2\cdot 2^{-2\delta k}}{9k^2\alpha}\right)^L \leq \frac{2^{-(3/2)\delta kL}}{|\mathcal{Z}(c,L)|} \leq \frac{2^{-\delta kL}}{|\mathcal{C}|\cdot\mathcal{Z}(c,L)|}, \tag{4.22}$$

where in the second to last inequality we applied $9\cdot 4ek^2 \leq 2^{(\delta/2)k}$ and the bound on the size of $\mathcal{Z}(c,L)$ given in (4.15), and in the last inequality we used (4.19). As S was picked as any subset of V with $|S| = \rho$ such that $\mathcal{E}_2(Z)$ holds, it follows that

$$\Pr_{S\sim\tau}\left(\Pr_{\Lambda\sim\mu|_{V\setminus S}}(\mathcal{E}_1(Z,S)) > \frac{2^{-\delta kL}}{|\mathcal{C}|\cdot|\mathcal{Z}(c,L)|}\right) \leq \Pr_{S\sim\tau}\left(\overline{\mathcal{E}_2(Z)}\right). \tag{4.23}$$

In order to prove (4.18), which finishes the proof, we need to show $\Pr_{S\sim\tau}\left(\overline{\mathcal{E}_2(Z)}\right) \leq (2ek\cdot 2^{-rk})^L$. The probability of $\overline{\mathcal{E}_2(Z)}$ can be bounded as follows. Recall that $|S| = \rho$. If $\rho < \ell$, then, by the definition of $\mathcal{E}_2(Z)$ in (4.20), we obtain $\Pr_{S\sim\tau}(\mathcal{E}_2(Z)) = 1$. Otherwise, the number of choices of S (with $|S| = \rho$) such that $|S\cap\mathrm{var}(Z)\cap V| \geq \ell$ is at most $\binom{|\mathrm{var}(Z)\cap V|}{\ell}\binom{|V|-\ell}{\rho-\ell}$. Hence, we have

$$\begin{aligned}\Pr_{S\sim\tau}\left(\overline{\mathcal{E}_2(Z)}\right) &\leq \binom{|V|}{\rho}^{-1}\binom{|\mathrm{var}(Z)\cap V|}{\ell}\binom{|V|-\ell}{\rho-\ell}\\ &= \frac{\rho(\rho-1)\cdots(\rho-\ell+1)}{|V|(|V|-1)\cdots(|V|-\ell+1)}\binom{|\mathrm{var}(Z)\cap V|}{\ell}\\ &\leq \left(\frac{\rho}{|V|}\right)^{\ell}\left(\frac{e|\mathrm{var}(Z)\cap V|}{\ell}\right)^{\ell} \leq \left(\frac{\rho}{|V|}ek\right)^{\ell},\end{aligned}$$

where we used $\ell := \lceil |\mathrm{var}(Z) \cap V|/k \rceil \geq |\mathrm{var}(Z) \cap V|/k$, $(p-i)/(q-i) \leq p/q$ for any $0 < i < p < q$ and $\binom{p}{q} \leq (ep/q)^q$. Combining this with the hypothesis $\rho \leq |V|/2^k$ and the bound $\ell \geq rL$, see (4.14), we obtain

$$\Pr_{S \sim \tau}\left(\overline{\mathcal{E}_2(Z)}\right) \leq \left(ek2^{-k}\right)^{\ell} \leq \left((ek)^r \cdot 2^{-rk}\right)^L \leq \left(2ek \cdot 2^{-rk}\right)^L. \tag{4.24}$$

The bound (4.18) follows from combining (4.23) and (4.24), which completes the proof. □

4.5 Sampling from small connected components

In this section we prove Lemma 4.14. Recall that Lemma 4.14 claims the existence of a procedure to sample from marginals of the uniform distribution on the satisfying assignments of Φ^Λ when the connected components of G_{Φ^Λ} have small size. Here we make this procedure explicit. Our algorithm exploits the fact that the tree-excess of logarithmic-sized subsets of G_Φ is bounded by a constant depending only on k, see Lemma 4.26, and the fact that when G_Φ is acyclic, we can exactly count and sample satisfying assignments efficiently via a dynamic programming algorithm (Proposition 4.30).

Proposition 4.30. *There is an algorithm that, for any k-CNF formula $\Phi = (\mathcal{V}, \mathcal{C})$ such that G_Φ is a tree, computes the number of satisfying assignments of Φ in time $O(4^k|\mathcal{C}|)$.*

Proof. We give an algorithm based on dynamic programming. Let us fix a vertex / clause c of G_Φ as the root and consider the corresponding directed tree structure $T := (G_\Phi, c)$. For any clause c' of Φ, let $T_{c'}$ be the subtree of T hanging from c'. For any assignment $\sigma\colon \mathrm{var}(c') \to \{\mathsf{F}, \mathsf{T}\}$, let $\mathrm{sa}(c', \sigma)$ denote the number of satisfying assignments of the formula determined by $T_{c'}$ that extend σ. Our goal is computing the number of satisfying assignments of Φ, which, under this notation, is equal to

$$\mathrm{sa}(\Phi) := \sum_{\sigma\colon\, \mathrm{var}(c) \to \{\mathsf{F},\mathsf{T}\}} \mathrm{sa}(c, \sigma). \tag{4.25}$$

We do this by computing $\mathrm{sa}(c', \sigma)$ for any clause c' and any assignment $\sigma\colon \mathrm{var}(c') \to \{\mathsf{F}, \mathsf{T}\}$. Using the tree structure of T, we show that $\mathrm{sa}(c', \sigma)$ satisfies a recurrence. There are two cases:

1. c' is a leaf. Then $\mathrm{sa}(c', \sigma) = 1$ if c' is satisfied by σ and 0 otherwise.

2. c' is not a leaf. Let $T_1, \ldots, T_l$ be the trees hanging from c' in T and let $c_1, \ldots, c_l$ be their roots. Then, since $T_1, \ldots, T_l$ do not share variables as G_Φ is acyclic, we have

$$\mathrm{sa}(c', \sigma) = \prod_{j=1}^{l} \sum_{\tau \in A(c_j, \sigma)} \mathrm{sa}(c_j, \tau),$$

where $A(c_j, \sigma)$ is the set of assignments of the variables in $\mathrm{var}(c_j)$ that agree with σ on $\mathrm{var}(c') \cap \mathrm{var}(c_j)$.

We can apply this recurrence with dynamic programming to compute $\mathrm{sa}(c, \sigma)$ for any assignment $\sigma\colon \mathrm{var}(c) \to \{\mathsf{F}, \mathsf{T}\}$. More explicitly, we compute $\mathrm{sa}(c', \sigma)$ by levels of the tree, starting from the deepest level, where all nodes are leaves, and ending at the root c. This involves computing at most 2^k entries $\mathrm{sa}(c', \cdot)$ per clause c' of Φ. After computing all the entries appearing in this recurrence, we compute the number of satisfying assignments of Φ, $\mathrm{sa}(\Phi)$, as in equation (4.25). The overall procedure takes at most $O(4^k|\mathcal{C}|)$ steps since each entry $\mathrm{sa}(c', \sigma)$ is accessed at most 2^k times when computing the corresponding entries for the parent of c', and there are at most $2^k|\mathcal{C}(T)|$ entries. □

In Algorithm 3 we give an algorithm based on Proposition 4.30 to count satisfying assignments of a k-CNF formula. Recall the folklore fact that if we can count satisfying assignments then we can sample from the marginal of μ_Ω on v by counting the satisfying assignments of $\Phi^{v\mapsto\mathsf{F}}$ and $\Phi^{v\mapsto\mathsf{T}}$.

Algorithm 3 Counting satisfying assignments via trees

Input: a k-CNF formula $\Phi = (\mathcal{V}, \mathcal{C})$

Output: The number of satisfying assignments of Φ.

1: Find a spanning forest T of G_Φ.
2: Let $\mathcal{V}_T$ be the set of variables that gives rise to edges of G_Φ that are not in T.
3: *count* $\leftarrow 0$.
4: **for all** $\Lambda\colon \mathcal{V}_T \to \{\mathsf{F}, \mathsf{T}\}$ **do**
5: Note that the graph G_{Φ^Λ} is acyclic. Hence, we can count the number of satisfying assignments of Φ^Λ in time $O(4^k|\mathcal{C}(\Phi^\Lambda)|)$ by applying Proposition 4.30 to each connected component of G_{Φ^Λ} and taking the product of the numbers obtained. Let $\mathrm{sa}(\Phi^\Lambda)$ be the result of this computation.
6: *count* $\leftarrow$ *count* $+ \mathrm{sa}(\Phi^\Lambda)$.
7: **end for**
8: **return** *count*

Proposition 4.31. *Let $\Phi = (\mathcal{V}, \mathcal{C})$ be a k-CNF formula and let c be the tree-excess of G_Φ. Then Algorithm 3 counts the number of satisfying assignments of Φ in time $O(2^{k(c+2)}|\mathcal{C}|)$.*

Proof. We note that, in the execution of Algorithm 3, we have $|\mathcal{V}_T| \leq kc$. Hence, there are at most 2^{kc} iterations of the for loop and each one takes $O(4^k|\mathcal{C}|)$ steps, so the running time follows. The fact that the algorithm is correct follows from the correctness of the procedure presented in Proposition 4.30. □

Even though the running time of Algorithm 3 is not polynomial in the size of the formula Φ (in fact, it is exponential in general), we obtain linear running time when the formulae considered have constant tree-excess. As shown in Lemma 4.26, this is the case for logarithmic-sized subsets of clauses of random formulae. We can now finish the proof of Lemma 4.14.

Lemma 4.14. *There is an integer $k_0 \geq 3$ such that, for any integers $k \geq k_0$, $b \geq 2k^4$ and any density $\alpha > 0$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. Let V be a subset of variables and let $\Lambda \colon V \to \{\mathsf{F}, \mathsf{T}\}$ be a partial assignment such that all the connected components in G_{Φ^Λ} have size at most $b \log(n)$. Then, there is an algorithm that, for any $S \subseteq \mathcal{V} \setminus V$, samples an assignment from $\mu_{\Omega^\Lambda}|_S$ in time $O(|S| \log n)$.*

Proof. We apply Lemma 4.26, so, w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$, any connected set of clauses in G_Φ with size at most $b \log(n)$ has tree-excess at most $c = \max\{1, 2b \log(e\alpha k^2)\} = O(1)$. First, we give an algorithm for the case $|S| = 1$. Let Φ, V and Λ as in the statement, and let $S = \{v\}$. Let H be the connected component of the clauses that contain v in G_{Φ^Λ}, and let $\Phi' = (\mathcal{V}', \mathcal{C}')$ be the subformula of Φ^Λ with $G_{\Phi'} = H$. The formula Φ' has size at most $b \log(n)$. Moreover, the graph $G_{\Phi'} = H$ has tree-excess at most c as H is a subgraph of G_Φ with size at most $b \log(n)$. Thus, we can apply Proposition 4.31 to count the number of satisfying assignments of $\Phi'^{v \to \mathsf{F}}$ and $\Phi'^{v \to \mathsf{T}}$ in time $O(2^{k(c+2)}|\mathcal{C}'|) = O(\log n)$. Let these numbers be t_0 and t_1 respectively. It is straightforward to use t_0 and t_1 to sample from the marginal of the distribution μ_{Ω^Λ} for v; we only have to sample an integer $t \in [0, t_0 + t_1)$ and output F if $t < t_0$ and T otherwise. The whole process takes time $O(\log n)$.

Finally, we argue how to extend this algorithm to the case $|S| > 1$. For this, first, we give an order to the variables in S, say $u_1, u_2, \ldots, u_\ell$. We then call the algorithm described in the paragraph above once for each variable in $u_1, u_2, \ldots, u_\ell$. The inputs of the algorithm in the j-th call are the variable u_j and the assignment $\Lambda_j = \Lambda \cup \tau_{j-1}$, where τ_{j-1} is the assignment obtained in the previous calls for $u_1, \ldots, u_{j-1}$. After this process, τ_ℓ is an assignment of all the variables in S that follows the distribution $\mu_{\Omega^\Lambda}|_S$. This assignment has been computed in $O(|S| \log n)$ steps as we wanted. □

4.6 Mixing time of the Markov chain

In this section we study the mixing time of the ρ-uniform-block Glauber dynamics on the marked variables and prove Lemma 4.10. As explained in Section 4.1.2, in order to conclude rapid mixing of this Markov chain we apply the spectral independence framework, which has recently been extended to the ρ-uniform-block Glauber dynamics [29]. Traditionally in path coupling or spectral independence arguments one has to bound a sum of influences by a constant in order to obtain rapid mixing of the single-site Glauber dynamics. However, due to the presence of high-degree variables, an $O(1)$ upper bound seems unattainable in the random k-SAT formula setting; in the worst case paths of high-degree variable may significantly affect influences. This seems also to be the case for other random models, such as the hardcore model on random graphs [17]. Here we show that that sums of influences are at most $\varepsilon \log n$ for small ε (Lemma 4.9). Even though this is generally not enough to conclude rapid mixing of the single-site Glauber dynamics, it turns out to be enough to conclude rapid mixing of the ρ-uniform-block Glauber dynamics for $\rho = \Theta(n)$. An essential ingredient in our argument is exploiting the auxiliary

variables in introduced in Section 4.3. Therefore, in this section we will work with $r = r_0 - \delta$ and a $(r, r_0, r_0, 2r_0)$-marking $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{a}}, \mathcal{V}_{\mathrm{c}})$. Since r is fixed, we drop it from the notation and write, for instance, $\mathcal{V}_{\mathrm{good}}$ instead of $\mathcal{V}_{\mathrm{good}}(r)$, in order to simplify the reading of this section.

This section is divided as follows. In Section 4.6.1, we explain why bounded-degree methods to bound the mixing time of the Glauber dynamics fail to generalise from the bounded-degree k-SAT model to the random k-SAT model. In Section 4.6.2 we prove Lemmas 4.40 and 4.9. In Section 4.6.3 we prove Lemma 4.10.

4.6.1 Previous work

In this section we explain why previously known arguments for showing rapid mixing of the Glauber dynamics on bounded-degree k-SAT formulae do not extend to the random k-SAT model. This section is not used in our work and may be skipped by a reader who just wants to understand our approach and result. The best result currently known on bounded-degree formulae is [74], where the authors show, for large enough k, how to efficiently sample satisfying assignments of k-CNF formulae in which their variables have maximum degree $\hat{\Delta} \leq C\, 2^{0.1742 \cdot k}/k^3$, where $C > 0$ is a constant that does not depend on k.[1] Their result actually holds in the more general setting of atomic constrain satisfaction problems (albeit with a different bound on $\hat{\Delta}$). As part of their work, they show that the single-site Glauber dynamics on a set of marked variables mixes quickly. Their argument is restricted to atomic CSPs with bounded-degree and strongly exploits the properties of the Glauber dynamics in this setting. They study the optimal coupling of the single-site Glauber dynamics, we refer to [92] for the definition of coupling of Markov chains. In such a coupling the goal is to show that two copies of the chain starting from truth assignments differing in at least a marked variable (a so-called discrepancy) can be coupled in a small number of steps. Here it is crucial that the marginals of the marked variables are near $1/2$, so the optimal coupling generates new discrepancies with small probability. At this stage, the high-level idea to conclude rapid mixing of the Glauber dynamics is bounding the probability that the dynamics has not coupled by a product of probabilities, each corresponding to the event that a clause is unsatisfied at a certain time, and aggregating over all possible discrepancy sequences.

The fundamental observation in [74], based on the work on monotone k-CNF formulae presented in [70], is that if there is an update of a marked variable that generates a discrepancy in the chains, then there is another marked variable where the chains disagree that is connected to the former variable through a path of clauses, where consecutive clauses in the path share at least a variable. Moreover, each one of the clauses in this path is unsatisfied by at least one of the two copies of the chain. As a consequence, from a discrepancy at time t one can find a sequence of discrepancies going back to time 0, and these discrepancies are joined by a path of clauses. Thus, the union bound over discrepancy sequences is essentially a union bound over

[1] In [74] the maximum degree $\hat{\Delta}$ of Φ is defined as the maximum over $c \in \mathcal{C}$ of the number of clauses that share a variable with c. Under this definition of $\hat{\Delta}$, their result holds for $\hat{\Delta} \leq C 2^{0.1742 \cdot k}/k^2$.

paths of clauses with a particular time structure, where the same clause can be appear in the path several times. Extending this idea to the random k-SAT model presents two main issues. First of all, the number of discrepancy sequences of any given length may be too large due to the presence of bad clauses and the fact that they can repeatedly appear in the sequence. Moreover, it may be the case that these discrepancy sequences mostly consist of bad clauses, which are always unsatisfied in both chains and, thus, the probability that they are unsatisfied is not small. Interestingly, similar issues arise when directly extending the bounded-degree approach based on the coupling process of [93, 43] to our setting. In [43] the mixing time argument only succeeds when $\hat{\Delta} \leq 2^{k/20}/(8k)$ and is also based on a union bound over path of clauses that are unsatisfied or contain discrepancies after running a coupling process. However, very importantly, these paths of clauses are simple (clauses are not repeated) and the combinatorial structures appearing in the coupling process are less complex than the discrepancy sequences of [74]. This allowed the authors of [49] to exploit the expansion properties of random k-CNF formulae to analyse the coupling process of [93] on the random setting. Here we incorporate novel ideas to the work of [49] in order to obtain a tighter analysis that leads to nearly linear running time of our sampling algorithm.

4.6.2 Spectral independence in the k-SAT model

In this section we prove Lemma 4.9. In order to bound the sum of influences of marked variables, we follow the coupling process technique that is standard in the literature [49, 93, 43]. In this work we introduce the concept of auxiliary variables in the coupling process and exploit the sparsity properties of logarithmic-sized sets of clauses, which allows us to conclude a $2^{-r_0 k} \log n$ spectral independence bound. The key idea is that if we progressively extend two assignments X and Y on auxiliary variables following the optimal coupling, with high probability over X and Y, at some point the formulae Φ^X and Φ^Y factorise in small connected components in spite of the presence of bad variables and, on top of that, Φ^X and Φ^Y share most of these connected components. Then we can bound influences between marked variables by analysing the connected components where Φ^X and Φ^Y differ. First, let us introduce the notation and results on couplings that we need.

Let μ and ν be two distributions over the same space $\widehat{\Omega}$. A coupling τ of μ and ν is a joint distribution over $\widehat{\Omega} \times \widehat{\Omega}$ such that the projection of τ on the first coordinate is μ and the projection on the second coordinate is ν. Recall that the total variation distance of μ and ν is defined by $d_{\mathrm{TV}}(\mu, \nu) = \frac{1}{2} \sum_{x \in \widehat{\Omega}} |\mu(x) - \nu(x)|$. If a random variable X has distribution μ, we also write $d_{\mathrm{TV}}(X, \nu)$ to mean $d_{\mathrm{TV}}(\mu, \nu)$. An important property of couplings is the coupling lemma.

Proposition 4.32 (Coupling lemma)**.** *Let τ be a coupling of μ and ν. Then $d_{\mathrm{TV}}(\mu, \nu) \leq \Pr_{(X,Y)\sim\tau}(X \neq Y)$. Moreover, there exists a coupling that achieves equality.*

The coupling τ of μ and ν that minimises $\Pr_{(X,Y)\sim\tau}(X \neq Y)$ is called *optimal*. Let us now assume that μ and ν are Bernoulli distributions with parameters $0 \leq p \leq q \leq 1$ respectively, so $\Pr_\mu(X = 1) = p$ and $\Pr_\nu(Y = 1) = q$. The *monotone coupling* τ of μ and ν is defined as

follows. We pick U uniformly at random in $[0,1]$ and set $X=1$ only when $U\le p$ and $Y=1$ only when $U\le q$. For this coupling we have $\Pr_{(X,Y)\sim\tau}(X\neq Y)=q-p=d_{\mathrm{TV}}(X,Y)$ and, hence, the monotone coupling is optimal. This optimal coupling will come up in the coupling process when sampling from the marginals of auxiliary variables.

Before presenting our coupling process, we show how we can bound a sum of influences between marked variables with the help of the coupling lemma. In all this section we fix a k-CNF formula Φ and a $(r,r_0,r_0,2r_0)$-marking $(\mathcal{V}_{\mathrm{m}},\mathcal{V}_{\mathrm{a}},\mathcal{V}_{\mathrm{c}})$ of Φ. Given two assignments Λ_1 and Λ_2 on disjoint sets of variables, recall that we denote by $\Lambda_1\cup\Lambda_2$ the combined assignment on the union of their domains.

Proposition 4.33. *Let $u\in\mathcal{V}_{\mathrm{m}}$ and $\Lambda\colon S\to\{\mathsf{F},\mathsf{T}\}$ with $S\subseteq\mathcal{V}_{\mathrm{m}}\setminus\{u\}$. Let (X,Y) be a coupling where X follows the distribution $\mu_{\Omega^{\Lambda\cup u\mapsto\mathsf{T}}}|_{\mathcal{V}_{\mathrm{m}}}$ and Y follows the distribution $\mu_{\Omega^{\Lambda\cup u\mapsto\mathsf{F}}}|_{\mathcal{V}_{\mathrm{m}}}$. Then*

$$\sum_{v\in\mathcal{V}_{\mathrm{m}}\setminus(S\cup\{u\})}\left|\mathcal{I}^{\Lambda}(u\to v)\right|\le\sum_{v\in\mathcal{V}_{\mathrm{m}}\setminus(S\cup\{u\})}\Pr\left(X(v)\neq Y(v)\right). \tag{4.26}$$

Proof. Let $v\in\mathcal{V}_{\mathrm{m}}$. Then for any $\omega\in\{\mathsf{F},\mathsf{T}\}$, we have $\Pr(v\mapsto\omega|\Lambda,u\mapsto\mathsf{T})=\Pr(X(v)=\omega)$ and $\Pr(v\mapsto\omega|\Lambda,u\mapsto\mathsf{F})=\Pr(Y(v)=\omega)$. Thus, by the coupling lemma,

$$\left|\mathcal{I}^{\Lambda}(u\to v)\right|=\left|\Pr(X(v)=\mathsf{T})-\Pr(Y(v)=\mathsf{T})\right|=d_{\mathrm{TV}}(X(v),Y(v))\le\Pr\left(X(v)\neq Y(v)\right),$$

and the proof follows by adding over $v\in\mathcal{V}_{\mathrm{m}}\setminus(S\cup\{u\})$. □

For two assignments X and Y on a subset of variables V, we say that X and Y have a *discrepancy* at $v\in V$ when $X(v)\neq Y(v)$. In [43] the authors manage to bound (4.26) by a constant that does not depend on n when the considered formula has bounded degree. However, their argument breaks under the presence of high-degree variables due to the fact that we cannot control the number of bad clauses in a path of clauses unless the path has length at least $\Omega(\log n)$. Here instead we perform the coupling process developed in [49] over auxiliary variables, which accounts for the presence of bad clauses.

Before presenting our algorithm for the coupling process on auxiliary variables, let us describe some of the notation and structures that are used in this algorithm. Let $u\in\mathcal{V}_{\mathrm{m}}$ and $\Lambda\colon S\to\{\mathsf{F},\mathsf{T}\}$ with $S\subseteq\mathcal{V}_{\mathrm{m}}\setminus\{u\}$. We start with two assignments $\widehat{X}$ and $\widehat{Y}$ that have a discrepancy at u and agree with Λ on S. In the coupling process we identify a set of failed clauses, denoted $\mathcal{F}_{\mathrm{d}}\cup\mathcal{F}_{\mathrm{u}}$. At each step of the process, we check if a clause is failed or extend the coupling to an auxiliary variable. It is important in our arguments that all clauses containing a discrepancy are failed, and that we make sure that the set of failed clauses is connected in G_Φ at all times. In order to achieve connectivity of failed clauses, at each step of the coupling process we only consider clauses that are adjacent to failed clauses in G_Φ. For ease of reading, here we present a list of the structures that appear in our algorithm.

1. $\mathcal{V}_{\mathrm{d}}$. Set of discrepancies, i.e., variables v with $\widehat{X}(v)\neq\widehat{Y}(v)$.
2. $\mathcal{F}_{\mathrm{d}}$. Set of all clauses containing a variable in $\mathcal{V}_{\mathrm{d}}$. These are failed clauses.

3. $\mathcal{V}_{\text{set}}$. Set of variables that are assigned a value in the coupling.

4. $\mathcal{F}_{\text{u}}$. Set of clauses that have been considered by the coupling process, and are either bad, or are unsatisfied by at least one of $\widehat{X}$ and $\widehat{Y}$ and have all their auxiliary variables in $\mathcal{V}_{\text{set}}$. These are failed clauses.

5. $\mathcal{C}_{\text{rem}}$. Set of clauses that have unassigned auxiliary variables or have not been explored yet.

Our coupling process on auxiliary variables is given in Algorithm 4.

Algorithm 4 The coupling process on auxiliary variables

Input: A k-CNF formula $\Phi = (\mathcal{V}, \mathcal{C})$, an $(r, r_0, r_0, 2r_0)$-marking $\mathcal{M} = (\mathcal{V}_{\text{m}}, \mathcal{V}_{\text{a}}, \mathcal{V}_{\text{c}})$, $u \in \mathcal{V}_{\text{m}}$ and $\Lambda \colon S \to \{\mathsf{F}, \mathsf{T}\}$ with $S \subseteq \mathcal{V}_{\text{m}} \setminus \{u\}$.

Output: a pair of assignments $\widehat{X}, \widehat{Y} \colon \mathcal{V}_{\text{set}} \to \{\mathsf{F}, \mathsf{T}\}$ for some set of variables $\mathcal{V}_{\text{set}}$ such that:
- $S \cup \{u\} \subseteq \mathcal{V}_{\text{set}} \subseteq S \cup \{u\} \cup \mathcal{V}_{\text{a}}$,
- $\widehat{X}$ and $\widehat{Y}$ agree with Λ on S, $\widehat{X}(u) = \mathsf{T}$ and $\widehat{Y}(u) = \mathsf{F}$.

We fix two total orders $\leq_{\mathcal{V}}$ and $\leq_{\mathcal{C}}$ over the variables and clauses of Φ. These are only relevant to have a pre-determined order in which clauses and variables are considered in this algorithm.
Initialise $\widehat{X}$ and $\widehat{Y}$ as Λ, and set $\widehat{X}(u) = \mathsf{T}$ and $\widehat{Y}(u) = \mathsf{F}$.
$\mathcal{V}_{\text{set}} \leftarrow S \cup \{u\}$, $\mathcal{V}_{\text{d}} \leftarrow \{u\}$, $\mathcal{F}_{\text{d}} \leftarrow \{c \in \mathcal{C} : u \in \text{var}(c)\}$, $\mathcal{F}_{\text{u}} \leftarrow \emptyset$, $\mathcal{C}_{\text{rem}} \leftarrow \mathcal{C}$.
while $\exists c \in \mathcal{C}_{\text{rem}} : \text{var}(c) \cap (\mathcal{V}_{\text{d}} \cup \text{var}(\mathcal{F}_{\text{u}})) \neq \emptyset$ **do**
  Let c be smallest clause according to $\leq_{\mathcal{C}}$ with $\text{var}(c) \cap (\mathcal{V}_{\text{d}} \cup \text{var}(\mathcal{F}_{\text{u}})) \neq \emptyset$.
  if c is a bad clause **then**
    Remove c from $\mathcal{C}_{\text{rem}}$ and add c to $\mathcal{F}_{\text{u}}$.
  end if
  if c is a good clause and $(\text{var}(c) \cap \mathcal{V}_{\text{a}}) \setminus \mathcal{V}_{\text{set}} = \emptyset$ **then**
    Remove c from $\mathcal{C}_{\text{rem}}$ (as all auxiliary variables in c have been set).
    if c is unsatisfied by at least one of $\widehat{X}$ and $\widehat{Y}$ **then**
      Add c to $\mathcal{F}_{\text{u}}$.
    end if
  end if
  if c is a good clause and $(\text{var}(c) \cap \mathcal{V}_{\text{a}}) \setminus \mathcal{V}_{\text{set}} \neq \emptyset$ **then**
    Let v be the smallest variable in $(\text{var}(c) \cap \mathcal{V}_{\text{a}}) \setminus \mathcal{V}_{\text{set}}$ (according to $\leq_{\mathcal{V}}$).
    Extend $\widehat{X}$ and $\widehat{Y}$ to v by sampling from the optimal coupling between the marginal distributions of $\mu_{\Omega^{\widehat{X}}}$ and $\mu_{\Omega^{\widehat{Y}}}$ on v, and add v to $\mathcal{V}_{\text{set}}$.
    if $\widehat{X}(v) \neq \widehat{Y}(v)$ **then**
      Add v to $\mathcal{V}_{\text{d}}$. Add all clauses containing v to $\mathcal{F}_{\text{d}}$.
    end if
  end if
end while
return $(\widehat{X}, \widehat{Y})$.

First, we analyse the sets $\mathcal{V}_{\text{set}}$, $\mathcal{V}_{\text{d}}$, $\mathcal{F}_{\text{d}}$, $\mathcal{F}_{\text{u}}$ and $\mathcal{C}_{\text{rem}}$ and prove the connectivity property of $\mathcal{F}_{\text{d}} \cup \mathcal{F}_{\text{u}}$. In the rest of this section we fix the inputs of Algorithm 4 unless stated otherwise.

Proposition 4.34 (Properties of the coupling process)**.** *The coupling process in Algorithm 4 terminates eventually and, at the end of the process, the sets $\mathcal{V}_{\text{set}}$, $\mathcal{V}_{\text{d}}$, $\mathcal{F}_{\text{d}}$, $\mathcal{F}_{\text{u}}$ and $\mathcal{C}_{\text{rem}}$ present the following properties:*

1. *We have $S \cup \{u\} \subseteq \mathcal{V}_{\text{set}} \subseteq \mathcal{V}_{\text{a}} \cup S \cup \{u\}$, $\mathcal{V}_{\text{d}} = \{v \in \mathcal{V}_{\text{set}} : \widehat{X}(v) \neq \widehat{Y}(v)\}$, and $\mathcal{F}_{\text{d}}$ is the set of clauses containing a variable in $\mathcal{V}_{\text{d}}$.*
2. *For all $c \in \mathcal{F}_{\text{u}}$ we have $\text{var}(c) \cap \mathcal{V}_{\text{a}} \subseteq \mathcal{V}_{\text{set}}$ and c is unsatisfied by at least one of $\widehat{X}$ and $\widehat{Y}$.*
3. *For all $c \in \mathcal{C}_{\text{rem}}$, we have $\text{var}(c) \cap (\mathcal{V}_{\text{d}} \cup \text{var}(\mathcal{F}_{\text{u}})) = \emptyset$.*
4. *For all $c \in \mathcal{C} \setminus (\mathcal{C}_{\text{rem}} \cup \mathcal{F}_{\text{u}})$, we have $\text{var}(c) \cap (\mathcal{V}_{\text{d}} \cup \text{var}(\mathcal{F}_{\text{u}})) \neq \emptyset$, $\text{var}(c) \cap \mathcal{V}_{\text{a}} \subseteq \mathcal{V}_{\text{set}}$ and c is satisfied by $\widehat{X}$ and $\widehat{Y}$.*
5. *The set $\mathcal{F}_{\text{d}} \cup \mathcal{F}_{\text{u}}$ is connected in G_Φ.*

Proof. Each iteration of the coupling procedure either removes a clause from $\mathcal{C}_{\text{rem}}$, or samples the values $\widehat{X}(v)$ and $\widehat{Y}(v)$ for an auxiliary variable v and adds v to $\mathcal{V}_{\text{set}} \subseteq \mathcal{V}$. As $\mathcal{C}_{\text{rem}}$ and $\mathcal{V}$ are finite, the coupling terminates after a finite number of iterations. We prove the five properties in the statement separately. First, we note that the sets $\mathcal{V}_{\text{set}}$, $\mathcal{V}_{\text{d}}$, $\mathcal{F}_{\text{d}}$, $\mathcal{F}_{\text{u}}$ never decrease in size during the execution of Algorithm 4, whereas the set $\mathcal{C}_{\text{rem}}$ never increases in size.

Property 1. Note that at the start of Algorithm 4 (line 3) this property holds. The result then follows from the fact that the sets $\mathcal{V}_{\text{set}}$, $\mathcal{V}_{\text{d}}$ and $\mathcal{F}_{\text{d}}$ are only updated from line 15 to line 20 of Algorithm 4, and these steps preserve Property 1.

Property 2. This follows from the facts that the set $\mathcal{F}_{\text{u}}$ is originally empty, it is only extended in lines 7 and 12, and bad clauses do not contain auxiliary variables.

Property 3. This property follows from the fact that clauses that satisfy $\text{var}(c) \cap (\mathcal{V}_{\text{d}} \cup \text{var}(\mathcal{F}_{\text{u}})) \neq \emptyset$ at some point are eventually removed from $\mathcal{C}_{\text{rem}}$ in either line 7 (if they are bad) or in line 10 (if they are good, once all the auxiliary variables of the clause are in $\mathcal{V}_{\text{set}}$).

Property 4. If $c \in \mathcal{C} \setminus (\mathcal{C}_{\text{rem}} \cup \mathcal{F}_{\text{u}})$, then c has been removed from $\mathcal{C}_{\text{rem}}$ in line 10 but it has not been added to $\mathcal{F}_{\text{u}}$ in line 12, which proves this property.

Property 5. We note that at the start of the coupling process (line 3) $\mathcal{F}_{\text{d}} \cup \mathcal{F}_{\text{u}}$ is connected. Let us analyse every line of the algorithm where the sets $\mathcal{F}_{\text{d}}$ and $\mathcal{F}_{\text{u}}$ are enlarged. When it comes to $\mathcal{F}_{\text{d}}$, this occurs in line 19 if this line is executed. Let c be the clause considered in that iteration of the coupling process and let v be the variable of c considered in line 16. We recall that $\text{var}(c) \cap (\mathcal{V}_{\text{d}} \cup \text{var}(\mathcal{F}_{\text{u}})) \neq \emptyset$ and $v \in (\text{var}(c) \cap \mathcal{V}_{\text{a}}) \setminus \mathcal{V}_{\text{set}}$. In line 19 we add all to $\mathcal{F}_{\text{d}}$ all the clauses containing v. Let C_v be the set of such clauses. Since $\emptyset \neq \text{var}(c) \cap (\mathcal{V}_{\text{d}} \cup \text{var}(\mathcal{F}_{\text{u}})) \subseteq \text{var}(c) \cap \text{var}(\mathcal{F}_{\text{d}} \cup \mathcal{F}_{\text{u}})$ and $c \in C_v$, we conclude that $\mathcal{F}_{\text{d}} \cup \mathcal{F}_{\text{u}} \cup C_v$ is connected as we wanted. When it comes to $\mathcal{F}_{\text{u}}$, we add clauses in lines 7 and 12. In this case, we add a clause c such that $\text{var}(c) \cap (\mathcal{V}_{\text{d}} \cup \text{var}(\mathcal{F}_{\text{u}})) \neq \emptyset$, so $\mathcal{F}_{\text{d}} \cup \mathcal{F}_{\text{u}} \cup \{c\}$ is connected in G_Φ. □

We can now prove our main result concerning the structure of $\Phi^{\widehat{X}}$ and $\Phi^{\widehat{Y}}$.

Lemma 4.35. *Let $\widehat{X}$ and $\widehat{Y}$ be the assignments returned by Algorithm 4 and let $\mathcal{C}_{\text{rem}}$ and $\mathcal{F}_{\text{u}}$ be as in Proposition 4.34. There are sets of clauses $\mathcal{C}_1 \subseteq \mathcal{C}_{\text{rem}}$ and $\mathcal{C}_2, \mathcal{C}_3 \subseteq \mathcal{F}_{\text{u}}$ such that $\Phi^{\widehat{X}} = (\mathcal{V} \setminus \mathcal{V}_{\text{set}}, \mathcal{C}_1 \cup \mathcal{C}_2)$ and $\Phi^{\widehat{Y}} = (\mathcal{V} \setminus \mathcal{V}_{\text{set}}, \mathcal{C}_1 \cup \mathcal{C}_3)$, where the variables in $\mathcal{V}_{\text{set}}$ are removed from the clauses in $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$.*

Proof. We determine the set of clauses that are unsatisfied by $\widehat{X}$ or $\widehat{Y}$ with the help of Proposition 4.34. We distinguish 3 disjoint cases:

- $c \in \mathcal{C}_{\text{rem}}$. Then $\text{var}(c) \cap \mathcal{V}_{\text{d}} = \emptyset$, so $\widehat{X}$ and $\widehat{Y}$ agree in all the variables in $\text{var}(\mathcal{C}_{\text{rem}}) \cap \mathcal{V}_{\text{set}}$. As a consequence, the restrictions of $\Phi^{\widehat{X}}$ and $\Phi^{\widehat{Y}}$ to $\mathcal{C}_{\text{rem}}$ give rise to the same CNF formula. Note that some of the clauses in $\mathcal{C}_{\text{rem}}$ might be satisfied by both $\widehat{X}$ and $\widehat{Y}$, but they are never satisfied by only one of the two assignments.
- $c \in \mathcal{F}_{\text{u}}$. Then c is unsatisfied by at least one of $\widehat{X}$ and $\widehat{Y}$ and, thus, it appears in at least one of $\Phi^{\widehat{X}}$ and $\Phi^{\widehat{Y}}$. The clause c may contain a variable $v \in \mathcal{V}_{\text{d}}$.
- $c \in \mathcal{C} \setminus (\mathcal{C}_{\text{rem}} \cup \mathcal{F}_{\text{u}})$. By Proposition 4.34, we have $\text{var}(c) \cap (\mathcal{V}_{\text{d}} \cup \text{var}(\mathcal{F}_{\text{u}})) \neq \emptyset$ and $\text{var}(c) \cap \mathcal{V}_{\text{a}} \subseteq \mathcal{V}_{\text{set}}$. Since $c \notin \mathcal{F}_{\text{u}}$, it follows that c is satisfied by both $\widehat{X}$ and $\widehat{Y}$ and, thus, c does not appear in any of the formulae $\Phi^{\widehat{X}}$ and $\Phi^{\widehat{Y}}$.

We conclude that we can write $\mathcal{C}^{\widehat{X}} = \mathcal{C}_1 \cup \mathcal{C}_2$ and $\mathcal{C}^{\widehat{Y}} = \mathcal{C}_1 \cup \mathcal{C}_3$, where $\mathcal{C}_1 \subseteq \mathcal{C}_{\text{rem}}$ and $\mathcal{C}_2, \mathcal{C}_3 \subseteq \mathcal{F}_{\text{u}}$ as we wanted. □

In order to further analyse the probability distribution of the output of Algorithm 4, we introduce the following definition.

Definition 4.36 (run, $\mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)$, $\tau_{\mathcal{R}}(\Phi, \mathcal{M}, u, \Lambda)$, $\mathcal{V}_{\text{set}}(R)$, $\mathcal{V}_{\text{d}}(R)$, $\mathcal{F}_{\text{u}}(R)$, $\mathcal{F}_{\text{d}}(R)$, $\mathcal{C}_{\text{rem}}(R)$)**.** *A* run *of Algorithm 4 is a sequence of all the random choices $(\widehat{X}(v), \widehat{Y}(v))$ made in line 17 when executing Algorithm 4. Let $\mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)$ be the set of all possible runs of Algorithm 4 for the inputs $\Phi, \mathcal{M}, u, \Lambda$ and let $\tau_{\mathcal{R}}(\Phi, \mathcal{M}, u, \Lambda)$ be the probability distribution that Algorithm 4 yields on $\mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)$. Each run $R \in \mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)$ determines the output $(\widehat{X}, \widehat{Y})$ and the sets $\mathcal{V}_{\text{set}}(R), \mathcal{V}_{\text{d}}(R), \mathcal{F}_{\text{u}}(R), \mathcal{F}_{\text{d}}(R), \mathcal{C}_{\text{rem}}(R)$ that are computed in Algorithm 4.*

With the aim of applying Proposition 4.33, we extend the coupling $(\widehat{X}, \widehat{Y})$ to all marked and auxiliary variables.

Definition 4.37 (The coupling (X, Y))**.** *Let $R \in \mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)$ and let $(\widehat{X}, \widehat{Y})$ be the corresponding output of the run R. Let $\leq_{\mathcal{V}}$ be a total order on the variables of Φ and let $v_1 \leq_{\mathcal{V}} v_2 \leq_{\mathcal{V}} \cdots \leq_{\mathcal{V}} v_t$ be the variables in $(\mathcal{V}_{\text{m}} \cup \mathcal{V}_{\text{a}}) \setminus \mathcal{V}_{\text{set}}$. We extend the assignments $\widehat{X}, \widehat{Y} \colon \mathcal{V}_{\text{set}} \to \{\mathsf{F}, \mathsf{T}\}$ to $v_1, v_2, \ldots, v_t$ inductively (as follows) to obtain a coupling (X, Y) such that X follows the distribution $\mu_{\Omega^{\Lambda \cup u \mapsto \mathsf{T}}|_{(\mathcal{V}_{\text{m}} \cup \mathcal{V}_{\text{a}}) \setminus \mathcal{V}_{\text{set}}}}$ and Y follows the distribution $\mu_{\Omega^{\Lambda \cup u \mapsto \mathsf{F}}|_{(\mathcal{V}_{\text{m}} \cup \mathcal{V}_{\text{a}}) \setminus \mathcal{V}_{\text{set}}}}$. Assume that X and Y are defined on $\mathcal{V}_{\text{set}} \cup \{v_1, v_2, \ldots, v_{j-1}\}$ for $j \in \{1, 2, \ldots, t\}$. Then we sample $(X(v_j), Y(v_j))$ from the optimal/monotone coupling of the marginal distributions (on v_j) of μ_{Ω^X} and μ_{Ω^Y}.*

Remark 4.38. *When $R \in \mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)$ follows the probability distribution $\tau_{\mathcal{R}}(\Phi, \mathcal{M}, u, \Lambda)$ (Definition 4.36), the pair of random assignments (X, Y) of Definition 4.37 is a coupling of the distributions $\mu_{\Omega^{\Lambda \cup u \mapsto \mathsf{T}}}|_{\mathcal{V}_{\mathrm{m}} \cup \mathcal{V}_{\mathrm{a}}}$ and $\mu_{\Omega^{\Lambda \cup u \mapsto \mathsf{F}}}|_{\mathcal{V}_{\mathrm{m}} \cup \mathcal{V}_{\mathrm{a}}}$.*

In Lemma 4.39 we bound the probabilities $\Pr(X(v) \neq Y(v)|R)$ for any $R \in \mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)$ and $v \in (\mathcal{V}_{\mathrm{m}} \cup \mathcal{V}_{\mathrm{a}}) \setminus \mathcal{V}_{\mathrm{set}}(R)$.

Lemma 4.39. *Let $R \in \mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)$. Let (X, Y) be the coupling of Definition 4.37. Then for any $v \in (\mathcal{V}_{\mathrm{m}} \cup \mathcal{V}_{\mathrm{a}}) \setminus \mathcal{V}_{\mathrm{set}}(R)$ we have $\Pr(X(v) \neq Y(v)|R) \leq 2^{-r_0 k+1}/k$.*

Proof. Let $\widehat{X}$ and $\widehat{Y}$ be the output of Algorithm 4 for the run R. Let $v_1, v_2, \ldots, v_t$ be the variables in $(\mathcal{V}_{\mathrm{m}} \cup \mathcal{V}_{\mathrm{a}}) \setminus \mathcal{V}_{\mathrm{set}}(R)$ in the order that they are considered in Definition 4.37. Let $j \in \{1, 2, \ldots, t\}$ and let $\Lambda', \Lambda'' \colon \mathcal{V}_{\mathrm{set}}(R) \cup \{v_1, v_2, \ldots, v_{j-1}\} \to \{\mathsf{F}, \mathsf{T}\}$ be two assignments such that $\Lambda'|_{\mathcal{V}_{\mathrm{set}}} = \widehat{X}$ and $\Lambda''|_{\mathcal{V}_{\mathrm{set}}} = \widehat{Y}$. When X agrees with Λ' and Y agrees with Λ'', the values $X(v_j)$ and $Y(v_j)$ are sampled from the optimal/monotone coupling between the marginals on v_j of the distributions $\mu_{\Omega^{\Lambda'}}$ and $\mu_{\Omega^{\Lambda''}}$. Let us denote these marginals by ν_X and ν_Y respectively. Thus, by the coupling lemma (Proposition 4.32) and Proposition 4.5 (or Lemma 4.23) on the marginals of marked and auxiliary variables, we have

$$\begin{aligned}\Pr\left(X(v_j) \neq Y(v_j)|\Lambda', \Lambda''\right) &= d_{\mathrm{TV}}(\nu_X, \nu_Y) = \left|\Pr(X(v_j) = \mathsf{T}|\Lambda') - \Pr(Y(v_j) = \mathsf{T}|\Lambda'')\right| \\ &\leq |\Pr(X(v_j) = \mathsf{T}|\Lambda') - 1/2| + |1/2 - \Pr(Y(v_j) = \mathsf{T}|\Lambda'')| \\ &\leq \exp\left(\frac{1}{k 2^{r_0 k}}\right) - 1.\end{aligned}$$

Applying the inequality $e^z \leq 1 + 2z$ for $z \in (0, 1)$, we find that $\Pr\left(X(v_j) \neq Y(v_j)|\Lambda', \Lambda''\right) \leq 2^{-r_0 k+1}/k$. Thus, from the arbitrary choice of Λ', Λ'' and the law of total probability we conclude that the bound $\Pr\left(X(v_j) \neq Y(v_j)|R\right) \leq 2^{-r_0 k+1}/k$ holds. □

Combining all the results presented up to this stage in the current section allows us relate the sum $\sum_{v \in \mathcal{V}_{\mathrm{m}} \setminus (S \cup \{u\})} \left|\mathcal{I}^{\Lambda}(u \to v)\right|$ to the coupling process over auxiliary variables. In fact, we bound this sum of influences between marked variables by the expected number of failed clauses in the coupling process on auxiliary variables. Recall that here $r = r_0 - \delta$.

Lemma 4.40. *There is an integer k_0 such that for any $k \geq k_0$ and any density α with $\alpha \leq 2^{(r_0 - \delta)k}/k^3$ the following holds w.h.p. over the choice of the random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. Let $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{a}}, \mathcal{V}_{\mathrm{c}})$ be an $(r_0 - \delta, r_0, r_0, 2r_0)$-marking of Φ, and let $u \in \mathcal{V}_{\mathrm{m}}$ and $\Lambda \colon S \to \{\mathsf{F}, \mathsf{T}\}$ with $S \subseteq \mathcal{V}_{\mathrm{m}} \setminus \{u\}$. Then for a random run R of the coupling process on the auxiliary variables (Algorithm 4), we have*

$$\sum_{v \in \mathcal{V}_{\mathrm{m}} \setminus (S \cup \{u\})} \left|\mathcal{I}^{\Lambda}(u \to v)\right| \leq 2^{-r_0 k+1} \mathbb{E}\left[|\mathcal{F}_{\mathrm{u}}(R)|\right].$$

Proof. Let (X, Y) be the coupling in Definition 4.37 for a (random) run $R \sim \tau_{\mathcal{R}}(\Phi, \mathcal{M}, u, \Lambda)$ of Algorithm 4. We are going to show that

$$\Pr(X(v) = Y(v)|R) = 1 \text{ for all } v \in V := (\mathcal{V}_{\mathrm{m}} \cup \mathcal{V}_{\mathrm{a}}) \setminus (\mathcal{V}_{\mathrm{set}}(R) \cup \mathrm{var}(\mathcal{F}_{\mathrm{u}}(R))). \tag{4.27}$$

Let $\widehat{X}, \widehat{Y} \colon \mathcal{V}_{\text{set}}(R) \to \{\mathsf{F}, \mathsf{T}\}$ be the output of Algorithm 4 for the run R. By Lemma 4.35 we conclude that we can write $\mathcal{C}^{\widehat{X}} = \mathcal{C}_1 \cup \mathcal{C}_2$ and $\mathcal{C}^{\widehat{Y}} = \mathcal{C}_1 \cup \mathcal{C}_3$, where $\mathcal{C}_1 \subseteq \mathcal{C}_{\text{rem}}(R)$ and $\mathcal{C}_2, \mathcal{C}_3 \subseteq \mathcal{F}_{\text{u}}(R)$. Thus, the variables in V (see (4.27) for a definition of V) either appear in a clause in $\mathcal{C}_1$ or they are not present in any of the formulae $\Phi^{\widehat{X}}$ and $\Phi^{\widehat{Y}}$. Moreover, by Proposition 4.34, we have $\operatorname{var}(c) \cap \operatorname{var}(c') = \emptyset$ for all $c \in \mathcal{C}_{\text{rem}}(R)$ and $c' \in \mathcal{F}_{\text{u}}(R)$. We conclude that the distributions $\mu_{\Omega^{\widehat{X}}}\big|_V$ and $\mu_{\Omega^{\widehat{Y}}}\big|_V$ agree – both are the uniform distribution over the satisfying assignments of the CNF formula $(V, \mathcal{C}_1)$. Let $v_1, v_2, \ldots, v_t$ be the variables in V in the order they are considered in the definition of the coupling (X, Y). By induction on $j \in \{1, 2, \ldots, t\}$, the marginals on v_j in Definition 4.37 are the same when coupling $X(v_j)$ and $Y(v_j)$. Thus, we have $X(v_j) = Y(v_j)$ for all $j \in \{1, 2, \ldots, t\}$.

Since $S \cup \{u\} \subseteq \mathcal{V}_{\text{set}}(R) \subseteq S \cup \{u\} \cup \mathcal{V}_{\text{a}}$, we have $\mathcal{V}_{\text{m}} \setminus V = S \cup \{u\} \cup (\mathcal{V}_{\text{m}} \cap \operatorname{var}(\mathcal{F}_{\text{u}}(R)))$. In light of Lemma 4.39 and (4.27), we find that

$$\sum_{v \in \mathcal{V}_{\text{m}} \setminus (S \cup \{u\})} \Pr(X(v) \neq Y(v) | R) \leq \sum_{v \in \mathcal{V}_{\text{m}} \cap \operatorname{var}(\mathcal{F}_{\text{u}}(R))} \Pr(X(v) \neq Y(v) | R) \leq \frac{2}{k} 2^{-r_0 k} |\operatorname{var}(\mathcal{F}_{\text{u}}(R))|.$$

From $|\operatorname{var}(\mathcal{F}_{\text{u}}(R))| \leq k |\mathcal{F}_{\text{u}}(R)|$ we conclude that

$$\sum_{v \in \mathcal{V}_{\text{m}} \setminus (S \cup \{u\})} \Pr(X(v) \neq Y(v) | R) \leq 2^{-r_0 k + 1} |\mathcal{F}_{\text{u}}(R)|. \tag{4.28}$$

In the rest of this proof we are going to aggregate (4.28) over $R \in \mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)$ with the aim of applying Proposition 4.33. Let (X, Y) be the coupling in Definition 4.37 for a (random) run $R \sim \tau_{\mathcal{R}}(\Phi, \mathcal{M}, u, \Lambda)$ of Algorithm 4. We have

$$\begin{aligned}
\sum_{v \in \mathcal{V}_{\text{m}} \setminus (S \cup \{u\})} \Pr(X(v) \neq Y(v)) &= \sum_{v \in \mathcal{V}_{\text{m}} \setminus (S \cup \{u\})} \sum_{R \in \mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)} \Pr(R) \Pr(X(v) \neq Y(v) | R) \\
&= \sum_{R \in \mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)} \Pr(R) \sum_{v \in \mathcal{V}_{\text{m}} \setminus (S \cup \{u\})} \Pr(X(v) \neq Y(v) | R) \\
&\leq 2^{-r_0 k + 1} \sum_{R \in \mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)} \Pr(R) |\mathcal{F}_{\text{u}}(R)| \\
&= 2^{-r_0 k + 1} \mathbb{E}\left[|\mathcal{F}_{\text{u}}(R)| \right].
\end{aligned}$$

Finally, we note that we can indeed apply Proposition 4.33 to the restriction of X and Y on $\mathcal{V}_{\text{m}}$ as (X, Y) is a coupling of the distributions $\mu_{\Omega^{\Lambda \cup u \mapsto \mathsf{T}}}\big|_{\mathcal{V}_{\text{m}} \cup \mathcal{V}_{\text{a}}}$ and $\mu_{\Omega^{\Lambda \cup u \mapsto \mathsf{F}}}\big|_{\mathcal{V}_{\text{m}} \cup \mathcal{V}_{\text{a}}}$ (Remark 4.38). This finishes the proof. □

In the remainder of this section we bound $\mathbb{E}\left[|\mathcal{F}_{\text{u}}(R)| \right]$, which would complete our proof of Lemma 4.9 when combined with Lemma 4.40. In order to do this we exploit the fact that $\mathcal{F}_{\text{u}}(R) \cup \mathcal{F}_{\text{d}}(R)$ is connected in G_Φ (Proposition 4.34), the local sparsity properties of random CNF formulae and the properties of the marking $(\mathcal{V}_{\text{m}}, \mathcal{V}_{\text{a}}, \mathcal{V}_{\text{c}})$. It is important that the bound on $\mathbb{E}\left[|\mathcal{F}_{\text{u}}(R)| \right]$ is $\operatorname{poly}(k) \log n$ in order to conclude fast mixing time of the ρ-uniform-block Glauber dynamics when applying the spectral independence framework. First, we bound the probability that some good clauses are failed in Algorithm 4. At first glance this seems to be a

straightforward task thanks to the fact that the marginals of marked and auxiliary variables are close to 1/2 (see Proposition 4.5). However, for any good clauses c_1 and c_2, the events that $c_1 \in \mathcal{F}_{\mathrm{d}}(R) \cup \mathcal{F}_{\mathrm{u}}(R)$ and $c_2 \in \mathcal{F}_{\mathrm{d}}(R) \cup \mathcal{F}_{\mathrm{u}}(R)$ may not be independent; any value given to the variables in c_1 may affects the marginals of the variables in c_2 and whether these variables are considered by the coupling process or not. However, we show that, as long as c_1 and c_2 do not share good variables, these dependencies are not very strong and we can indeed bound the probability that $c_1, c_2 \in \mathcal{F}_{\mathrm{d}}(R) \cup \mathcal{F}_{\mathrm{u}}(R)$ with a careful probability argument that analyses the coupling process step by step, see Lemma 4.44. With this in mind, we introduce the following definitions.

Definition 4.41 ($\mathcal{R}_t(\Phi, \mathcal{M}, u, \Lambda)$, $\mathcal{A}_{\leq t}$)**.** *For a positive integer t, we let $\mathcal{R}_t(\Phi, \mathcal{M}, u, \Lambda)$ be the set containing for each $R \in \mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)$ a tuple with the first $\min\{t, \mathrm{length}(R)\}$ entries of the sequence R. That is, $\mathcal{R}_t(\Phi, \mathcal{M}, u, \Lambda)$ is the set containing all possible sequences of the first t choices that Algorithm 4 makes in line 17. Note that if $R \in \mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)$ has $\mathrm{length}(R) \leq t$, then $R \in \mathcal{R}_t(\Phi, \mathcal{M}, u, \Lambda)$. Each $R_t \in \mathcal{R}_t(\Phi, \mathcal{M}, u, \Lambda)$ determines two partial assignments Λ' and Λ'' of marked and auxiliary variables that correspond to the assignments $\widehat{X}$ and $\widehat{Y}$ after $\mathrm{length}(R_t)$ iterations of line 17 following R_t. Let $\mathcal{A}_{\leq t}$ be the σ-algebra containing all the subsets of $\mathcal{R}_t(\Phi, \mathcal{M}, u, \Lambda)$.*

Intuitively, $\mathcal{A}_{\leq t}$ contains all the possible events that may occur in the first t iterations of line 17, which is the only randomised operation in Algorithm 4. When bounding the probability that a clause is failed, we will express this event in terms of events concerning the values that $\widehat{X}$ and $\widehat{Y}$ take on its variables. This motivates Definition 4.42.

Definition 4.42 ($D_v(j)$)**.** *We define the following events for variable $v \in \mathcal{V}_{\mathrm{a}}$ and a random run $R \sim \tau_{\mathcal{R}}(\Phi, \mathcal{M}, u, \Lambda)$ of Algorithm 4. Let $D_v(1)$ be the event that $v \in \mathcal{V}_{\mathrm{set}}(R)$ and $\widehat{X}(v) \neq \widehat{Y}(v)$. Let $D_v(2)$ be the event that $v \in \mathcal{V}_{\mathrm{set}}(R)$ and $\widehat{X}(v) = \mathsf{F}$. Let $D_v(3)$ be the event that $v \in \mathcal{V}_{\mathrm{set}}(R)$ and $\widehat{X}(v) = \mathsf{T}$. Let $D_v(4)$ be the event that $v \in \mathcal{V}_{\mathrm{set}}(R)$ and $\widehat{Y}(v) = \mathsf{F}$. Let $D_v(5)$ be the event that $v \in \mathcal{V}_{\mathrm{set}}(R)$ and $\widehat{Y}(v) = \mathsf{T}$.*

Finally, in order to study the events $D_v(j)$ for $v \in V$ we will have to reason about the first time that a variable in V is added to $\mathcal{V}_{\mathrm{set}}(R)$, which motivates the following definition.

Definition 4.43 ($\tau(V)$, $f(V)$)**.** *For a set of auxiliary variables V, we let $\tau(V)$ be the random variable that takes the value t if the first time that a variable in V is added to $\mathcal{V}_{\mathrm{set}}(R)$ in Algorithm 4 is the t-th time line 17 is executed, and we denote by $f(V)$ this variable. We set $\tau(V) = \infty$ if $V \cap \mathcal{V}_{\mathrm{set}}(R) = \emptyset$, in which case $f(V)$ is not defined.*

We now have all the tools that we need to analyse the coupling process step by step.

Lemma 4.44. *Let $V \subseteq \mathcal{V}_{\mathrm{a}}$ and let $i_v \in \{1, 2, 3, 4, 5\}$ for each $v \in V$. Let $h(1) = 2^{-r_0 k + 1}/k$ and $h(i) = \frac{\exp(1/k)}{2}$ for $i \in \{2, 3, 4, 5\}$. Then, we have*

$$\Pr_{R \sim \tau_{\mathcal{R}}(\Phi, \mathcal{M}, u, \Lambda)} \left(\bigcap\nolimits_{v \in V} D_v(i_v) \right) \leq \prod_{v \in V} h(i_v).$$

Proof. We are going to prove, for any positive integer t and $A \in \mathcal{A}_{\leq t}$,

$$\Pr\left(\bigcap\nolimits_{v\in V} D_v(i_v)\middle| A, \tau(V) = t\right) \leq \prod_{v\in V} h(i_v). \tag{4.29}$$

The lemma will then follow from the arbitrary choice of A and t and the law of total probability.

We carry out the proof of (4.29) by induction on $M = |V|$. Equation (4.29) holds when V is empty. Let us assume that (4.29) holds when $|V| < M$. Let V be a set of auxiliary variables with $M = |V|$ and indexes i_v for all $v \in V$, let t be a positive integer and let $A \in \mathcal{A}_{\leq t}$. To simplify the notation, for each $w \in V$ we define $A_t(w, V) = A \cap [\tau(V) = t] \cap [f(V) = w]$. Then, we have

$$\begin{aligned}\Pr\left(\bigcap\nolimits_{v\in V} D_v(i_v)\middle| A, \tau(V) = t\right) \leq \sum_{w\in V} &\Pr\left(f(V) = w | A, \tau(V) = t\right) \cdot \Pr\left(D_w(i_w)|\, A_t(w, V)\right) \\ &\cdot \Pr\left(\bigcap\nolimits_{v\in V\setminus\{w\}} D_v(i_v)\middle| A_t(w, V), D_w(i_w)\right).\end{aligned}$$

We note that $\tau(V \setminus \{w\}) > t$ when conditioning on $\tau(V) = t$ and $f(V) = w$. Let $A' = A_t(w, V) \cap D_w(i_w)$. We have

$$\begin{aligned}\Pr\left(\bigcap\nolimits_{v\in V\setminus\{w\}} D_v(i_v)\middle| A'\right) = \sum_{j=t+1}^{\infty} &\Pr\left(\tau(V \setminus \{w\}) = j | A'\right) \\ &\cdot \Pr\left(\bigcap\nolimits_{v\in V\setminus\{w\}} D_v(i_v)\middle| A', \tau(V \setminus \{w\}) = j\right).\end{aligned}$$

By our induction hypothesis for $V \setminus \{w\}$, the condition $\tau(V \setminus \{w\}) = j$ and the event $A' \in \mathcal{A}_{\leq j}$, we find that

$$\Pr\left(\bigcap\nolimits_{v\in V\setminus\{w\}} D_v(i_v)\middle| A'\right) \leq \sum_{j=t+1}^{\infty} \Pr\left(\tau(V \setminus \{w\}) = j | A'\right) \prod_{v\in V\setminus\{w\}} h(i_v) \leq \prod_{v\in V\setminus\{w\}} h(i_v).$$

As a consequence, we obtain

$$\begin{aligned}\Pr\left(\bigcap\nolimits_{v\in V} D_v(i_v)\middle| A, \tau(V) = t\right) \leq \sum_{w\in V} &\Pr\left(f(V) = w | A, \tau(V) = t\right) \cdot \Pr\left(D_w(i_w)|\, A_t(w, V)\right) \\ &\cdot \prod_{v\in V\setminus\{w\}} h(i_v).\end{aligned}$$

We are going to show that $\Pr(D_w(i_w)|A_t(w, V)) \leq h(i_w)$. Once we have proved this, the proof of (4.29) is completed by noting that $\sum_{w\in V} \Pr\left(f(V) = w|A, \tau(V) = t\right) = 1$.

Let us now bound $\Pr(D_w(i_w)|A_t(w, V))$. Recall here that $A_t(w, V)$ implies the event $w \in \mathcal{V}_{\text{set}}(R)$. Recall also that $A_t(w, V) \in \mathcal{A}_{\leq t}$, see Definition 4.41. For each $R_t \in A_t(w, V) \subseteq \mathcal{R}_t(\Phi, \mathcal{M}, u, \Lambda)$, we are going to apply Proposition 4.5 and the fact that $\widehat{X}(w)$ and $\widehat{Y}(w)$ follow the optimal coupling between two marginal distributions on v of the form $\mu_{\Omega^{\Lambda'}}$ and $\mu_{\Omega^{\Lambda''}}$ for some assignments Λ', Λ'' on some marked and auxiliary variables that are determined by R_t. Here it is important for applying Proposition 4.5 that the event $A_t(w, V)$ is in $\mathcal{A}_{\leq t}$, so every

partial run $R_t \in A_t(w,V)$ only gives information about what has happened in Algorithm 4 before w is added to $\mathcal{V}_{\mathrm{set}}(R)$. Thus, aggregating over all possible runs $R_t \in A_t(w,V)$, we find that

$$\begin{aligned}\max\left\{\Pr\left(\widehat{X}(w)=\mathsf{F}\middle|\, A_t(w,V)\right),\Pr\left(\widehat{X}(w)=\mathsf{T}\middle|\, A_t(w,V)\right)\right\} &\leq \frac{1}{2}\exp\left(\frac{1}{k2^{r_0 k}}\right)\\ &\leq \frac{1}{2}\exp\left(\frac{1}{k}\right),\end{aligned} \tag{4.30}$$

where the probability is over the random run $R \sim \tau_{\mathcal{R}}(\Phi,\mathcal{M},u,\Lambda)$. The bound (4.30) also applies with $\widehat{Y}$ instead of $\widehat{X}$. In particular, we conclude that $\Pr(D_w(j)|A_t(w,V)) \leq \exp(1/k)/2 = h(j)$ for all $j \in \{2,3,4,5\}$. Moreover, using the definition of optimal coupling for two Bernoulli distributions, the probability that $\widehat{X}(w) \neq \widehat{Y}(w)$ can be bounded as

$$\begin{aligned}\Pr\left(\widehat{X}(w)\neq\widehat{Y}(w)\middle|\, A_t(w,V)\right) &= \left|\Pr\left(\widehat{X}(w)=\mathsf{T}\middle|\, A_t(w,V)\right)-\Pr\left(\widehat{Y}(w)=\mathsf{T}\middle|\, A_t(w,V)\right)\right|\\ &\leq \left|\Pr\left(\widehat{X}(w)=\mathsf{T}\middle|\, A_t(w,V)\right)-1/2\right|\\ &\quad+\left|1/2-\Pr\left(\widehat{Y}(w)=\mathsf{T}\middle|\, A_t(w,V)\right)\right|\\ &\leq \exp\left(\frac{1}{k2^{r_0 k}}\right)-1.\end{aligned}$$

Hence, applying the bound $e^z \leq 1+2z$ for $z \in (0,1)$ and the definition of the event $D_{v_j}(1)$, we have $\Pr(D_{v_j}(1)|A_t(w,V)) \leq 2/(k2^{r_0 k}) = h(1)$. This finishes the proof of (4.29). From the arbitrary choice of A and t and the law of total probability, the statement follows. □

We can now bound the probability that some good clauses are failed with the help of Lemma 4.44.

Lemma 4.45. *Let Φ,u,Λ be the input of Algorithm 4. Let $c_1,\ldots,c_\ell \in \mathcal{C}_{\mathrm{good}}$ such that the variable u does not appear in any of the clauses in $c_1,\ldots,c_\ell$, and $\mathrm{var}(c_i)\cap\mathrm{var}(c_j)\cap\mathcal{V}_{\mathrm{good}}=\emptyset$ for all $1\leq i<j\leq\ell$. Then, for $R\sim\tau_{\mathcal{R}}(\Phi,\mathcal{M},u,\Lambda)$, we have $\Pr(c_1,\ldots,c_\ell\in\mathcal{F}_{\mathrm{d}}(R)\cup\mathcal{F}_{\mathrm{u}}(R)) \leq 2^{(-r_0k+4)\ell}$.*

Proof. Let $c_1,\ldots,c_\ell$ be some good clauses of Φ as in the statement. The hypothesis that u does not appear in any of these clauses is necessary as if $u\in\mathrm{var}(c)$ then $c\in\mathcal{F}_{\mathrm{d}}(R)$ by definition. We consider a random run $R\sim\tau_{\mathcal{R}}(\Phi,\mathcal{M},u,\Lambda)$ of Algorithm 4 and let $\widehat{X},\widehat{Y}$ be the (random) output of Algorithm 4 for the run R. For $j\in\{1,2,\ldots,\ell\}$, let $F_j(1)$ be the event that there is $v\in\mathrm{var}(c_j)\cap\mathcal{V}_{\mathrm{a}}$ such that $v\in\mathcal{V}_{\mathrm{set}}(R)$ and $\widehat{X}(v)\neq\widehat{Y}(v)$, let $F_j(2)$ be the event that $\mathrm{var}(c_j)\cap\mathcal{V}_{\mathrm{a}}\subseteq\mathcal{V}_{\mathrm{set}}(R)$ and c_j is unsatisfied by $\widehat{X}$, and let $F_j(3)$ be the event that $\mathrm{var}(c_j)\cap\mathcal{V}_{\mathrm{a}}\subseteq\mathcal{V}_{\mathrm{set}}(R)$ and c_j is unsatisfied by $\widehat{Y}$. In light of Proposition 4.34, we have $[c_1,\ldots,c_\ell\in\mathcal{F}_{\mathrm{d}}(R)\cup\mathcal{F}_{\mathrm{u}}(R)] = \bigcap_{j=1}^{\ell}(F_j(1)\cup F_j(2)\cup F_j(3))$. We obtain

$$\Pr\left(\bigcap\nolimits_{j=1}^{\ell}(F_j(1)\cup F_j(2)\cup F_j(3))\right) \leq \sum_{(i_1,i_2,\ldots,i_\ell)\in\{1,2,3\}^\ell}\Pr\left(\bigcap\nolimits_{j=1}^{\ell}F_j(i_j)\right). \tag{4.31}$$

We note that $F_j(1) = \bigcup_{v \in \mathrm{var}(c_j) \cap \mathcal{V}_\mathrm{a}} D_v(1)$, see Definition 4.42. Let $(i_1, i_2, \ldots, i_\ell) \in \{1,2,3\}^\ell$, and let $I_1 = \{j : i_j = 1\}$, $I_2 = \{j : i_j = 2\}$ and $I_3 = \{j : i_j = 3\}$. If the event $\bigcap_{j \in I_1} F_j(i_j)$ holds, then, for each $j \in I_1$ there is a variable $u_j \in \mathrm{var}(c_j) \cap \mathcal{V}_\mathrm{a}$ such that $D_{u_j}(1)$ holds. Thus, for the set of tuples $T = \prod_{j \in I_1} (\mathrm{var}(c_j) \cap \mathcal{V}_\mathrm{a})$, where $\prod$ here denotes the cartesian product of sets, we have

$$\bigcap_{j \in I_1} F_j(i_j) = \bigcup_{(u_1, u_2, \ldots, u_{|I_1|}) \in T} \bigcap_{j \in I_1} D_{u_j}(1). \tag{4.32}$$

Now we explain how we bound the probability of the event $\left(\bigcap_{j \in I_2 \cup I_3} F_j(i_j)\right) \cap \left(\bigcap_{j \in I_1} D_{u_j}(1)\right)$ for a tuple $(u_1, u_2, \ldots, u_{|I_1|}) \in T$. We are going to show that

$$\Pr\left(\left(\bigcap\nolimits_{j \in I_2 \cup I_3} F_j(i_j)\right) \cap \left(\bigcap\nolimits_{j \in I_1} D_{u_j}(1)\right)\right) \leq \left(\frac{\exp(1/k)}{2}\right)^{(k-3) r_0 |I_2 \cup I_3|} \left(\frac{2}{k 2^{r_0 k}}\right)^{|I_1|}. \tag{4.33}$$

The proof of (4.33) is not as straightforward as it may seem at first glance due to the dependencies among the events $F_j(i_j)$, $D_{u_j}(1)$. The key idea is re-writing the LHS of (4.33) as in the statement of Lemma 4.44. Indeed we note that for each $j \in I_2$ and for each variable $v \in \mathrm{var}(c_j) \cap \mathcal{V}_\mathrm{a}$, the event $F_j(2)$ implies that there is $i_v \in \{2,3\}$ such that $D_v(i_v)$ holds, concluding $F_j(2) = \bigcap_{v \in \mathrm{var}(c_j) \cap \mathcal{V}_\mathrm{a}} D_v(i_v)$, see Definition 4.42. Analogously, for each $j \in I_3$ and for each variable $v \in \mathrm{var}(c_j) \cap \mathcal{V}_\mathrm{a}$, we find $i_v \in \{4,5\}$ such that $F_j(3) = \bigcap_{v \in \mathrm{var}(c_j) \cap \mathcal{V}_\mathrm{a}} D_v(i_v)$. Therefore, we have

$$\left(\bigcap\nolimits_{j \in I_2 \cup I_3} F_j(i_j)\right) \cap \left(\bigcap\nolimits_{j \in I_1} D_{u_j}(1)\right) = \bigcap_{v \in V_f} D_v(i_v), \tag{4.34}$$

where V_f contains exactly all the auxiliary variables in the clauses c_j with $j \in I_2 \cup I_3$ and the variables $u_1, u_2, \ldots, u_{|I_1|}$. Recall now that each good clause contains at least $r_0(k-3)$ auxiliary variables, and, thus, the bound given in (4.33) follows from (4.34) and Lemma 4.44. Combining (4.33), (4.32) and (4.31), and counting the number of tuples in T, we conclude that

$$\begin{aligned}
\Pr\left(\bigcap\nolimits_{j=1}^{\ell} (F_j(1) \cup F_j(2) \cup F_j(3))\right) &\leq \sum_{(i_1, i_2, \ldots, i_\ell) \in \{1,2,3\}^\ell} k^{|I_1|} \left(\frac{\exp(1/k)}{2}\right)^{(k-3) r_0 |I_2 \cup I_3|} \left(\frac{2}{k 2^{r_0 k}}\right)^{|I_1|} \\
&\leq \sum_{(i_1, i_2, \ldots, i_\ell) \in \{1,2,3\}^\ell} \left(\frac{e 2^{3 r_0}}{2^{k r_0}}\right)^{|I_2 \cup I_3|} \left(\frac{2}{2^{r_0 k}}\right)^{|I_1|} \\
&= \left(\frac{e 2^{3 r_0}}{2^{k r_0}} + \frac{e 2^{3 r_0}}{2^{k r_0}} + \frac{2}{2^{r_0 k}}\right)^{\ell},
\end{aligned}$$

where we used the multinomial theorem. The result now follows from $2e 2^{3 r_0} + 2 \leq 2^4$. □

Following [49] and motivated by Lemma 4.45, we introduce the combinatorial structure that we use in our proof of Lemma 4.9 to bound the expected number of failed clauses.

Definition 4.46 ($G^{\leq k}$, $\mathcal{D}_3(G_\Phi, c, \ell)$)**.** *For a graph $G = (V, E)$ and a positive integer k, let $G^{\leq k}$ be the graph with vertex set V in which vertices u and v are connected if and only if there is a path from u to v in G of length at most k. Given the graph G_Φ, a clause c and a positive integer ℓ, let $\mathcal{D}_3(G_\Phi, c, \ell)$ be the set of subsets $T \subseteq V(G_\Phi)$ such that the following holds:*

1. $|T| = \ell$ *and* $c \in T$*;*
2. *for any* $c_1, c_2 \in T$, $\operatorname{var}(c_1) \cap \operatorname{var}(c_2) \cap \mathcal{V}_{\text{good}} = \emptyset$*;*
3. *the graph* $G_\Phi^{\leq 3}[T]$*, which is the subgraph of* $G_\Phi^{\leq 3}$ *induced by* T*, is connected;*
4. *we have* $|T \cap \mathcal{C}_{good}| \geq (1 - 8/k)\ell$*.*

In [49] the authors consider connected sets in $G_\Phi^{\leq 4}$ instead of $G_\Phi^{\leq 3}$. Here we manage to perform our union bound on $\mathcal{D}_3(G_\Phi, c, \ell)$ thanks to the fact that the set of failed clauses is connected in our refinement of the coupling process.

Lemma 4.47 ([49, Corollary 8.19] for $G^{\leq 3}$)**.** *Let* $G = (V, E)$ *be a connected graph, let* $v \in V$ *and let* ℓ *be a positive integer. Let* $n_{G,\ell}(v)$ *denote the number of connected induced subgraphs of* G *with size* ℓ *containing* v*. Then, for* $\ell' = \min\{3\ell, |V|\}$*, we have* $n_{G^{\leq 3},\ell}(v) \leq 2^{\ell'} n_{G,\ell'}(v)$*.*

Proof. Let T be a connected subgraph of $G^{\leq 3}$ with size ℓ containing v. We claim that, for all positive ℓ, we can find a connected subset H of G with size $\ell' = \min\{3\ell, |V|\}$ containing T. The proof is straightforward by induction on ℓ, see [49, Lemma 8.18] for the analogous result on $G^{\leq 4}$. We note that there are at most $\binom{\ell'}{\ell-1} \leq 2^{\ell'}$ subsets T of H containing v that could be mapped to H by the previous construction. Hence, we conclude that $n_{G^{\leq 3},\ell}(v) \leq 2^{\ell'} n_{G,\ell'}(v)$ as we wanted. □

Lemma 4.48 ([49, Lemma 7.9] for $\mathcal{D}_3(G_\Phi, c, \ell)$)**.** *Let* ℓ *be an integer which is at least* $\log n$*. W.h.p. over the choice of* Φ*, every clause* $c \in \mathcal{C}_{\text{good}}$ *has the property that the size of* $\mathcal{D}_3(G_\Phi, c, \ell)$ *is at most* $(18k^2\alpha)^{3\ell}$*.*

Proof. This follows from bounding the number of connected sets of size ℓ in $G_\Phi^{\leq 3}$ that contain c by combining Lemmas 4.29 and 4.47. □

We have now all the tools that we need to bound the expected number of failed clauses in the coupling process given in Algorithm 4 and complete the proof of Lemma 4.9.

Lemma 4.9. *There is an integer* $k_0 \geq 3$ *such that for any integer* $k \geq k_0$ *and any density* α *with* $\alpha \leq 2^{r_0 k/3}/k^3$ *the following holds. W.h.p. over the choice of the random* k*-CNF formula* $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$*, for any* $(r_0 - \delta, r_0, r_0, 2r_0)$*-marking* $(\mathcal{V}_{\text{m}}, \mathcal{V}_{\text{a}}, \mathcal{V}_{\text{c}})$ *of* Φ*, the distribution* $\mu_\Omega|_{\mathcal{V}_{\text{m}}}$ *is* $(2^{-(r_0-\delta)k} \log n)$*-spectrally independent.*

Proof. Let $u \in \mathcal{V}_{\text{m}}$ and $\Lambda \colon S \to \{\mathsf{F}, \mathsf{T}\}$ with $S \subseteq \mathcal{V}_{\text{m}} \setminus \{u\}$. First of all, we apply Lemma 4.40 to bound $\sum_{v \in \mathcal{V}_{\text{m}} \setminus (S \cup \{u\})} \left|\mathcal{I}^\Lambda(u \to v)\right|$ by $2^{-r_0 k + 1} \mathbb{E}\left[|\mathcal{F}_{\text{u}}(R)|\right]$, where $R \sim \tau_{\mathcal{R}}(\Phi, \mathcal{M}, u, \Lambda)$. In the rest of this proof we show that $\Pr(|\mathcal{F}_{\text{u}}(R)| \geq 2k^4 \log n) \leq O(1/n)$ and, thus, for large enough n, using the fact that $|\mathcal{F}_{\text{u}}(R)| \leq m \leq \alpha n$, we have $\mathbb{E}\left[|\mathcal{F}_{\text{u}}(R)|\right] = \sum_{R \in \mathcal{R}(\Phi, \mathcal{M}, u, \Lambda)} \Pr(R) |\mathcal{F}_{\text{u}}(R)| \leq 4k^4 \log n$. Putting all this together, and using the fact that $8k^4 \leq 2^{\delta k}$ for large enough k (here $\delta = 0.00001$) we would obtain the bound $\sum_{v \in \mathcal{V}_{\text{m}} \setminus (S \cup \{u\})} \left|\mathcal{I}^\Lambda(u \to v)\right| \leq 8 \cdot 2^{-r_0 k} k^4 \log n \leq 2^{-(r_0 - \delta)k} \log n$ and, thus, the result would follow.

So to finish we just need to show that, w.h.p. over the choice of Φ, $\Pr(|\mathcal{F}_{\mathrm{u}}(R)| \geq 2k^4 \log n) \leq O(1/n)$. Let $L = \lceil 2k^4 \log n \rceil$ and let $\ell = \lceil 0.5k^4 \log n \rceil$. First, we are going to show that, w.h.p. over the choice of Φ, the following holds:

$$\text{if } Z \subseteq \mathcal{C} \text{ is connected and } |Z| = L, \text{ then } \exists c \in Z \cap \mathcal{C}_{\mathrm{good}} \text{ and } T \in \mathcal{D}_3(G_\Phi, c, \ell) \text{ with } T \subseteq Z. \tag{4.35}$$

In order to prove (4.35), we are going to find a large independent set of $Z \cap \mathcal{C}_{\mathrm{good}}$, and we are going to extend it with some clauses in $Z \cap \mathcal{C}_{\mathrm{bad}}$ to obtain $T \in \mathcal{D}_3(G_\Phi, c, \ell)$. We need three results that hold w.h.p. over the choice of Φ: Lemmas 4.15, 4.27 and 4.26. We note that we can apply Lemma 4.15 for $r = r_0 - \delta$ as our density satisfies $\alpha \leq 2^{r_0 k/3}/k^3 \leq \lceil 2^{(r_0-\delta)k} \rceil / k^3 = \Delta_r/k^3$, where $\delta = 0.00001$. For Z as in (4.35) we have $|Z| \geq 2k^4 \log n$, so by Lemma 4.27 with $a = 2k^4$, we find that $|\mathrm{var}(Z)| \geq 2k^4 \log n$ and, thus, in light of Lemma 4.15, we conclude that $|Z \cap \mathcal{C}_{\mathrm{good}}| \geq (1 - 1/k)|Z|$ and $|Z \cap \mathcal{C}_{\mathrm{bad}}| \leq |Z|/k$. From Lemma 4.26 with $b = 4k^4$, w.h.p. over the choice of Φ, all connected sets of clauses with size at most $4k^4 \log n$ have tree-excess at most $t := \max\{1, 8k^4 \log(ek^2\alpha)\}$. Thus, we can find $U \subseteq Z \cap \mathcal{C}_{\mathrm{good}}$ such that U is a forest (disjoint union of trees) and $|U| \geq (1 - 1/k)|Z| - t$. In particular, U is bipartite, so there is $I \subseteq U$ such that $\mathrm{var}(c) \cap \mathrm{var}(c') = \emptyset$ for all $c, c' \in I$ and $|I| \geq |U|/2 \geq (1 - 1/k)L/2 - t/2 \geq \frac{1}{2}k^4 \log n$, where the last inequality holds for large enough n. Let I' be an independent set of $Z \cap \mathcal{C}_{\mathrm{good}}$ with the largest possible size. Then we have shown that $|I'| \geq \ell = \lceil \frac{1}{2}k^4 \log n \rceil$.

We claim that the set $T' := I' \cup (Z \cap \mathcal{C}_{\mathrm{bad}})$ is connected in $(G_\Phi[Z])^{\leq 3}$, where $G_\Phi[Z]$ is the subgraph of G_Φ induced by Z. Assume for contradiction that T' is not connected in $(G_\Phi[Z])^{\leq 3}$. In this case, we can write $T' = S_1 \cup S_2$ such that for all $c_1 \in S_1$ and $c_2 \in S_2$, the shortest path between c_1 and c_2 through clauses in Z has length at least 4. Let $(c_1, c_2) \in S_1 \times S_2$ be the pair with the shortest path in Z, and let this path be $c_1 = e_1, e_2, \ldots, e_j = c_2$. Then $j \geq 5$ and $e_2, \ldots, e_{j-1} \in Z \setminus T'$. Moreover, we find that $\mathrm{var}(e_3) \cap \mathrm{var}(c) = \emptyset$ for all $c \in T'$ as otherwise $e_1, e_2, \ldots, e_j$ would not be the shortest path between S_1 and S_2. Moreover, since T' contain all bad clauses in Z, we conclude that e_3 is a good clause. It follows that $I' \cup \{e_3\}$ is an independent set of good clauses of Z, which contradicts the fact that I' has largest possible size among such sets.

Finally, as $|T'| \geq \ell$, we can find a good clause c and a subset T of T' with size ℓ such that $c \in T$, T is connected in $G_\Phi^{\leq 3}$ and $|T \cap \mathcal{C}_{\mathrm{bad}}| \leq |Z \cap \mathcal{C}_{\mathrm{bad}}| \leq L/k \leq 8\ell/k$. We conclude that $T \in \mathcal{D}_3(G_\Phi, c, \ell)$. This finishes the proof of (4.35).

In the rest of the proof we use (4.35) to bound $\Pr(|\mathcal{F}_{\mathrm{u}}(R)| \geq L)$. Recall that the set of failed clauses $\mathcal{F}_{\mathrm{d}}(R) \cup \mathcal{F}_{\mathrm{u}}(R)$ is connected (Proposition 4.34). If $|\mathcal{F}_{\mathrm{u}}(R)| \geq L$, then there is $Z \subseteq \mathcal{F}_{\mathrm{d}}(R) \cup \mathcal{F}_{\mathrm{u}}(R)$ with $|Z| = L$ such that Z is connected in G_Φ, and, thus, we can find c and T as in (4.35). We have shown that the event $|\mathcal{F}_{\mathrm{u}}(R)| \geq L$ is contained in the event that there

is a good clause c and $T \in \mathcal{D}_3(\Phi, c, \ell)$ such that $T \subseteq \mathcal{F}_{\mathrm{d}}(R) \cup \mathcal{F}_{\mathrm{u}}(R)$. As a consequence, we have

$$\begin{aligned}\Pr\left[|\mathcal{F}_{\mathrm{u}}(R)| \geq L\right] &\leq \sum_{c \in \mathcal{C}} \sum_{T \in \mathcal{D}_3(\Phi,c,\ell)} \Pr\left[T \subseteq \mathcal{F}_{\mathrm{d}}(R) \cup \mathcal{F}_{\mathrm{u}}(R)\right] \\ &\leq \sum_{c \in \mathcal{C}} \sum_{T \in \mathcal{D}_3(\Phi,c,\ell)} \Pr\left[T \cap \mathcal{C}_{\mathrm{good}} \subseteq \mathcal{F}_{\mathrm{d}}(R) \cup \mathcal{F}_{\mathrm{u}}(R)\right].\end{aligned}$$

We note that for any $T \in \mathcal{D}_3(\Phi, c, \ell)$ there is at most one good clause c' that contains the marked variable u. Thus, by definition of $\mathcal{D}_3(\Phi, c, \ell)$, there are at least $(1 - 8/k)\ell - 1$ good clauses in T that do not contain the variable u. Hence, we can apply Lemma 4.48 on the size of $\mathcal{D}_3(\Phi, c, \ell)$ and Lemma 4.45 on the probability of good clauses (that do not share good variables) failing to further obtain

$$\Pr\left[|\mathcal{F}_{\mathrm{u}}(R)| \geq L\right] \leq m \left(18k^2\alpha\right)^{3\ell} 2^{-(r_0 k - 4)[(1-8/k)\ell - 1]}.$$

In what follows it is essential that $\alpha \leq 2^{r_0 k/3}/k^3$, and this is the only proof in this paper where we need this bound on the density – other proofs only require the less restrictive bounds $\alpha \leq 2^{(r_0-\delta)k}/k^3$ or $\alpha \leq 2^{(r_0-3\delta)k}/k^3$. Thus, we conclude that

$$\Pr\left[|\mathcal{F}_{\mathrm{u}}(R)| \geq L\right] \leq m \left(18\frac{2^{r_0 k/3}}{k}\right)^{3\ell} 2^{-(r_0 k-4)(1-8/k)\ell}\, 2^{r_0 k - 4} = m \left(\frac{18^3}{k^3} 2^{8r_0 + 4(1-8/k)}\right)^{\ell} 2^{r_0 k - 4}.$$

Finally, for large enough k we find that $\Pr\left[|\mathcal{F}_{\mathrm{u}}(R)| \geq L\right] \leq m e^{-\ell} 2^{r_0 k} \leq m n^{-0.5k^4} 2^{r_0 k} = O(1/n)$ as we wanted. □

4.6.3 Mixing time of the ρ-uniform-block Glauber dynamics

Finally, we combine the results in this section with Lemma 4.8 to complete the proof of Lemma 4.10.

Remark 4.49. *The distribution $\mu_\Omega|_{\mathcal{V}_{\mathrm{m}}}$ on assignments of the marked variables (Definition 4.6) is b-marginally bounded for $b = 1 - (1/2)\exp(1/k)$ by Proposition 4.5 (or, equivalently, Lemmas 4.21 and 4.23). Since $\exp(1/k) \leq 1 + 2/k$, we have $b \geq 1/2 - 1/k \geq 1/e$ for $k \geq 8$.*

Lemma 4.10. *There is a function $k_0(\theta) = \Theta(\log(1/\theta))$ such that, for any $\theta \in (0, 1)$, for any integer $k \geq k_0(\theta)$ and any density α with $\alpha \leq 2^{0.039k}$ the following holds. W.h.p. over the choice of the random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$, for any $(r_0 - \delta, r_0, r_0, 2r_0)$-marking $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{a}}, \mathcal{V}_{\mathrm{c}})$ of Φ and for $\rho = \lceil 2^{-k-1}|\mathcal{V}_{\mathrm{m}}| \rceil$, the ρ-uniform-block Glauber dynamics for updating the marked variables has mixing time $T_{\mathrm{mix}}(\rho, \varepsilon/2) \leq T := \lceil 2^{2k+3} n^{\theta} \log \frac{2n}{\varepsilon^2} \rceil$.*

Proof. In view of Lemma 4.9, as $\alpha \leq 2^{0.039k} \leq 2^{r_0 k/3}/k^3$ for large enough k, w.h.p. over the choice of Φ, the distribution $\mu_\Omega|_{\mathcal{V}_{\mathrm{m}}}$ is η-spectrally independent for $\eta = 2^{-(r_0-\delta)k} \log n$. Moreover, this distribution is b-marginally bounded for $b = 1/e$ when $k \geq 8$. We are going to apply Lemma 4.8 with $V = \mathcal{V}_{\mathrm{m}}$, $\mu = \mu_\Omega|_{\mathcal{V}_{\mathrm{m}}}$, $M = |\mathcal{V}_{\mathrm{m}}|$ and $\kappa = 2^{-k-1}$. First, we check that the hypothesis $M \geq \frac{2}{\kappa}(4\eta/b^2 + 1)$ of Lemma 4.8 holds. By Corollary 4.25 with $r = r_0 - \delta$ and

$V = \mathcal{V}_{\mathrm{m}}$, we have $M \geq (r_0 - \delta)(k\alpha/\Delta_r)n = \Omega(n)$, so $M \geq \frac{2}{\kappa}(4\eta/b^2 + 1)$ holds for large enough n as $\frac{2}{\kappa}(4\eta/b^2 + 1) = O(\log n)$. Hence, we can apply Lemma 4.8 to obtain

$$T_{\mathrm{mix}}(\rho, \varepsilon/2) \leq \left\lceil C_\rho \frac{M}{\rho} \left(\log\log \frac{1}{\mu_{\min}} + \log \frac{2}{\varepsilon^2} \right) \right\rceil,$$

where $\rho = \lceil \kappa M \rceil$ and $C_\rho = (2/\kappa)^{4\eta/b^2+1}$. We have

$$C_\rho = \exp\left((\log 2)(k+2)\left(\frac{4\eta}{b^2} + 1\right)\right) \leq 2^{k+2} \exp\left(\frac{(\log 2)(\log n)(k+2)4e^2}{2^{(r_0-\delta)k}} \right),$$

so there exists a function $k_0(\theta) = \Theta(\log(1/\theta))$ such that when $k \geq k_0(\theta)$, we have $C_\rho \leq 2^{k+2} n^\theta$. In light of Remark 4.49, we have $\mu_{\min} \geq b^M$, so $\log\log(1/\mu_{\min}) \leq \log(M \log(1/b)) = \log M$ as $b = 1/e$. Thus, we conclude that

$$T_{\mathrm{mix}}(\rho, \varepsilon/2) \leq \left\lceil 2^{2k+3} n^\theta \left(\log M + \log \frac{2}{\varepsilon^2} \right) \right\rceil \leq \left\lceil 2^{2k+3} n^\theta \log \frac{2n}{\varepsilon^2} \right\rceil. \qquad \square$$

4.7 Proof of Theorem 1.8

In this section we complete the proof of Theorem 1.8. The proofs in this section do not present any challenging steps. In fact, they amount to combining the main technical results that have already been proved in this work. We start by showing that the calls to the method Sample in Algorithm 1 are unlikely to ever return error, that is, the connected components of G_{Φ^Λ} have size at most $2k^4(1+\xi)\log(n)$ almost every time the method is called. As pointed out in our proof outline, this is a straightforward consequence of Lemma 4.12 and the fact that the probability distribution of the output of the Glauber dynamics is $(1/k)$-uniform (Corollary 4.24).

Lemma 4.50. *Let $\theta \in (0,1)$. There is an integer $k_0 \geq 3$ such that, for any integers $k \geq k_0$, $\xi \geq 1$ and any density $\alpha \leq 2^{(r_0-3\delta)k}/k^3$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. In the execution of Algorithm 1 with input Φ, with probability at least $1 - n^{-3\xi}$ over the random choices made by Algorithm 1, every time that the algorithm calls the method* Sample(Φ^Λ, S)*, the connected components of G_{Φ^Λ} have size at most $2k^4(1+\xi)\log(n)$.*

Proof. Let $\varepsilon = n^{-\xi}$ and let $T = \lceil 2^{2k+3} n^\theta \log \frac{2n}{\varepsilon^2} \rceil$ be the mixing time established in Lemma 4.10. Note that $\alpha \leq 2^{(r_0-3\delta)k}/k^3 \leq 2^{(r_0-\delta)k}/k^3$, so we an indeed compute the marking $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{a}}, \mathcal{V}_{\mathrm{c}})$ in Algorithm 1 with the help of Lemma 4.21. We need $\alpha \leq 2^{(r_0-3\delta)k}/k^3$ so that we can apply Lemma 4.12 with $r = r_0 - \delta$. Algorithm 1 calls the method Sample exactly $T+1$ times in total: T times in line 7 (when simulating the ρ-uniform-block Glauber dynamics) and one time in line 10 to extend the assignment X_T of marked variables to all variables.

Let $t \in \{0, 1, \ldots, T\}$ and let π_t be the probability distribution of X_t, where X_t is the state of the ρ-block-uniform Glauber dynamics on the marked variables described in Algorithm 1 after t steps. Recall that $\rho = \lceil 2^{-k-1}|\mathcal{V}_{\mathrm{m}}| \rceil$ and that X_0 is chosen uniformly at random. First, we focus on the case $t < T$. We are going to apply Lemma 4.12 with $r = r_0 - \delta$, $a = 2k^4$,

$b = 2a(1+\xi)$, $V = \mathcal{V}_{\mathrm{m}}$, $\mu = \pi_t$ and this choice of ρ. The set $\mathcal{V}_{\mathrm{m}}$ is r_0-distributed by the definition of $(r_0 - \delta, r_0, r_0, 2r_0)$-marking (Definition 4.3). Moreover, π_t is $(1/k)$-uniform by Corollary 4.24, and we have $\rho \leq |\mathcal{V}_{\mathrm{m}}|/2^k$. Hence, we can indeed apply Lemma 4.12. Consider the following experiment described in Lemma 4.12 for $L = \lceil a(1+\xi)\log n \rceil$, which satisfies $a \log n \leq L \leq b \log n$. First, draw $S \subseteq \mathcal{V}_{\mathrm{m}}$ from the uniform distribution τ over subsets of $\mathcal{V}_{\mathrm{m}}$ with size ρ. Then, sample an assignment Λ_{t+1} from $\pi_t|_{\mathcal{V}_{\mathrm{m}} \setminus S}$, the marginal of π_t on $\mathcal{V}_{\mathrm{m}} \setminus S$. Denote by $\mathcal{F}$ the event that that there is a connected set of clauses Y of Φ with $|Y| \geq L$ such that all clauses in Y are unsatisfied by Λ_{t+1}. Then we have

$$\Pr_{S \sim \tau}\left(\Pr_{\Lambda_{t+1} \sim \pi_t|_{\mathcal{V}_{\mathrm{m}} \setminus S}}(\mathcal{F}) \leq 2^{-\delta k L}\right) \geq 1 - 2^{-\delta k L}. \tag{4.36}$$

Alternatively, this experiment is the same as first sampling an assignment X_t of all variables in $\mathcal{V}_{\mathrm{m}}$ from π_t, and then restricting the assignment to a random set $S \sim \tau$, obtaining Λ_{t+1}. Note that this exact experiment occurs before calling the method Sample in the t-th step of the ρ-uniform-block Glauber dynamics in Algorithm 1. Thus, in light of (4.36), the probability that in step $t+1$ of the execution of Algorithm 1 the graph $G_{\Phi^{\Lambda_{t+1}}}$ has a connected component with size at least L is at most $2^{-\delta k L} + 2^{-\delta k L}$, where the first $2^{-\delta k L}$ comes from the probability of choosing a wrong set $S \sim \tau$ in (4.36) and the second $2^{-\delta k L}$ comes from the bound on the probability of the event $\mathcal{F}$ once we have chosen S. We have shown that with probability at least $1 - 2{\cdot}2^{-\delta k L}$, all the connected components of the graph $G_{\Phi^{\Lambda_t}}$ appearing in step $t+1$ of the execution of Algorithm 1 have size less than L. We have $2{\cdot}2^{-\delta k L} \leq 2{\cdot}n^{-\delta k a(1+\xi)\log 2} \leq n^{-5\xi}$ for large enough k, so the probability that Sample returns error at step $t+1$ is at most $n^{-5\xi}$. The case $t = T$ is analogous, the only difference here is that we call Sample on Φ^{X_T}, where $X_T \sim \pi_T$ is an assignment of all marked variables, so we apply Lemma 4.12 with $\rho = 0$ instead of $\rho = \lceil 2^{-k-1}|\mathcal{V}_{\mathrm{m}}| \rceil$.

Finally, we carry out a union bound over $t \in \{0, 1, \ldots, T\}$, so the probability that any of the calls to Sample returns error is at most $(T+1)n^{-5\xi} \leq n^{-3\xi}$ for large enough n as $T = O(n^\theta \log n) = O(n \log n)$. □

Once we have established Lemmas 4.10, 4.14, and 4.50, the proof of Theorem 1.8 follows as below.

Theorem 1.8. *For any real $\theta \in (0,1)$, there is $k_0 \geq 3$ with $k_0 = O(\log(1/\theta))$ such that, for any integers $k \geq k_0$ and $\xi \geq 1$, and for any positive real $\alpha \leq 2^{0.039k}$, the following holds.*

There is an efficient algorithm to sample from the satisfying assignments of a random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$ within $n^{-\xi}$ total variation distance of the uniform distribution. The algorithm runs in time $O(n^{1+\theta})$, and succeeds w.h.p. over the choice of Φ.

Proof. Let $k_0(\theta) = \Theta(\log(1/\theta))$ be large enough so that, for all integers $k \geq k_0(\theta)$, $\xi \geq 1$ and all densities $\alpha \leq 2^{0.039 \cdot k}$, the conclusions of Lemmas 4.21, 4.10, 4.14, and 4.50 hold w.h.p. over the choice of the random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. These lemmas are enough to analyse Algorithm 1 and tackle this proof. We analyse the distribution μ_{alg} of the output of Algorithm 1.

This distribution outputs either a satisfying assignment of the input formula Φ or *error*. Let $\varepsilon = n^{-\xi}$. Let $\mathcal{E}$ be the event that running Algorithm 1 outputs *error*. This happens with probability at most $\varepsilon/4$ when computing the marking $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{a}}, \mathcal{V}_{\mathrm{c}})$ in line 2 of the algorithm, and in lines 7 and 10 if the method Sample$(\hat{\Phi}, S)$ returns error, which occurs when $G_{\hat{\Phi}}$ has a connected component with size more than $2k^4(1+\xi)\log(n)$. In view of Lemma 4.50, the probability that Algorithm 1 outputs *error* due to the failure of the method Sample is at most $n^{-3\xi} = \varepsilon^3$. We conclude that the probability that the algorithm outputs error is bounded above by $\varepsilon/4+\varepsilon^3 \leq \varepsilon/2$. Let $\mu_{Glauber}$ be the distribution that Algorithm 1 would output if there were no errors (that is, the distribution assuming that the method Sample always outputs from the appropriate distribution). Then $d_{\mathrm{TV}}(\mu_{alg}, \mu_{Glauber})$ is the probability that an error occurs, which is at most $\varepsilon/2$. Let $\pi_{Glauber}$ be the distribution output by the ρ-uniform-block Glauber dynamics on $\mathcal{V}_{\mathrm{m}}$ after T steps. By Lemma 4.10 on the mixing time of the Glauber dynamics, we have $d_{\mathrm{TV}}(\pi_{Glauber}, \mu_\Omega|_{\mathcal{V}_{\mathrm{m}}}) \leq \varepsilon/2$. As $\mu_{Glauber}$ comes from sampling an assignment X_T from $\pi_{Glauber}$ and then completing X_T to all $\mathcal{V}$ by sampling from $\mu_\Omega(\cdot|X_T)$, we have $d_{\mathrm{TV}}(\mu_{Glauber}, \mu_\Omega) \leq d_{\mathrm{TV}}(\pi_{Glauber}, \mu_\Omega|_{\mathcal{V}_{\mathrm{m}}}) \leq \varepsilon/2$. We find that $d_{\mathrm{TV}}(\mu_{alg}, \mu_\Omega) \leq d_{\mathrm{TV}}(\mu_{alg}, \mu_{Glauber}) + d_{\mathrm{TV}}(\mu_{Glauber}, \mu_\Omega) \leq \varepsilon/2 + \varepsilon/2 = \varepsilon$ as we wanted. The running time of Algorithm 1 is now easily obtained by adding up the running times of the following subroutines. The good clauses and good variables are computed in time $O(n+km) = O(n)$, see Proposition 4.2. The marking $(\mathcal{V}_{\mathrm{m}}, \mathcal{V}_{\mathrm{a}}, \mathcal{V}_{\mathrm{c}})$ is computed with probability at least $1-\varepsilon/4$ in time $O(n\Delta_r k^2 \log(4/\varepsilon)) = O(n\log n)$, see Lemma 4.21. Recall that there are $T+1 = O(n^\theta \log(n/\varepsilon^2)) = O(n^\theta \log n)$ calls to the method Sample(Φ', S), and each call takes time $O(|S|\log n) = O(n\log n)$ by Lemma 4.14. We conclude that the running time of Algorithm 1 is $O(n^{1+\theta}\log(n)^2)$. The result now follows by choosing $k_1 = k_0(\theta/2)$, so the running time for $k \geq k_1$ is $O(n^{1+\theta/2}\log(n)^2) = O(n^{1+\theta})$. □

We have now proved that it is possible to (approximately) sample uniformly at random from the satisfying assignments of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. At this point, standard techniques can be applied to obtain a randomised approximation scheme for counting the satisfying assignments of Φ. These techniques are based on the self-reducibility of k-SAT [77]. The following remark shows how to obtain a randomised approximation scheme that runs in time $O(n^\theta(n/\varepsilon)^2)$ following [43, Chapter 7], where the authors base their counting algorithm on the simulated annealing method [117, 71, 81].

Remark 4.51 (Approximate counting for random k-SAT formulae)**.** *Let $k_0(\theta)$ be the integer depending on $\theta \in (0,1)$ obtained in Theorem 1.8. Let $k_1 = k_0(\theta/2)$, let $k \geq k_1(\theta)$ be an integer, let ξ be a positive integer and let $\alpha \leq 2^{0.039k}$ be a density. We apply Theorem 1.8 to obtain an algorithm to sample from the satisfying assignments of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$ within $n^{-4\xi}$ total variation distance from the uniform distribution. This algorithm runs in time $O(n^{1+\theta/2})$ and succeeds w.h.p. over the choice of Φ.*

Let $\varepsilon \in (0,1)$ with $\varepsilon \geq n^{-\xi}$. A modified version of the approximate counting algorithm of [43, Section 7], using $O(\varepsilon^{-2} n \log(n/\varepsilon))$ samples from the sampling algorithm above, approximates the number of satisfying assignments of the k-CNF formula Φ with multiplicative error ε, thus,

achieving running time $O(n^{\theta/2}(n/\varepsilon)^2 \log(n/\varepsilon)) = O(n^{\theta}(n/\varepsilon)^2)$. Here we describe these minor modifications.

Let Ω_{bad} be the set of assignments $X\colon \mathcal{V} \to \{\mathsf{F},\mathsf{T}\}$ that satisfy the bad clauses of Φ. For $X \in \Omega_{\text{bad}}$, we define $F(X)$ to be the set of good clauses that are not satisfied by X. For $\kappa \in \mathbb{R}$, we define $w_\kappa(X) = \exp(-\kappa|F(X)|)$ and we define the partition function $Z(\kappa) = \sum_{X\in\Omega_{\text{bad}}} w_\kappa(X)$, which was introduced in Section 1.1.3 of this thesis. The simulated annealing algorithm of [43, Section 7] uses $Z(\kappa)$ (with Ω^* from Definition 4.4 in place of Ω_{bad}) to approximate the number of satisfying assignments of Φ. We note that $Z(0) = |\Omega_{\text{bad}}|$, which can be computed in linear time in n using the exact counting algorithm given in Proposition 4.31. Here one has to use the fact that the connected components of $G_{\Phi'}$ for the formula $\Phi' = (\mathcal{V}, \mathcal{C}_{\text{bad}})$ have size at most $2k^4 \log n$, see Lemma 4.64 from Section 4.9 and Lemma 4.27, and the fact that these connected component have tree-excess upper bounded as a function of k (Lemma 4.26). Once one has performed these modifications, the algorithm given in [43, Section 7] applies without any difficulties.

4.8 Proof of Theorems 1.10 and 1.12

In this section we exploit Lemma 4.12 to prove Theorems 1.10 and 1.12 on the connectivity and looseness of the solution space of random k-CNF formulae. We recall that the density threshold in Theorems 1.10 and 1.12 is $\alpha \le 2^{0.227k}$, significantly larger than our algorithmic threshold in Theorem 1.8, which is $\alpha \le 2^{0.039k}$. In order to conclude connectivity for densities up to $2^{0.227k}$, we let $r_1 = 0.227092$ and consider the threshold $\Delta_r = \lceil 2^{rk} \rceil$ for $r = r_1 - \delta$ in the definition of high-degree variables instead of $\Delta_{r_0-\delta} = \lceil 2^{(r_0-\delta)k} \rceil$. In all this section we set $r = r_1 - \delta$, so we omit r in the notation and we write $\mathcal{V}_{\text{good}}$ instead of $\mathcal{V}_{\text{good}}(r)$ in order to improve the reading experience. We work with an $(r, r_1, 0, r_1)$-marking $(\mathcal{V}_{\text{m}}, \emptyset, \mathcal{V}_{\text{c}})$ (the set of auxiliary variables is empty), which we can find w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$ as in Lemma 4.22. Let us briefly recall some of the properties of this marking. First of all, by definition, the set $\mathcal{V}_{\text{m}}$ is r_1-distributed and is a subset of $\mathcal{V}_{good}$. Moreover, the distribution $\mu_\Omega|_{\mathcal{V}_{\text{m}}}$ is $(1/k)$-uniform by Lemma 4.23. In light of Lemma 4.12 for $r = r_1 - \delta$, these properties allow us to show that, when sampling $\Lambda \sim \mu_\Omega|_{\mathcal{V}_{\text{m}}}$, the connected components of Φ^Λ are logarithmic in size with probability $1 - o(1)$ over the choice $\Lambda \sim \mu_\Omega|_{\mathcal{V}_{\text{m}}}$. In fact, this is also the case when $\Lambda \sim \mu_\Omega|_{\mathcal{V}_{\text{m}}\setminus\{v\}}$ for any variable v.

Corollary 4.52. *There is an integer $k_0 \ge 3$ such that, for any integer $k \ge k_0$, any density $\alpha \le \alpha_1 := 2^{(r_1-3\delta)k}$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$.*

Let V be a set of good variables of Φ that is r_1-distributed, let μ be a $(1/k)$-uniform distribution over the assignments $V \to \{\mathsf{F},\mathsf{T}\}$ and let $v \in V$. Then, with probability at least $1 - n^{-k}$ over the choice $\Lambda \sim \mu|_{V\setminus\{v\}}$, the connected components of Φ^Λ have size at most $2k^4 \log n$.

Proof. The result is an application of Lemma 4.12 with $r = r_1 - \delta$, $b = 4k^4$, $\rho = 1$ and $L = \lceil 2k^4 \log n \rceil$. We need large enough k_0 such that $2^{-\delta k L} \le 2^{-\delta 2k^5 \log n} \le n^{-k}$ for all $k \ge k_0$. For these parameters, in the setting of Lemma 4.12, the distribution τ is the uniform distribution

over the variables in V. The experiment in the statement of Lemma 4.12 consists in sampling $v \sim \tau$ and then sampling $\Lambda \sim \mu|_{V\setminus\{v\}}$. Let $\mathcal{F}_v$ be the event, concerning the choice $\Lambda \sim \mu|_{V\setminus\{v\}}$, that there is a connected set of clauses Y of Φ with $|Y| \geq \lceil 2k^4 \log n \rceil$ such that all clauses in Y are unsatisfied by Λ. Then by Lemma 4.12 we have $\Pr_{v\sim\tau}\left(\Pr_{\Lambda\sim\mu|_{V\setminus\{v\}}}(\mathcal{F}_v) \leq 2^{-\delta k L}\right) \geq 1-2^{-\delta k L}$. From $2^{-\delta k L} \leq n^{-k}$, we obtain the bound $\Pr_{v\sim\tau}\left(\Pr_{\Lambda\sim\mu|_{V\setminus\{v\}}}(\mathcal{F}_v) \leq 2^{-\delta k L}\right) \geq 1-n^{-k}$. Since τ is the uniform distribution over the variables in V, for $v \sim \tau$, either the event that $\Pr_{\Lambda\sim\mu|_{V\setminus\{v\}}}(\mathcal{F}_v) \leq 2^{-\delta k L}$ has probability 1 or it has probability at most $1 - 1/|V| \leq 1 - 1/n$. The latter option is not possible due to $\Pr_{v\sim\tau}\left(\Pr_{\Lambda\sim\mu|_{V\setminus\{v\}}}(\mathcal{F}_v) \leq 2^{-\delta k L}\right) \geq 1 - n^{-k}$ and $k \geq 3$. Thus, we conclude that $\Pr_{v\sim\tau}\left(\Pr_{\Lambda\sim\mu|_{V\setminus\{v\}}}(\mathcal{F}_v) \leq 2^{-\delta k L}\right) = 1$, so for any $v \in V$ we have $\Pr_{\Lambda\sim\mu|_{V\setminus\{v\}}}(\mathcal{F}_v) \leq 2^{-\delta k L} \leq n^{-k}$. That is, for any $v \in V$, with probability at least $1 - n^{-k}$ over the choice of $\Lambda \sim \mu|_{V\setminus\{v\}}$ the connected components of Φ^Λ have size at most $L - 1 = \lceil 2k^4 \log n \rceil - 1 < 2k^4 \log n$ as we wanted to prove. □

Our connectivity and looseness results will follow from Corollary 4.52. In Section 4.8.1 we prove Theorem 1.10 and in Section 4.8.2 we prove Theorem 1.12.

4.8.1 Proof of Theorem 1.10

We consider Algorithm 5 that receives two satisfying assignments of a k-CNF formula Φ as the input and constructs a path between them. Before introducing this algorithm, recall that the graph H_Φ is the dependency graph of the variables of Φ introduced in Definition 4.13.

Algorithm 5 Finding a $(\mathrm{poly}(k)\log n)$-path between two satisfying assignments

Input: a k-CNF formula $\Phi = (\mathcal{V}, \mathcal{C})$ with n variables, an $(r, r_1, 0, r_1)$-marking $(\mathcal{V}_\mathrm{m}, \emptyset, \mathcal{V}_\mathrm{c})$ of Φ, and two satisfying assignments σ, σ'.

1: Let $v_1, v_2, \ldots, v_\ell$ be the variables in $\mathcal{V}_\mathrm{m}$.
2: $\zeta_0 \leftarrow \sigma$.
3: **for** $i \in [\ell]$ **do**
4: Find $\zeta_i \in \Omega$ with marked variables specified by $\zeta_i(v_j) = \begin{cases} \sigma'(v_j), & j \leq i; \\ \sigma(v_j), & j > i; \end{cases}$ such that $\|\zeta_i - \zeta_{i-1}\|_1$ is minimised.
5: **end for**
6: $\xi_0 = \zeta_\ell$
7: Let $\tau' = \sigma'|_{\mathcal{V}_\mathrm{m}}$ and suppose that $H_{\Phi^{\tau'}}$ has connected components $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_t$.
8: **for** $i \in [t]$ **do**
9: Let $\xi_i \in \Omega$ be defined as $\xi_i(v) = \begin{cases} \sigma'(v), & v \in \left(\mathcal{V} \setminus \bigcup_{j=1}^t \mathcal{E}_j\right) \cup \left(\bigcup_{j=1}^i \mathcal{E}_j\right); \\ \zeta_\ell(v), & v \in \bigcup_{j=i+1}^t \mathcal{E}_j. \end{cases}$
10: **end for**
11: **return** The path $\sigma = \zeta_0 \leftrightarrow \cdots \leftrightarrow \zeta_\ell = \xi_0 \leftrightarrow \cdots \leftrightarrow \xi_r = \sigma'$.

To prove Theorem 1.10, it suffices to show that the output of Algorithm 5 is with high probability a D-path in the solution space for $D = 2k^5 \log n$ for the inputs $\sigma \sim \mu_\Omega$ and $\sigma' \sim \mu_\Omega$. We will not actually require $\sigma \sim \mu_\Omega$ and $\sigma' \sim \mu_\Omega$ in the proof; instead we will just use the fact that the restrictions of σ and σ' on $\mathcal{V}_\mathrm{m}$ follow a $(1/k)$-uniform distribution as guaranteed by Lemma 4.23, see the proof of Lemma 4.54 for details.

We need the following two lemmas to establish Theorem 1.10. The first lemma (Lemma 4.53) shows that all the truth assignments ζ_i, ξ_i in the algorithm exist and satisfy the formula (i.e. the algorithm is well-defined), implying our constructed path is indeed a valid path comprising only satisfying assignments. The second lemma (Lemma 4.54) shows that w.h.p., two adjacent assignments differ by at most $2k^5 \log n$ variables. This result is an application of Corollary 4.52.

Lemma 4.53. *For any k-CNF formula Φ with n variables, any $(r, r_1, 0, r_1)$-marking $(\mathcal{V}_\mathrm{m}, \emptyset, \mathcal{V}_\mathrm{c})$ of Φ, and any two satisfying assignments σ, σ', Algorithm 5 on these inputs is well-defined in the following sense:*

1. *It is always possible to implement Line 4 such that $\zeta_i \in \Omega$.*
2. *We have $\xi_i \in \Omega$ for each $i \in [t]$.*

Proof. To prove item 1, we are going to show that for any partial assignment $X \colon \mathcal{V}_\mathrm{m} \to \{\mathsf{F}, \mathsf{T}\}$, we have $\Pr_{\mu_\Omega}(X) > 0$ and, thus, can extend X to some satisfying assignment. If this claim holds, then we can indeed compute the satisfying assignments $\zeta_1, \zeta_2, \ldots, \zeta_\ell$ in Algorithm 5. Recall that the distribution $\mu_\Omega|_{\mathcal{V}_\mathrm{m}}$ is $(1/k)$-uniform, see Lemma 4.23. From the definition of $(1/k)$-uniform distribution, we find that an analogous statement to Proposition 4.5 holds for our $(r, r_1, 0, r_1)$-marking (here $r = r_1 - \delta$): for any $v \in \mathcal{V}_\mathrm{good}(r)$, any $V \subseteq \mathcal{V}_\mathrm{m}$ with $v \notin V$, and any $\Lambda \colon V \to \{\mathsf{F}, \mathsf{T}\}$, we have

$$\max\left\{\Pr_{\mu_{\Omega^\Lambda}}\left(v \mapsto \mathsf{F} \middle| \Lambda\right), \Pr_{\mu_\Omega}\left(v \mapsto \mathsf{T} \middle| \Lambda\right)\right\} \leq \frac{1}{2}\exp\left(\frac{1}{k}\right).$$

Thus, by induction on the size of a set $S \subseteq \mathcal{V}_\mathrm{m}$, we conclude that any assignment $\Lambda \colon S \to \{\mathsf{F}, \mathsf{T}\}$ has $\Pr_{\mu_\Omega}(\Lambda) > 0$, finishing the proof of item 1.

Next consider item 2. Let $\tau' = \sigma'|_{\mathcal{V}_\mathrm{m}}$ as in Algorithm 5. All clauses that do not appear in $G_{\Phi^{\tau'}}$ are satisfied by the partial assignment τ'. Now consider two satisfying assignments Λ, Λ' such that $\Lambda(\mathcal{V}_\mathrm{m}) = \Lambda'(\mathcal{V}_\mathrm{m}) = \tau'$. Let $G_{\Phi^{\tau'}}$ have connected components $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{t'}$. In particular, $\Lambda|_{\mathrm{var}(\mathcal{C}_i)}$ and $\Lambda'|_{\mathrm{var}(\mathcal{C}_i)}$ each satisfy all clauses in $\mathcal{C}_i$. Each clause in $\Phi^{\tau'}$ is in exactly one connected component $\mathcal{C}_i$. Consequently, any assignment X such that $X|_{\mathcal{V}_\mathrm{m}} = \tau'$ and $X|_{\mathrm{var}(\mathcal{C}_i)} \in \{\Lambda|_{\mathrm{var}(\mathcal{C}_i)}, \Lambda'|_{\mathrm{var}(\mathcal{C}_i)}\}$ for all $i \in [t']$ is a satisfying assignment (any variables that do not appear in $\mathcal{V}_\mathrm{m} \cup \left(\bigcup_{i=1}^{t'} \mathrm{var}(\mathcal{C}_i)\right)$ can be chosen arbitrarily). We note that there are two types of connected components of H_{Φ^τ}. The first type are those corresponding to $\mathrm{var}(\mathcal{C}_i)$ for some $i \in [t']$. The second type are those connected components with variables in $\mathcal{V} \setminus \left(\mathcal{V}_\mathrm{m} \cup \left(\bigcup_{i=1}^{t'} \mathrm{var}(\mathcal{C}_i)\right)\right)$. These connected components are singleton and consist of one variable v that does not appear in Φ^τ or, equivalently, every clause of Φ containing v is satisfied by τ. As a consequence, taking

$\Lambda = \zeta_\ell$, $\Lambda' = \sigma'$ and $X = \xi_i$ in the argument above, we conclude that $\xi_0, \xi_1, \ldots, \xi_t$ are satisfying assignments by construction in Algorithm 5 and item 2 holds. □

Lemma 4.54. *There is an integer $k_0 \geq 3$ such that, for any integer $k \geq k_0$, any density $\alpha \leq 2^{(r_1 - 3\delta)k}$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. In Algorithm 5 with inputs the formula Φ, an $(r, r_1, 0, r_1)$-marking of Φ and the two satisfying assignments σ and σ', with probability at least $1 - 1/n$ over the choices $\sigma \sim \mu_\Omega, \sigma' \sim \mu_\Omega$, we have*

1. $\|\zeta_i - \zeta_{i-1}\|_1 \leq 2k^5 \log n$ *for all* $i \in [\ell]$;
2. $\|\xi_i - \xi_{i-1}\|_1 \leq 2k^5 \log n$ *for all* $i \in [t]$.

Proof. Let Φ and $(\mathcal{V}_\mathrm{m}, \emptyset, \mathcal{V}_\mathrm{a})$ be the first two inputs of Algorithm 5, and let $v_1, v_2, \ldots, v_\ell$ be the variables in $\mathcal{V}_\mathrm{m}$ in the order considered in Algorithm 5. Let $\sigma \sim \mu_\Omega$ and $\sigma' \sim \mu_\Omega$. Let $\sigma = \zeta_0 \leftrightarrow \cdots \leftrightarrow \zeta_\ell = \xi_0 \leftrightarrow \cdots \leftrightarrow \xi_r = \sigma'$ be the path between σ and σ' output by Algorithm 5. In light of Lemma 4.53, the assignments $\zeta_0, \zeta_1, \ldots, \zeta_\ell, \xi_1, \ldots, \xi_r$ are satisfying assignments of Φ. We also note that the set of marked variables $\mathcal{V}_\mathrm{m}$ is r_1-distributed and does not contain bad variables by Definition 4.3. We are going to apply Corollary 4.52 with $V = \mathcal{V}_\mathrm{m}$ several times in this proof. In view of Lemma 4.23, the distribution $\mu_\Omega|_{\mathcal{V}_\mathrm{m}}$ is $(1/k)$-uniform, and this will be relevant when applying Corollary 4.52. We prove that Item 1 holds with probability at least $1 - 1/(2n)$ and that Item 2 holds with probability $1 - 1/(2n)$, so the result follows from a union bound.

Item 1. Let $i \in [\ell]$ and let τ_i be the restriction of ζ_i to $\mathcal{V}_\mathrm{m}$. By construction, τ_i agrees with σ' on $v_1, v_2, \ldots, v_i$ and it agrees with σ on $v_{i+1}, v_{i+2}, \ldots, v_\ell$. Let Λ_i denote the restriction of τ_i on $\mathcal{V}_\mathrm{m} \setminus \{v_i\}$, which agrees with ζ_i and ζ_{i-1} on $\mathcal{V}_\mathrm{m} \setminus \{v_i\}$. Recall that, by definition, ζ_i is the satisfying assignment that extends τ_i that minimises $\|\zeta_i - \zeta_{i-1}\|_1$, see Algorithm 5. We consider the connected components of $G_{\Phi^{\Lambda_i}}$, which can be seen as CNF formulae with variables in $\mathcal{V}_\mathrm{c} \cup \{v_i\}$ due to the fact that all marked variables other than v_i are set by Λ_i. Each one of these connected components are satisfied as CNF formulae by the assignments ζ_i and ζ_{i-1}. We conclude that ζ_i and ζ_{i-1} agree on the variables of all these connected components except for those variables in the connected component of the clauses containing v_i, where ζ_i and ζ_{i-1} may disagree. Let us denote this connected component by C_{v_i}, which is empty when all the clauses containing v_i are satisfied by Λ_i. We have $\|\zeta_i - \zeta_{i-1}\|_1 \leq k|C_{v_i}|$, where the factor k comes from the fact that each clause has at most k variables. We now bound the size of C_{v_i}. Since the restrictions of σ and σ' to $\mathcal{V}_\mathrm{m}$ follow $\mu_\Omega|_{\mathcal{V}_\mathrm{m}}$, which is $(1/k)$-uniform, we find, by Definition 4.7, that τ_i also follows an $(1/k)$-uniform distribution over the assignments $\mathcal{V}_\mathrm{m} \to \{\mathsf{F}, \mathsf{T}\}$. Let us denote this distribution by μ_i. Then $\Lambda_i \sim \mu_i|_{\mathcal{V}_\mathrm{m} \setminus \{v_i\}}$ and, by Corollary 4.52 with $V = \mathcal{V}_\mathrm{m}$, $\Lambda = \Lambda_i$ and $\mu = \mu_i$, with probability at least $1 - n^{-k}$ over the choice $\Lambda_i \sim \mu_i|_{\mathcal{V}_\mathrm{m} \setminus \{v_i\}}$, the connected component $C_{v_i} \subset G_{\Phi^{\Lambda_i}}$ containing v_i has at most $2k^4 \log n$ clauses. Thus, with probability at least $1 - n^{-k}$, we have $\|\zeta_i - \zeta_{i-1}\|_1 \leq k|C_{v_i}| \leq 2k^5 \log n$. By a union bound over $i \in [\ell]$ and the fact that $k \geq 3$ and $\ell \leq n$, we conclude that, with probability at least $1 - 1/n^2$, we have $\|\zeta_i - \zeta_{i-1}\|_1 \leq 2k^5 \log n$ for all $i \in [\ell]$.

Item 2. Let $\tau' = \sigma'|_{\mathcal{V}_\mathrm{m}}$ as in Algorithm 5. By construction, $\xi_0 = \zeta_\ell$ and $\xi_t = \sigma'$ agree with τ' on $\mathcal{V}_\mathrm{m}$. Since $\sigma' \sim \mu_\Omega$, we have $\tau' \sim \mu_\Omega|_{\mathcal{V}_\mathrm{m}}$, which is $(1/k)$-uniform by Lemma 4.23. In view of Corollary 4.52 for $V = \mathcal{V}_\mathrm{m}$, $\Lambda = \tau'$ and $\mu = \mu_\Omega|_{\mathcal{V}_\mathrm{m}}$, with probability at least $1 - n^{-k}$, all of the connected components of $G_{\Phi^{\tau'}}$, have size at most $2k^4 \log n$. Thus, all the connected components of $H_{\Phi^{\tau'}}$ have size at most $2k^5 \log n$. By construction, see Line 9 in Algorithm 5, the assignments ξ_{i-1} and ξ_i agree on the variables in all the connected components of $H_{\Phi^{\tau'}}$ except for the variables in the i-th connected component, where they may disagree. Thus, they disagree on at most $2k^5 \log n$ variables. This gives the desired result. □

We can now complete the proof of Theorem 1.10.

Theorem 1.10. *There is $k_0 \geq 3$ and a polynomial $p(k)$ with non-negative integer coefficients such that, for any integer $k \geq k_0$, and for any positive real $\alpha \leq 2^{0.227k}$, the following claim holds with high probability over the choice of a random k-CNF formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. Two satisfying assignments chosen uniformly at random are $p(k)\log(n)$-connected with probability at least $1 - 1/n$.*

Proof. Since $\alpha \leq 2^{0.227k} \leq 2^{(r_1 - 3\delta)k}/k^3 \leq 2^{(r_1-\delta)k}/k^3$ for large enough k, w.h.p. over the choice of Φ, there is an $(r, r_1, 0, r_1)$-marking $(\mathcal{V}_\mathrm{m}, \emptyset, \mathcal{V}_\mathrm{c})$ of Φ, see Lemma 4.22. We run Algorithm 5 with inputs Φ, and the associated marking $(\mathcal{V}_\mathrm{m}, \emptyset, \mathcal{V}_\mathrm{c})$. W.h.p. over the choice of Φ, Lemma 4.54 holds. Therefore, with probability at least $1 - 1/n$ over the choice of two random satisfying assignments $\sigma \sim \mu_\Omega$ and $\sigma' \sim \mu_\Omega$, the output path of Algorithm 5 is well-defined by Lemma 4.53 and satisfies that $\|\zeta_i - \zeta_{i-1}\|_1 \leq 2k^5 \log n$ for all $i \in [\ell]$ and $\|\xi_i - \xi_{i-1}\|_1 \leq 2k^5 \log n$ for all $i \in [t]$ by Lemma 4.54. Hence, it is a D-path in the solution space Ω for $D = 2k^5 \log n$ as we wanted. □

4.8.2 Proof of Theorem 1.12

We next show $O(\log n)$-looseness for all variables with high probability over (Φ, σ) for random k-CNF instances Φ and uniformly random satisfying assignment $\sigma \in \Omega$. Consequently, in an *algorithmic* regime where $\alpha \ll 2^{ck}$ for some $c < 1$, we resolve a conjecture of [1]. Our proof exploits Corollary 4.52 on the size of the connected components of Φ^Λ. It is important in our arguments that every variable in the formula is *flippable*.

Definition 4.55. *Let $\Phi = \Phi(k, n, m)$ be a random k-CNF. A variable $v \in \mathcal{V}$ is flippable if there exists a pair of satisfying assignments (X, Y) to Φ, in one of which $X(v) = \mathsf{F}$ and in the other $Y(v) = \mathsf{T}$.*

Lemma 4.56. *For $\alpha < 2^{k-2}$, with high probability over the choice of $\Phi = \Phi(k, m, n)$, all variables in Φ are flippable.*

Proof. Observe that we can define an NAE-SAT problem based on Φ. By definition, any NAE-satisfying assignment ensures that every clause contains at least one satisfied literal and at

least one unsatisfied literal. By Theorem 2 in [4], with high probability Φ is NAE-satisfiable. Consequently, we can find some assignment σ that NAE-satisfies Φ with high probability, and then the opposite assignment $\overline{\sigma}$ also NAE-satisfies Φ by the symmetry of NAE-SAT solutions. In particular, both σ and $\overline{\sigma}$ are solutions to the original SAT formula Φ. Observe that for every variable $v \in V$ we have $X(v) = \mathsf{T}$ and $X(v) = \mathsf{F}$ in exactly one of $\sigma, \overline{\sigma}$ and thus, with high probability, every variable in Φ is flippable. □

Lemma 4.57. *For any variable $v \in \mathcal{V}$ and any partial assignment $X \colon \mathcal{V}_{\mathrm{m}} \setminus \{v\} \to \{\mathsf{F}, \mathsf{T}\}$, we have*

$$\Pr_{\mu_\Omega}(v \mapsto \mathsf{F} | X) > 0 \quad \textit{and} \quad \Pr_{\mu_\Omega}(v \mapsto \mathsf{T} | X) > 0.$$

Proof. We prove $\Pr_{\mu_\Omega}(v \to \mathsf{F}|X) > 0$; the proof of $\Pr_{\mu_\Omega}(v \to \mathsf{T}|X) > 0$ is analogous. We distinguish two cases.

The first case is when v is a good variable. Lemma 4.23 gives $\Pr_{\mu_\Omega}(v \mapsto \mathsf{F}|X, \Lambda_{\mathrm{bad}}) \geq 1 - \exp(1/k)/2 > 0$ for any satisfying assignment of the bad clauses Λ_{bad}. Thus, we have $\Pr_{\mu_\Omega}(v \mapsto \mathsf{F}|X) > 0$.

The second case is when v is a bad variable. By Lemma 4.56 there exists a satisfying assignment σ with $\sigma(v) = \mathsf{F}$. Let $\Lambda_{\mathrm{bad}} = \sigma|_{\mathcal{V}_{\mathrm{bad}}}$ be the assignment on bad variables and so in particular $\Pr_{\mu_\Omega}(\Lambda_{\mathrm{bad}}) > 0$. Then by Lemma 4.23 we have $\Pr_{\mu_\Omega}(X|\Lambda_{\mathrm{bad}}) \geq (1-\exp(1/k)/2)^{|\mathcal{V}_{\mathrm{m}}|} > 0$. This implies that $\Pr_{\mu_\Omega}(X, \Lambda_{\mathrm{bad}}) > 0$ and in particular $\Pr_{\mu_\Omega}(v \mapsto \mathsf{F}, X) > 0$, so $\Pr_{\mu_\Omega}(v \mapsto \mathsf{F}|X) > 0$. □

We can now prove Theorem 1.12 with the help of Corollary 4.52.

Theorem 1.12. *There is $k_0 \geq 3$ such that, for any integer $k \geq k_0$, and for any positive real $\alpha \leq 2^{0.227k}$, the random k-CNF formula $\Phi(k, n, \lfloor \alpha n \rfloor)$ is $\mathrm{poly}(k)\log(n)$-loose.*

Proof. Note that $2^{0.227k} \leq 2^{(r_1 - 3\delta)k} \leq 2^{(r_1-\delta)k}/k^3$ for large enough k. Thus, w.h.p. over the choice of Φ, there is an $(r, r_1, \emptyset, r_1)$-marking $(\mathcal{V}_{\mathrm{m}}, \emptyset, \mathcal{V}_{\mathrm{c}})$ of Φ, see Lemma 4.21. The distribution $\mu_\Omega|_{\mathcal{V}_{\mathrm{m}}}$ is $(1/k)$-uniform by Lemma 4.23. Hence, Corollary 4.52 holds for $V = \mathcal{V}_{\mathrm{m}}$ and $\mu = \mu_\Omega|_{\mathcal{V}_{\mathrm{m}}}$. Let v be a variable of Φ. Let $\sigma \sim \mu_\Omega$ and let Λ be the restriction of σ to $\mathcal{V}_{\mathrm{m}} \setminus \{v\}$. Then, with probability at least $1 - n^{-k}$, the connected components of G_{Φ^Λ} have size at most $2k^4 \log n$. Let $\mathcal{C}_j^\Lambda$ be the connected component containing the variable v, which is empty if all clauses containing v are satisfied. Let ω be the negation of $\sigma(v)$. By Lemma 4.57, we have $\Pr_{\mu_\Omega}(v \mapsto \omega|\Lambda) > 0$. Therefore, there is an assignment Y of the variables in $\mathrm{var}(\mathcal{C}_j^\Lambda)$ that satisfies the clauses in $\mathcal{C}_j^\Lambda$ and has $Y(v) = \omega$. We construct the assignment σ' that has $\sigma'(v) = \omega$, agrees with Y in $\mathrm{var}(\mathcal{C}_j^\Lambda)$ and agrees with σ in the rest of the variables of Φ. In particular, this assignment agrees with Λ and satisfies each one of the connected components of Φ^Λ. Thus, σ' is a satisfying assignment of Φ. Moreover, w.h.p. σ' differs with σ in at most $2k^5 \log n$ variables (the variables in $\mathrm{var}(\mathcal{C}_j^\Lambda)$). We have shown that, w.h.p. over the choice of Φ, with probability at least $1 - n^{-k}$ a random assignment $\sigma \sim \mu_\Omega$ is $(2k^5 \log n)$-loose, so the statement follows. □

4.9 Proofs of Lemmas 4.15 and 4.16

In this section we prove Lemmas 4.15 and 4.16. The proofs of these results are independent of the rest of this chapter and, in fact, follow from slightly modifying some results in [49], without involving any other material. We include the proofs here for completeness.

Recall that Lemmas 4.15 is [49, Lemma 8.16] with a less restrictive bound on the density of the formula and a more restrictive definition of good variables/clauses, see Section 4.2 for details. Moreover, the obtained upper bound on the number of bad clauses in our version of [49, Lemma 8.16] is tighter. The original proof of Lemma 4.15 given in [49, Section 8] is split into a sequence of results on random formulae. Here we restate some of these results — only those whose statement needs to change as a consequence of our definition of good variables/clauses and the tighter upper bound. We also explain how these changes affect the proofs if any modifications are necessary.

Let us fix some notation first. The results stated in this section only hold for large enough k unless we say otherwise. We note that in [49] the density α is at most $2^{k/300}/k^3$ and $\Delta = 2^{k/300}$, where Δ is the threshold in the definition of high-degree variables, and the proofs are carried out for these particular values. It turns out that, following the proofs in [49, Section 8], the only properties of α and Δ needed in order to proof Lemma 4.15 are that, for $r \in (0, 1/(2\log 2))$, we have $\Delta_r = \lceil 2^{rk} \rceil$ and α is bounded above by Δ_r/k^3 (note the subscript r here to indicate that Δ_r depends on r). First, we need some definitions. For any set of variables $S \subseteq \mathcal{V}$ of Φ, we denote by $\mathrm{HD}(S, r)$ the set of high-degree variables in S (recall that a variable is of high-degree if the degree of v is at least Δ_r).

Lemma 4.58 ([49, Lemma 8.1])**.** *Let $r \in (0, 1)$. There is a positive integer k_0 such that for any integer $k \geq k_0$, $\Delta_r = \lceil 2^{rk} \rceil$, and any density α with $\alpha \leq \Delta_r/k^3$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. The size of $\mathcal{V}_0(r) := \mathrm{HD}(\mathcal{V}, r)$ is at most $(\alpha/\Delta_r)n/2^{k^{10}}$.*

Proof. The proof is the same to that of [49, Lemma 8.1], apart from one change that we highlight here. The degrees of the variables in Φ have the same distribution as a balls-and-bins experiment with km balls and n bins. Let $D_1, D_2, \ldots, D_n$ be independent variables following the Poisson distribution $\mathrm{Poi}(\mu)$ with parameter $\mu = k\alpha$. The degrees of the variables of Φ have the same distribution as $\{D_1, D_2, \ldots, D_n\}$ conditioned on the event $\mathcal{E}$ that $D_1 + D_2 + \cdots + D_n = m$, see for instance [92, Chapter 5.4]. Let $U = \{i \in [n] : D_i \geq \Delta_r\}$. We want to show that $\Pr(|U| > (\alpha/\Delta_r)n/2^{k^{10}} | \mathcal{E}) = o(1)$. In [49, Lemma 8.1] the authors show that $\Pr(|U| > n/2^{k^{10}} | \mathcal{E}) = o(1)$. Their bound is not tight, but it is enough for their purposes. In fact, one can change k^{10} by any polynomial and the result would still hold for large enough k. Here we obtain the extra factor α/Δ_r by slightly modifying the application of the tail bound $\Pr(\mathrm{Poi}(\mu) \geq x) \leq e^{-\mu}(e\mu)^x/x^x$. For $x = \Delta_r$, instead of using the bound $e^{-\mu}(e\mu)^x/x^x \leq e^{-\Delta_r} \leq 2^{-k^{10}-1}$, which holds for large enough k as $\mu/x \leq k^{-2}$ and Δ_r is exponential in k, we use the bound $e^{-\mu}(e\mu)^x/x^x \leq (e\mu/x)e^{-x+1} \leq (\alpha/\Delta_r)2^{-k^{10}-1}$. The rest of the proof is analogous; we have $\mathbb{E}[|U|] \geq n(\alpha/\Delta_r)2^{-k^{10}-1}$, so by a Chernoff bound we find that $\Pr(|U| \geq (\alpha/\Delta_r)n/2^{k^{10}}) \leq \exp(-\Omega(n))$. From the connection

between a balls-and-bins experiment and the Poisson distribution, see [92, Theorem 5.7], we conclude that $\Pr(|U| \geq (\alpha/\Delta_r)n/2^{k^{10}}|\mathcal{E}) \leq \exp(-\Omega(n))$ as we wanted. □

Corollary 4.59 ([49, Corollary 8.4])**.** *There is a positive integer k_0 such that for any integer $k \geq k_0$ and any density α with $\alpha \leq 2^k/(ek^3)$ the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. For every set of variables Y such that $2 \leq |Y| \leq n/2^k$, the number of clauses that contain at least 3 variables from Y is at most $|Y|$.*

Proof. This is a consequence of [49, Lemma 35] with $b = 3$ and $t = 2/(b-1) = 1$, whose proof only requires $\alpha \leq 2^k/(ek^3)$. □

Recall that the graph H_Φ is the dependency graph of the variables of Φ, see Definition 4.13.

Lemma 4.60 ([49, Lemma 8.8])**.** *Let $r \in (0, 1)$. There is a positive integer k_0 such that for any integer $k \geq k_0$, $\Delta_r = \lceil 2^{rk} \rceil$, and any density α with $\alpha \leq \Delta_r/k^3$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. Every connected set U of variables in H_Φ with size at least $2k^4 \log n$ satisfies that $|\mathrm{HD}(U, r)| \leq \frac{1}{2k^3}|U|$.*

Proof. The proof is that of [49, Lemma 8.8], with the difference that $\delta_0 = 1/(2k^3)$ instead of $\delta_0 = 1/21600$, as the exact value of δ_0 does not play a role in the proof as long as, for $\theta_0 = \Delta_r - 2(k+1)$, we have $\delta_0\theta_0 \log \frac{\theta_0}{k^2\alpha} \geq 3\log k + \log \alpha$, which holds for large enough k when $\delta_0 = \mathrm{poly}(k)$. Moreover, the only restriction on α is that of Corollary 4.59, and the fact that $\alpha \leq \Delta_r/k^3$. □

Lemma 4.61 ([31, Lemma 2.4] and [49, Lemma 8.10])**.** *Let $k \geq 3$ be an integer and let α be a positive real number with $\alpha \leq e^{k/2}/(2e^2k^2)$. For any $\varepsilon \in [1/n, 1)$ (depending on n) such that $\varepsilon < e^{-3k}$ for all n, the following holds w.h.p. over the choice of the random formula $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. Let Z be a set of clauses with size at most εn and let $c_1, \ldots, c_l \in \mathcal{C} \setminus Z$ be distinct clauses. For $s \in \{1, 2, \ldots, \ell\}$, let $N_s := \mathrm{var}(Z) \cup \bigcup_{j=1}^{s-1} \mathrm{var}(c_j)$. If $|\mathrm{var}(c_s) \cap N_s| \geq 3$ for all $s \in \{1, 2, \ldots, \ell\}$, then $\ell \leq \varepsilon n$.*

Proof. The proof is almost identical to the proof of [31, Lemma 2.4]. There are four differences. First, here, as it is also the case in [49, Lemma 44], ε can depend on n. This will arise later in this proof. Second, the proof of [31, Lemma 2.4] is carried out for the condition $|\mathrm{var}(c_s) \cap N_s| \geq \lambda$, where λ is an integer with $\lambda > 4$. Here we set $\lambda = 3$ and impose stricter hypotheses on α and ε to compensate for a smaller λ. Their (more relaxed) hypotheses on α and ε are $\alpha \leq 2^k \log 2$, $\varepsilon \leq k^{-3}$ and $\varepsilon^\lambda \leq (2e)^{-4k}/e$. Third, we substitute the last inequality of [31, Equation 4], which is

$$\left[\left(\frac{em/n}{\varepsilon}\right)^2 \exp(2k)(2k\varepsilon)^\lambda\right]^{\varepsilon n} \leq \left[(2e)^{2k}\,\varepsilon^{\lambda/2}\right]^{\varepsilon n},$$

by the inequality

$$\begin{aligned}\left[\left(\frac{em/n}{\varepsilon}\right)^2 \exp(2k)(2k\varepsilon)^\lambda\right]^{\varepsilon n} &\leq \left[(em/n)^2 \exp(2k)(2k)^3\varepsilon\right]^{\varepsilon n} \\ &\leq \left[\exp(3k-1)\varepsilon\right]^{\varepsilon n},\end{aligned} \tag{4.37}$$

where we used $\lambda = 3$ and $m/n \leq \alpha \leq e^{k/2}/(2e^2k^2)$. Now, as it is done in [49, Lemma 8.10], we distinguish two cases depending on ε. If $\varepsilon \geq 10(\log n)/n$, then using this in conjunction with $\varepsilon < e^{-3k}$, the right hand size of (4.37) is bounded by $e^{-\varepsilon n} \leq 1/n^{10} = o(1/n)$. If $1/n \leq \varepsilon < 10(\log n)/n$, then, for large enough n, the right hand size of (4.37) is bounded above by $\exp(3k-1)\varepsilon = o(1)$. The last difference between the proofs is that our argument works for all $k \geq 3$, whereas the bound [31, Equation 4] only holds for large k. □

The remaining results in this section do not need any changes in their original proofs, other than that every time Corollary 8.4, Lemma 8.8 and Lemmas 8.10-8.16 are invoked in [49, Section 8], we use the version given in this section instead. We note that the statements of these results are slightly different to their [49, Section 8] versions, and these changes are again due to the fact that we use $\lambda = 3$ instead of $\lambda = k/10$ in the definition of good variables/clauses.

Corollary 4.62 ([49, Corollary 8.11])**.** *Let $r \in (0, 1/(2\log 2)]$. There is a positive integer k_0 such that for any integer $k \geq k_0$, $\Delta_r = \lceil 2^{rk} \rceil$, and any density α with $\alpha \leq \Delta_r/k^3$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. Let Z be a set of clauses with size at most $2n/2^{k^{10}}$ and let $c_1, \ldots, c_l \in \mathcal{C} \setminus Z$ be distinct clauses. For $s \in \{1, 2, \ldots, \ell\}$, let $N_s := \mathrm{var}(Z) \cup \bigcup_{j=1}^{s-1} \mathrm{var}(c_j)$. If $|\mathrm{var}(c_s) \cap N_s| \geq 3$ for all $s \in \{1, 2, \ldots, \ell\}$, then $\ell \leq |Z|$.*

Proof. The proof given in [49, Corollary 8.11] also applies here. We note that the density α is at most $e^{k/2}/(2e^2k^2)$ so we can indeed apply Lemma 4.61 when the proof given in [49, Corollary 8.11] invokes [49, Lemma 8.10]. □

Lemma 4.63 ([49, Lemma 8.13])**.** *Let $r \in (0, 1/(2\log 2)]$. There is a positive integer k_0 such that for any integer $k \geq k_0$, $\Delta_r = \lceil 2^{rk} \rceil$, and any density α with $\alpha \leq \Delta_r/k^3$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. For any bad component S of variables, we have $|S| \leq 2k|\mathrm{HD}(S, r)|$.*

Proof. The proof given in [49, Lemma 8.13] applies using our versions of [49, Lemma 8.1, Corollary 8.4 and Corollary 8.11]. □

Lemma 4.64 ([49, Lemma 8.14])**.** *Let $r \in (0, 1/(2\log 2)]$. There is a positive integer k_0 such that for any integer $k \geq k_0$, $\Delta_r = \lceil 2^{rk} \rceil$, and any density α with $\alpha \leq \Delta_r/k^3$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. Every bad component S has size at most $2k^4 \log n$.*

Proof. The proof given in [49, Lemma 8.14] applies using our versions of [49, Lemma 8.8 and Lemma 8.13]. □

Lemma 4.65 ([49, Lemma 8.15])**.** *Let $r \in (0, 1/(2\log 2)]$. There is a positive integer k_0 such that for any integer $k \geq k_0$, $\Delta_r = \lceil 2^{rk} \rceil$, and any density α with $\alpha \leq \Delta_r/k^3$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. For every connected set of S variables with size at least $2k^4 \log n$, we have $|S \cap \mathcal{V}_{\mathrm{bad}}| \leq |S|/k^2$.*

Proof. The proof is analogous to that given in [49, Lemma 8.15]. The only differences are that we apply Lemma 4.60 instead of [49, Lemma 8.8], we apply Lemma 4.63 instead of [49, Lemma 8.13], and we have $\delta_0 = 1/(2k^3)$ instead of $\delta_0 = 1/21600$. □

Lemma 4.15 ([49, Lemma 8.16])**.** *Let $r \in (0, 1/(2\log 2)]$. There is a positive integer k_0 such that for any integer $k \geq k_0$, $\Delta_r = \lceil 2^{rk} \rceil$, and any density α with $\alpha \leq \Delta_r/k^3$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. For every connected set of clauses Y in G_Φ such that $|\mathrm{var}(Y)| \geq 2k^4 \log n$, we have $|Y \cap \mathcal{C}_{\mathrm{bad}}(r)| \leq |Y|/k$.*

Proof. The same proof applies using our versions of [49, Corollary 8.4 and Lemma 8.15]. □

Lemma 4.16 ([49, Lemma 8.12])**.** *Let $r \in (0, 1/(2\log 2)]$. There is a positive integer k_0 such that for any integer $k \geq k_0$, $\Delta_r = \lceil 2^{rk} \rceil$, and any density α with $\alpha \leq \Delta_r/k^3$, the following holds w.h.p. over the choice of $\Phi = \Phi(k, n, \lfloor \alpha n \rfloor)$. We have $|\mathcal{C}_{\mathrm{bad}}(r)| \leq 2(\alpha/\Delta_r)n/2^{k^{10}}$ and $|\mathcal{V}_{\mathrm{bad}}(r)| \leq 2(k+1)(\alpha/\Delta_r)n/2^{k^{10}}$.*

Proof. We consider the set of high-degree variables $\mathcal{V}_0(r) = \mathrm{HD}(\mathcal{V}, r)$, which w.h.p. over the choice of Φ has $|\mathcal{V}_0(r)| \leq (\alpha/\Delta_r)n/2^{k^{10}}$ by Lemma 4.58. In view of Corollary 4.59 with $Y = \mathcal{V}_0(r)$, we have $|\mathcal{C}_0(r)| \leq |\mathcal{V}_0(r)| \leq n/2^{k^{10}}$, where $\mathcal{C}_0(r)$ is the set of clauses with at least 3 variables in $\mathcal{V}_0(r)$, see Algorithm 2. From Corollary 4.62 and the construction of $\mathcal{C}_{\mathrm{bad}}(r)$ in Algorithm 2, we find that $|\mathcal{C}_{\mathrm{bad}}(r)| \leq 2|\mathcal{C}_0(r)| \leq 2|\mathcal{V}_0(r)| \leq 2(\alpha/\Delta_r)n/2^{k^{10}}$. By construction of $\mathcal{V}_{\mathrm{bad}}(r)$, see Algorithm 2, we conclude that $|\mathcal{V}_{\mathrm{bad}}(r)| \leq |\mathcal{V}_0(r)| + k|\mathcal{C}_{\mathrm{bad}}(r)| \leq 2(k+1)(\alpha/\Delta_r)n/2^{k^{10}}$. □

4.10 Proof of Lemma 4.8

In this section we collect the results from [29] that one needs to combine to obtain Lemma 4.8 on the mixing time of the ρ-uniform-block Glauber dynamics. The proof is independent from the rest of this chapter and we include it here for completeness.

Definition 4.66. *Let μ be a distribution supported on $\Omega \subseteq [q]^V$. Let $f\colon \Omega \to \mathbb{R}_{\geq 0}$. We denote the entropy of f by $\mathrm{Ent}_\mu(f)$, that is, $\mathrm{Ent}_\mu(f) = \mathbb{E}_\mu(f \log f)) - \mathbb{E}_\mu(f)\log(\mathbb{E}_\mu(f))$ when $\mathbb{E}_\mu(f) > 0$, and $\mathrm{Ent}_\mu(f) = 0$ when $\mathbb{E}_\mu(f) = 0$. For $S \subseteq V$, we denote $\mathrm{Ent}_\mu^S(f) = \mathbb{E}_{\tau \sim \mu|_{V\setminus S}} \mathrm{Ent}_\mu(f|\,\tau)$, where $\mathrm{Ent}_\mu(f|\,\tau)$ is the entropy of f conditioning to the event that the assignment drawn from μ agrees with τ in $V \setminus S$.*

Let $\rho \in \{1, 2, \ldots, n\}$. We say that μ satisfies the ρ-uniform block factorisation of entropy (with constant C_ρ) if for all $f\colon \Omega \to \mathbb{R}_{\geq 0}$ we have

$$\frac{\rho}{n}\mathrm{Ent}_\mu(f) \leq C_\rho \frac{1}{\binom{n}{\rho}} \sum_{S \in \binom{V}{\rho}} \mathrm{Ent}_\mu^S(f).$$

One of the main results of [29] is showing that μ satisfies the ρ-uniform block factorisation of entropy when the distribution μ is η-spectrally independent and b-marginally bounded. In the proof of [17, Corollary 19] the authors observe that the proof of Lemma 4.67 also holds when η depends on n and, in particular, in the case $\eta = \varepsilon \log n$.

Lemma 4.67 ([29, Lemma 2.4])**.** *The following holds for any reals $b, \eta > 0$, any $\kappa \in (0,1)$ and any integer n with $n \geq \frac{2}{\kappa}(4\eta/b^2 + 1)$.*

Let $q \geq 2$ be an integer, let V be a set of size n and let μ be a distribution over $[q]^V$. If μ is b-marginally bounded and η-spectrally independent, then μ satisfies the $\lceil \kappa n \rceil$-uniform block factorisation of entropy with constant $C = (2/\kappa)^{4\eta/b^2+1}$.

It turns out that one can bound the mixing time of the ρ-uniform-block Glauber dynamics when the target distribution μ satisfies the ρ-uniform block factorisation of entropy.

Lemma 4.68 (See, e.g., [29, Lemma 2.6 and Fact 3.5(4)] or [17, Lemma 17])**.** *Let $q \geq 2$, $\rho \geq 1$ be integers and V be a set of size $n \geq \rho + 1$. Let μ be a distribution supported on $\Omega \subseteq [q]^V$ that satisfies the ρ-uniform-block factorisation of entropy with multiplier C_ρ. Then, for any $\varepsilon > 0$, the mixing time of the ρ-uniform-block Glauber dynamics on μ satisfies, for $\mu_{\min} = \min_{\Lambda \in \Omega} \mu(\Lambda)$,*

$$T_{\mathrm{mix}}(\varepsilon) \leq \left\lceil C_\rho \frac{n}{\rho} \left(\log \log \frac{1}{\mu_{\min}} + \log \frac{1}{2\varepsilon^2} \right) \right\rceil.$$

We can now prove Lemma 4.8.

Lemma 4.8. *The following holds for any reals $b, \eta > 0$, any $\kappa \in (0,1)$ and any integer M with $M \geq \frac{2}{\kappa}(4\eta/b^2 + 1)$. Let V be a set of size M, let μ be a distribution over the assignments $V \to \{\mathsf{F}, \mathsf{T}\}$, let $\Omega = \{\Lambda \colon V \to \{\mathsf{F}, \mathsf{T}\} : \mu(\Lambda) > 0\}$ and let $\mu_{\min} = \min_{\Lambda \in \Omega} \mu(\Lambda)$. If μ is b-marginally bounded and η-spectrally independent, then, for $\rho = \lceil \kappa M \rceil$ and $C_\rho = (2/\kappa)^{4\eta/b^2+1}$, we have*

$$T_{\mathrm{mix}}(\rho, \varepsilon) \leq \left\lceil C_\rho \frac{M}{\rho} \left(\log \log \frac{1}{\mu_{\min}} + \log \frac{1}{2\varepsilon^2} \right) \right\rceil.$$

Proof of Lemma 4.8. The proof of Lemma 4.8 follows directly from combining Lemmas 4.67 and 4.68. □

Chapter 5

Conclusion and open questions

This thesis has established several computational complexity results for counting problems arising in statistical mechanics. The research conducted in this thesis demonstrates the interplay between approximate counting and statistical mechanics. This interplay has garnered considerable attention from the research community in the past years, and has lead to several fundamental questions in both fields. As we have illustrated, one of these remarkable questions is that of understanding the computational complexity of sampling from the distribution of spin systems. This problem is intricately linked, via self-reducibily arguments, to approximating the partition function of the model, a question that naturally emerges in the complexity of counting due to its connections to combinatorics.

In this thesis, we have capitalised on recent advancements in approximate counting. These recent breakthroughs have showcased connections between approximate counting and various areas of mathematics, such as complex analysis, complex dynamics, and the revival of Markov Chain Monte Carlo algorithms for efficiently sampling in spin systems. We have delved into these connections to obtain both inapproximability and tractability results, focusing primarily on the Ising and Potts models, as well as the random k-SAT model.

In Chapter 2 we have studied the complexity of approximating the partition function of the q-state Potts model and the closely related Tutte polynomial on planar graphs. Following recent trends in both statistical physics and algorithmic research, we have allowed the edge interaction y to be any complex number. We have established a complete classification of the complexity of approximating the partition function of the Potts model for all non-real values of the parameters (Theorem 1.1), concluding #P-hardness of approximation for almost all parameters. Our techniques apply to all $q \geq 2$ in the Tutte world, and further complement/refine previous results for the real Tutte plane. Moreover, we have answered a question raised by Bordewich, Freedman, Lovász and Welsh in the context of quantum computation (Section 2.6.2). In order to do this, we have introduced the concept of approximate shifts in the complex-plane and shown how we can implement approximations of real edge interactions starting with non-real parameters. Another key development of our work is a reduction from exact evaluation of the Tutte polynomial to approximate computation.

Even though we have fully resolved the complexity map of approximating the partition function of the Potts model, several questions remain regarding more general models. These questions seem to require the development of new techniques, both from the perspective of finding novel approximate shifts and further refining our current reductions. First of all, our results on non-real edge interactions for the Tutte polynomial only hold for $q \geq 2$, and it is interesting

to find new ideas to analyse other values of q. Regarding real parameters, the complexity map of approximating and determining the sign of the Tutte polynomial at real (q, γ) is still not fully understood, including famous points such as $q = 5$, $\gamma = -q$ (recall that $\gamma = y - 1$ in this notation). The evaluation of the Tutte polynomial at $q = 5$ and $\gamma = -q$ counts the number of nowhere-zero 5-flows in a graph (up to an easily computable factor), which is conjectured to be non-zero in the famous Tutte's 5-flow conjecture. Another direction worth studying is the complexity of approximating the graph homomorphism polynomial, which also generalises the Potts model. A complete classification is known for exact evaluation of this partition function, but an approximate counting version of this result seems still out of reach.

The situation becomes even more interesting when one consider bounded-degree graphs. In Chapter 3 we have studied the complexity of approximating the partition function $Z_{\text{Ising}}(G; \beta)$ in terms of the relation between the edge interaction β and a parameter Δ which is an upper bound on the maximum degree of the input graph G. We have established both new tractability results and new intractability results. Our tractability results show that $Z_{\text{Ising}}(-; \beta)$ has an FPTAS when $|\beta - 1|/|\beta + 1| < \tan(\pi/(4\Delta - 4))$, and we have reached this result by proving that there are no inputs G that make the partition function 0 when β is in this range (Theorem 1.5). Our result significantly extends the known zero-free region of the Ising model (and hence the known approximation results). Regarding intractability, we have shown that it is #P-hard to approximate $Z_{\text{Ising}}(-; \beta)$ when $\beta \in \mathbb{C}$ is an algebraic number such that $\beta \notin \mathbb{R} \cup \{i, -i\}$ and $|\beta - 1|/|\beta + 1| > 1/\sqrt{\Delta - 1}$ (Theorem 1.7). Moreover, we have demonstrated situations in which zeros of the partition function imply hardness of approximation in the Ising model.

These are the first results to show intractability of approximating $Z_{\text{Ising}}(-, \beta)$ on bounded degree graphs with complex β, and several questions remain. Here we leave the tantalising problem of improving our restriction $|\beta-1|/|\beta+1| < 1/\sqrt{\Delta - 1}$ in Theorem 1.7 to $|\beta-1|/|\beta+1| < 1/(\Delta - 1)$, which is unreachable with our constructions due to the symmetry of the Ising model without a field, see Section 3.4. In fact, a complete complexity map of approximability seems out of reach using the current knowledge in this area of research, as there is currently no technique that allows us to find the largest zero-free region of the partition function. This is the case even for other extremely well-studied models in statistical mechanics and approximate counting, such as the hardcore model. Another important open question is showing that a zero of the partition function at an edge interaction β imply hardness of approximation for this edge interaction (Conjecture 3.4). This question has actually been resolved in the case of the hardcore model/independent set polynomial. However, in the Ising model we are restricted by the fact that trees without pinnings do not implement any meaningful edge interactions due to the symmetry of the model and, thus, current techniques are not enough to prove Conjecture 3.4 as we have discussed in Section 3.5.

Finally, in Chapter 4 we have delved into the random k-SAT model, providing the first almost-uniform sampler for satifying assignments in this model that runs in almost-linear time. Our algorithm holds even when the density of the formula scales exponentially with k, where correlation decay arguments fail. In our proofs we have shown how to relate local sparsity

properties of the random k-SAT model to the geometry of the space of satisfying assignments via marking techniques, thus, leading also to results about connectivity and looseness of satisfying assignments of random k-CNF formulas. These arguments have also allowed us to provide spectral independence bounds for the Glauber dynamics, this being the first instance of an application of spectral independence that does not rely on correlation decay or spatial mixing.

An open problem is finding the optimal density threshold for sampling in the random k-SAT model (recall that the threshold for our algorithm is $\alpha \leq 2^{0.039k}$). At the moment there is no strong reason to believe that this threshold should not be close to the satisfiability threshold of the random k-SAT model, which is $\alpha_\star(k) = 2^k \log 2 + O(1)$. In fact, algorithms to find satisfying assignment have managed to succeed for densities up to $(1 + o_k(1))2^k(\log k)/k$, though going beyond such densities is a major open problem in the area. In the case of random monotone k-CNF formulas (i.e. the case when every literal is positive), fast sampling has been shown to be feasible when $\alpha = O(1)2^k/k$, see [70]. In this line of thought, another open area of research is analysing the situation in related models such as NAE-SAT. On another note, recall that our algorithm requires k to be large enough. We leave the open problem of finding a fast sampling algorithm for small k. Recall that our techniques build on marking approaches that have been highly successful in the literature; in these the large k condition is needed when applying the Lóvasz Local lemma to find a marking. A novel technique to find such markings may lead to further progress in the area. A place to start this line of research is the random 2-SAT model, where there is a formula for the free energy of the model, see [2]. However, the sampling problem is not straightforward even for 2-SAT, as correlation decay results that hold for the whole satisfiability spectrum are unknown. In conclusion, our results open the door to study a wide range of fundamental sampling problems in approximate counting and random models.

Bibliography

[1] Dimitris Achlioptas and Amin Coja-Oghlan. Algorithmic barriers from phase transitions. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008*, pages 793–802. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.11.

[2] Dimitris Achlioptas, Amin Coja-Oghlan, Max Hahn-Klimroth, Joon Lee, Noëla Müller, Manuel Penschuck, and Guangyan Zhou. The number of satisfying assignments of random 2-SAT formulas. *Random Structures Algorithms*, 58(4):609–647, 2021. doi:10.1002/rsa.20993.

[3] Dimitris Achlioptas, Amin Coja-Oghlan, and Federico Ricci-Tersenghi. On the solution-space geometry of random constraint satisfaction problems. *Random Structures Algorithms*, 38(3):251–268, 2011. doi:10.1002/rsa.20323.

[4] Dimitris Achlioptas and Cristopher Moore. The asymptotic order of the random k-SAT threshold. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 779–788. IEEE, 2002. doi:10.1109/SFCS.2002.1182003.

[5] Vedat Levi Alev and Lap Chi Lau. Improved analysis of higher order random walks and applications. In *STOC '20—Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1198–1211. ACM, New York, 2020. doi:10.1145/3357713.3384317.

[6] Konrad Anand and Mark Jerrum. Perfect sampling in infinite spin systems via strong spatial mixing. *SIAM J. Comput.*, 51(4):1280–1295, 2022. doi:10.1137/21M1437433.

[7] Nima Anari, Kuikui Liu, and Shayan Oveis Gharan. Spectral independence in high-dimensional expanders and applications to the hardcore model. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science*, pages 1319–1330. 2020. doi:10.1109/FOCS46700.2020.00125.

[8] Alexander Barvinok. *Combinatorics and complexity of partition functions*, volume 30 of *Algorithms and Combinatorics*. Springer, Cham, 2016. doi:10.1007/978-3-319-51829-9.

[9] Alexander Barvinok and Nicholas Barvinok. More on zeros and approximation of the Ising partition function. *Forum Math. Sigma*, 9:Paper No. e46, 18, 2021. doi:10.1017/fms.2021.40.

[10] Alexander Barvinok and Guus Regts. Weighted counting of solutions to sparse systems of equations. *Combin. Probab. Comput.*, 28(5):696–719, 2019. doi:10.1017/s0963548319000105.

[11] Alan F. Beardon. *Iteration of rational functions*, volume 132 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1991. Complex analytic dynamical systems. doi:10.1007/978-1-4612-4422-6.

[12] Ioana Bena, Michel Droz, and Adam Lipowski. Statistical mechanics of equilibrium and nonequilibrium phase transitions: the Yang-Lee formalism. *Internat. J. Modern Phys. B*, 19(29):4269–4329, 2005. doi:10.1142/S0217979205032759.

[13] Ferenc Bencs and Péter Csikvári. Note on the zero-free region of the hard-core model. *arXiv preprint*, 2020. arXiv:1807.08963.

[14] Ferenc Bencs, Jeroen Huijben, and Guus Regts. Approximating the chromatic polynomial is as hard as computing it exactly. *arXiv preprint*, 2022. arXiv:2211.13790.

[15] Ivona Bezáková, Andreas Galanis, Leslie Ann Goldberg, and Daniel Štefankovič. Inapproximability of the independent set polynomial in the complex plane. *SIAM J. Comput.*, 49(5):STOC18–395–STOC18–448, 2020. doi:10.1137/18M1184485.

[16] Ivona Bezáková, Andreas Galanis, Leslie Ann Goldberg, and Daniel Štefankovič. The complexity of approximating the matching polynomial in the complex plane. *ACM Trans. Comput. Theory*, 13(2):Art. 13, 37, 2021. doi:10.1145/3448645.

[17] Ivona Bezáková, Andreas Galanis, Leslie Ann Goldberg, and Daniel Štefankovič. Fast sampling via spectral independence beyond bounded-degree graphs. In *49th EATCS International Conference on Automata, Languages, and Programming*, volume 229 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 21, 16. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2022. doi:10.4230/lipics.icalp.2022.21.

[18] Ivona Bezáková, Daniel Štefankovič, Vijay V. Vazirani, and Eric Vigoda. Accelerating simulated annealing for the permanent and combinatorial counting problems. *SIAM J. Comput.*, 37(5):1429–1454, 2008. doi:10.1137/050644033.

[19] Antonio Blanca, Pietro Caputo, Zongchen Chen, Daniel Parisi, Daniel Štefankovič, and Eric Vigoda. On mixing of Markov chains: coupling, spectral independence, and entropy factorization. volume 27, pages Paper No. 142, 42, 2022. doi:10.1214/22-ejp867.

[20] M. Bordewich, M. Freedman, L. Lovász, and D. Welsh. Approximate counting and quantum computation. *Combin. Probab. Comput.*, 14(5-6):737–754, 2005. doi:10.1017/S0963548305007005.

[21] Russell J. Bradford and James H. Davenport. Effective tests for cyclotomic polynomials. In *Symbolic and algebraic computation (Rome, 1988)*, volume 358 of *Lecture Notes in Comput. Sci.*, pages 244–251. Springer, Berlin, 1989. URL: https://doi.org/10.1007/3-540-51084-2_22, doi:10.1007/3-540-51084-2_22.

[22] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph classes: a survey*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999. doi:10.1137/1.9780898719796.

[23] Michael Brin and Garrett Stuck. *Introduction to dynamical systems*. Cambridge University Press, Cambridge, 2002. doi:10.1017/CBO9780511755316.

[24] Jason I. Brown, Carl Hickman, Alan D. Sokal, and David G. Wagner. On the chromatic roots of generalized theta graphs. *J. Combin. Theory Ser. B*, 83(2):272–297, 2001. doi:10.1006/jctb.2001.2057.

[25] Pjotr Buys. Cayley trees do not determine the maximal zero-free locus of the independence polynomial. *Michigan Math. J.*, 70(3):635–648, 2021. doi:10.1307/mmj/1599206419.

[26] Pjotr Buys, Andreas Galanis, Viresh Patel, and Guus Regts. Lee-Yang zeros and the complexity of the ferromagnetic Ising model on bounded-degree graphs. *Forum Math. Sigma*, 10:Paper No. e7, 43, 2022. doi:10.1017/fms.2022.4.

[27] Jin-Yi Cai, Xi Chen, and Pinyan Lu. Graph homomorphisms with complex values: a dichotomy theorem. volume 42, pages 924–1029, 2013. doi:10.1137/110840194.

[28] Zongchen Chen, Kuikui Liu, and Eric Vigoda. Rapid mixing of Glauber dynamics up to uniqueness via contraction. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science*, pages 1307–1318. 2020. doi:10.1109/FOCS46700.2020.00124.

[29] Zongchen Chen, Kuikui Liu, and Eric Vigoda. Optimal mixing of glauber dynamics: entropy factorization via high-dimensional expansion. In *STOC '21—Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1537–1550. ACM, New York, 2021. doi:10.1145/3406325.3451035.

[30] Amin Coja-Oghlan. A better algorithm for random k-SAT. *SIAM J. Comput.*, 39(7):2823–2864, 2010. doi:10.1137/09076516X.

[31] Amin Coja-Oghlan and Alan Frieze. Analyzing Walksat on random formulas. *SIAM J. Comput.*, 43(4):1456–1485, 2014. doi:10.1137/12090191X.

[32] Amin Coja-Oghlan, Noela Müller, and Jean B. Ravelomanana. Belief propagation on the random k-SAT model. *Ann. Appl. Probab.*, 32(5):3718–3796, 2022. doi:10.1214/21-aap1772.

[33] Amin Coja-Oghlan and Angelica Y. Pachon-Pinzon. The decimation process in random k-SAT. *SIAM J. Discrete Math.*, 26(4):1471–1509, 2012. doi:10.1137/110842867.

[34] Amin Coja-Oghlan and Konstantinos Panagiotou. The asymptotic k-SAT threshold. *Adv. Math.*, 288:985–1068, 2016. doi:10.1016/j.aim.2015.11.007.

[35] Amin Coja-Oghlan and Daniel Reichman. Sharp thresholds and the partition function. In *Journal of Physics: Conference Series*, volume 473, page 012015. IOP Publishing, 2013. doi:10.1088/1742-6596/473/1/012015.

[36] Amin Coja-Oghlan and Nick Wormald. The number of satisfying assignments of random regular k-SAT formulas. *Combin. Probab. Comput.*, 27(4):496–530, 2018. doi:10.1017/S0963548318000263.

[37] David de Boer, Pjotr Buys, Lorenzo Guerini, Han Peters, and Guus Regts. Zeros, chaotic ratios and the computational complexity of approximating the independence polynomial. *arXiv preprint*, 2021. arXiv:2104.11615.

[38] Jian Ding, Allan Sly, and Nike Sun. Proof of the satisfiability conjecture for large k. *Ann. of Math. (2)*, 196(1):1–388, 2022. doi:10.4007/annals.2022.196.1.1.

[39] Martin Dyer, Leslie Ann Goldberg, Catherine Greenhill, and Mark Jerrum. The relative complexity of approximate counting problems. *Algorithmica*, 38(3):471–500, 2004. Approximation algorithms. doi:10.1007/s00453-003-1073-y.

[40] Martin Dyer and Catherine Greenhill. The complexity of counting graph homomorphisms. *Random Structures Algorithms*, 17(3-4):260–289, 2000. doi:10.1002/1098-2418(200010/12)17:3/4<260::AID-RSA5>3.3.CO;2-N.

[41] Ioannis Z. Emiris, Bernard Mourrain, and Elias P. Tsigaridas. Real algebraic numbers: Complexity analysis and experimentation. In Peter Hertling, Christoph M. Hoffmann, Wolfram Luther, and Nathalie Revol, editors, *Reliable Implementation of Real Number Algorithms: Theory and Practice*, pages 57–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi:10.1007/978-3-540-85521-7_4.

[42] Paul Erdős and László Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and finite sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th birthday), Vol. II*, pages 609–627. Colloq. Math. Soc. János Bolyai, 1975. URL: https://www.renyi.hu/~p_erdos/1975-34.pdf.

[43] Weiming Feng, Heng Guo, Yitong Yin, and Chihao Zhang. Fast sampling and counting k-SAT solutions in the local lemma regime. *J. ACM*, 68(6):Art. 40, 42, 2021. doi:10.1145/3469832.

[44] Weiming Feng, Kun He, and Yitong Yin. Sampling constraint satisfaction solutions in the local lemma regime. In *STOC '21—Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1565–1578. ACM, New York, 2021. doi:10.1145/3406325.3451101.

[45] Jacob Focke, Leslie Ann Goldberg, and Stanislav Zivny. The complexity of counting surjective homomorphisms and compactions. *SIAM J. Discrete Math.*, 33(2):1006–1043, 2019.

[46] Alan Frieze and Stephen Suen. Analysis of two simple heuristics on a random instance of k-SAT. *J. Algorithms*, 20(2):312–355, 1996. doi:10.1006/jagm.1996.0016.

[47] Andreas Galanis, Leslie A. Goldberg, and Andres Herrera-Poyatos. The complexity of approximating the complex-valued Ising model on bounded degree graphs. *SIAM J. Discrete Math.*, 36(3):2159–2204, 2022. doi:10.1137/21M1454043.

[48] Andreas Galanis, Leslie Ann Goldberg, Heng Guo, and Andrés Herrera-Poyatos. Fast sampling of satisfying assignments from random k-sat. *arXiv preprint*, 2022. arXiv:2206.15308.

[49] Andreas Galanis, Leslie Ann Goldberg, Heng Guo, and Kuan Yang. Counting solutions to random CNF formulas. *SIAM J. Comput.*, 50(6):1701–1738, 2021. doi:10.1137/20M1351527.

[50] Andreas Galanis, Leslie Ann Goldberg, and Andrés Herrera-Poyatos. The complexity of approximating the complex-valued potts model. In *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 36:1–36:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2020.36.

[51] Andreas Galanis, Leslie Ann Goldberg, and Andrés Herrera-Poyatos. The complexity of approximating the complex-valued potts model. *Comput. Complexity*, 31(1):Paper No. 2, 2022. doi:10.1007/s00037-021-00218-x.

[52] Andreas Galanis, Daniel Štefankovič, and Eric Vigoda. Inapproximability of the partition function for the antiferromagnetic Ising and hard-core models. *Combin. Probab. Comput.*, 25(4):500–559, 2016. doi:10.1017/S0963548315000401.

[53] Hans-Otto Georgii. *Gibbs measures and phase transitions*, volume 9 of *De Gruyter Studies in Mathematics*. Walter de Gruyter & Co., Berlin, second edition, 2011. doi:10.1515/9783110250329.

[54] Christopher D. Godsil. Matchings and walks in graphs. *J. Graph Theory*, 5(3):285–297, 1981. doi:10.1002/jgt.3190050310.

[55] Leslie Ann Goldberg and Heng Guo. The complexity of approximating complex-valued Ising and Tutte partition functions. *Comput. Complexity*, 26(4):765–833, 2017. doi:10.1007/s00037-017-0162-2.

[56] Leslie Ann Goldberg and Mark Jerrum. Approximating the partition function of the ferromagnetic Potts model. *J. ACM*, 59(5):Art. 25, 31, 2012. doi:10.1145/2371656.2371660.

[57] Leslie Ann Goldberg and Mark Jerrum. Inapproximability of the Tutte polynomial of a planar graph. *Comput. Complexity*, 21(4):605–642, 2012. doi:10.1007/s00037-012-0046-4.

[58] Leslie Ann Goldberg and Mark Jerrum. Inapproximability of the Tutte polynomial of a planar graph. *Comput. Complexity*, 21(4):605–642, 2012. doi:10.1007/s00037-012-0046-4.

[59] Leslie Ann Goldberg and Mark Jerrum. The complexity of computing the sign of the Tutte polynomial. *SIAM J. Comput.*, 43(6):1921–1952, 2014. doi:10.1137/12088330X.

[60] Leslie Ann Goldberg and Mark Jerrum. Approximating pairwise correlations in the Ising model. *ACM Trans. Comput. Theory*, 11(4):Art. 23, 20, 2019. doi:10.1145/3337785.

[61] Heng Guo, Mark Jerrum, and Jingcheng Liu. Uniform sampling through the Lovász local lemma. *J. ACM*, 66(3):Art. 18, 31, 2019. doi:10.1145/3310131.

[62] Heng Guo, Chao Liao, Pinyan Lu, and Chihao Zhang. Zeros of Holant problems: locations and algorithms. *ACM Trans. Algorithms*, 17(1):Art. 4, 25, 2021. doi:10.1145/3418056.

[63] Heng Guo, Jingcheng Liu, and Pinyan Lu. Zeros of ferromagnetic 2-spin systems. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms*, pages 181–192. SIAM, Philadelphia, PA, 2020. URL: https://dl.acm.org/doi/abs/10.5555/3381089.3381100.

[64] Bernhard Haeupler, Barna Saha, and Aravind Srinivasan. New constructive aspects of the Lovász local lemma. *J. ACM*, 58(6):Art. 28, 28, 2011. doi:10.1145/2049697.2049702.

[65] Aram W. Harrow, Saeed Mehraban, and Mehdi Soleimanifar. Classical algorithms, correlation decay, and complex zeros of partition functions of quantum many-body systems. In *STOC '20—Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 378–386. ACM, New York, [2020] ©2020. doi:10.1145/3357713.3384322.

[66] Kun He, Chunyang Wang, and Yitong Yin. Sampling Lovász local lemma for general constraint satisfaction solutions in near-linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science—FOCS 2022*, pages 147–158. IEEE Computer Soc., Los Alamitos, CA, 2022. doi:10.1109/FOCS54457.2022.00021.

[67] Kun He, Chunyang Wang, and Yitong Yin. Deterministic counting Lovász local lemma beyond linear programming. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3388–3425. SIAM, Philadelphia, PA, 2023. doi:10.1137/1.9781611977554.ch130.

[68] Kun He, Kewen Wu, and Kuan Yang. Improved bounds for sampling solutions of random CNF formulas. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3330–3361. SIAM, Philadelphia, PA, 2023. doi:10.1137/1.9781611977554.ch128.

[69] Ole J. Heilmann and Elliott H. Lieb. Theory of monomer-dimer systems. *Comm. Math. Phys.*, 25:190–232, 1972. URL: http://projecteuclid.org/euclid.cmp/1103857921.

[70] Jonathan Hermon, Allan Sly, and Yumeng Zhang. Rapid mixing of hypergraph independent sets. *Random Structures Algorithms*, 54(4):730–767, 2019. doi:10.1002/rsa.20830.

[71] Mark Huber. Approximation algorithms for the normalizing constant of Gibbs distributions. *Ann. Appl. Probab.*, 25(2):974–985, 2015. doi:10.1214/14-AAP1015.

[72] Bill Jackson. A zero-free interval for chromatic polynomials of graphs. *Combinatorics, Probability and Computing*, 2(3):325–336, 1993. doi:10.1017/S0963548300000705.

[73] François Jaeger, Dirk L. Vertigan, and Dominic J. A. Welsh. On the computational complexity of the Jones and Tutte polynomials. *Math. Proc. Cambridge Philos. Soc.*, 108(1):35–53, 1990. doi:10.1017/S0305004100068936.

[74] Vishesh Jain, Huy Tuan Pham, and Thuy-Duong Vuong. On the sampling Lovász local lemma for atomic constraint satisfaction problems. *arXiv preprint*, 2021. arXiv:2102.08342.

[75] Vishesh Jain, Huy Tuan Pham, and Thuy Duong Vuong. Towards the sampling Lovász Local Lemma. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science—FOCS 2021*, pages 173–183. IEEE Computer Soc., Los Alamitos, CA, 2022. doi:10.1109/FOCS52979.2021.00025.

[76] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM J. Comput.*, 22(5):1087–1116, 1993. doi:10.1137/0222066.

[77] 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. doi:10.1016/0304-3975(86)90174-X.

[78] R. Kannan, A. K. Lenstra, and L. Lovász. Polynomial factorization and nonrandomness of bits of algebraic and some transcendental numbers. *Math. Comp.*, 50(181):235–250, 1988. doi:10.2307/2007927.

[79] Tali Kaufman and Izhar Oppenheim. High order random walks: beyond spectral gap. *Combinatorica*, 40(2):245–281, 2020. doi:10.1007/s00493-019-3847-0.

[80] Ker-I Ko. *Complexity theory of real functions*. Progress in Theoretical Computer Science. Birkhäuser Boston, Inc., Boston, MA, 1991. doi:10.1007/978-1-4684-6802-1.

[81] Vladimir Kolmogorov. A faster approximation algorithm for the Gibbs partition function. In *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 228–249. PMLR, 06–09 Jul 2018. URL: https://proceedings.mlr.press/v75/kolmogorov18a.html.

[82] Michael Kowalczyk and Jin-Yi Cai. Holant problems for 3-regular graphs with complex edge functions. *Theory Comput. Syst.*, 59(1):133–158, 2016. doi:10.1007/s00224-016-9671-7.

[83] Florent Krząkał a, Andrea Montanari, Federico Ricci-Tersenghi, Guilhem Semerjian, and Lenka Zdeborová. Gibbs states and the set of solutions of random constraint satisfaction problems. *Proc. Natl. Acad. Sci. USA*, 104(25):10318–10323, 2007. doi:10.1073/pnas.0703685104.

[84] Greg Kuperberg. How hard is it to approximate the Jones polynomial? *Theory Comput.*, 11:183–219, 2015. doi:10.4086/toc.2015.v011a006.

[85] Elliott H. Lieb and Alan D. Sokal. A general Lee-Yang theorem for one-component and multicomponent ferromagnets. *Comm. Math. Phys.*, 80(2):153–179, 1981. URL: http://projecteuclid.org/euclid.cmp/1103919874.

[86] Jingcheng Liu, Alistair Sinclair, and Piyush Srivastava. Fisher zeros and correlation decay in the Ising model. *J. Math. Phys.*, 60(10):103304, 12, 2019. doi:10.1063/1.5082552.

[87] Jingcheng Liu, Alistair Sinclair, and Piyush Srivastava. The Ising partition function: zeros and deterministic approximation. *J. Stat. Phys.*, 174(2):287–315, 2019. doi:10.1007/s10955-018-2199-2.

[88] Jingcheng Liu, Alistair Sinclair, and Piyush Srivastava. A deterministic algorithm for counting colorings with 2Δ colors. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science*, pages 1380–1404. IEEE Comput. Soc. Press, Los Alamitos, CA, [2019] ©2019. doi:10.1109/FOCS.2019.00085.

[89] Ryan L. Mann and Michael J. Bremner. Approximation Algorithms for Complex-Valued Ising Models on Bounded Degree Graphs. *Quantum*, 3:162, July 2019. doi:10.22331/q-2019-07-11-162.

[90] Marc Mézard, Thierry Mora, and Riccardo Zecchina. Clustering of solutions in the random satisfiability problem. *Physical Review Letters*, 94(19):197205, 2005. doi:10.1103/PhysRevLett.94.197205.

[91] John Milnor. *Dynamics in one complex variable*, volume 160 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ, third edition, 2006.

[92] Michael Mitzenmacher and Eli Upfal. *Probability and computing*. Cambridge University Press, Cambridge, 2005. Randomized algorithms and probabilistic analysis. doi:10.1017/CBO9780511813603.

[93] Ankur Moitra. Approximate counting, the Lovász local lemma, and inference in graphical models. *J. ACM*, 66(2):Art. 10, 25, 2019. doi:10.1145/3268930.

[94] Rémi Monasson and Riccardo Zecchina. Statistical mechanics of the random K-satisfiability model. *Phys. Rev. E (3)*, 56(2):1357–1370, 1997. doi:10.1103/PhysRevE.56.1357.

[95] Andrea Montanari and Devavrat Shah. Counting good truth assignments of random k-SAT formulae. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1255–1264. ACM, New York, 2007. URL: https://dl.acm.org/doi/abs/10.5555/1283383.1283518.

[96] Thierry Mora, Marc Mézard, and Riccardo Zecchina. Pairs of sat assignments and clustering in random boolean formulae. *arXiv preprint*, 2007. URL: https://arxiv.org/abs/cond-mat/0506053.

[97] Robin A. Moser and Gábor Tardos. A constructive proof of the general Lovász local lemma. *J. ACM*, 57(2):Art. 11, 15, 2010. doi:10.1145/1667053.1667060.

[98] Wolfgang Mulzer. Five proofs of Chernoff's bound with applications. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, 1(124):59–76, 2018. URL: http://bulletin.eatcs.org/index.php/beatcs/article/view/525.

[99] Danny Nam, Allan Sly, and Youngtak Sohn. One-step replica symmetry breaking of random regular NAE-SAT. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science—FOCS 2021*, pages 310–318. IEEE Computer Soc., Los Alamitos, CA, 2022. doi:10.1109/FOCS52979.2021.00039.

[100] Terrence Napier and Mohan Ramachandran. *An introduction to Riemann surfaces*. Cornerstones. Birkhäuser/Springer, New York, 2011.

[101] Viresh Patel and Guus Regts. Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials. *SIAM J. Comput.*, 46(6):1893–1919, 2017. doi:10.1137/16M1101003.

[102] Viresh Patel, Guus Regts, et al. Approximate counting using taylor's theorem: a survey. *Bulletin of EATCS*, 138(3), 2022. URL: http://150.140.5.98/index.php/beatcs/article/view/725.

[103] Han Peters and Guus Regts. On a conjecture of Sokal concerning roots of the independence polynomial. *Michigan Math. J.*, 68(1):33–55, 2019. doi:10.1307/mmj/1541667626.

[104] Han Peters and Guus Regts. Location of zeros for the partition function of the Ising model on bounded degree graphs. *J. Lond. Math. Soc. (2)*, 101(2):765–785, 2020. doi:10.1112/jlms.12286.

[105] Renfrey B. Potts. Some generalized order-disorder transformations. *Proc. Cambridge Philos. Soc.*, 48:106–109, 1952.

[106] J. Scott Provan and Michael O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Comput.*, 12(4):777–788, 1983. doi:10.1137/0212053.

[107] Alistair Sinclair, Piyush Srivastava, and Marc Thurley. Approximation algorithms for two-state anti-ferromagnetic spin systems on bounded degree graphs. *J. Stat. Phys.*, 155(4):666–686, 2014. doi:10.1007/s10955-014-0947-5.

[108] Allan Sly, Nike Sun, and Yumeng Zhang. The number of solutions for random regular NAE-SAT. *Probab. Theory Related Fields*, 182(1-2):1–109, 2022. doi:10.1007/s00440-021-01029-5.

[109] Alan D. Sokal. The multivariate Tutte polynomial (alias Potts model) for graphs and matroids. In *Surveys in combinatorics 2005*, volume 327 of *London Math. Soc. Lecture Note Ser.*, pages 173–226. Cambridge Univ. Press, Cambridge, 2005. doi:10.1017/CBO9780511734885.009.

[110] Joel Spencer. Asymptotic lower bounds for Ramsey functions. *Discrete Math.*, 20(1):69–76, 1977/78. doi:10.1016/0012-365X(77)90044-9.

[111] Elias M. Stein and Rami Shakarchi. *Complex analysis*, volume 2 of *Princeton Lectures in Analysis*. Princeton University Press, Princeton, NJ, 2003.

[112] Ian Stewart. *Galois theory*. CRC Press, Boca Raton, FL, fourth edition, 2015.

[113] Adam Wojciech Strzeboński. Computing in the field of complex algebraic numbers. *J. Symbolic Comput.*, 24(6):647–656, 1997. doi:10.1006/jsco.1997.0158.

[114] Morwen B. Thistlethwaite. A spanning tree expansion of the Jones polynomial. *Topology*, 26(3):297–309, 1987. doi:10.1016/0040-9383(87)90003-6.

[115] Leslie G. Valiant. The complexity of computing the permanent. *Theoret. Comput. Sci.*, 8(2):189–201, 1979. doi:10.1016/0304-3975(79)90044-6.

[116] Dirk Vertigan. The computational complexity of Tutte invariants for planar graphs. *SIAM J. Comput.*, 35(3):690–712, 2005. doi:10.1137/S0097539704446797.

[117] Daniel Štefankovič, Santosh Vempala, and Eric Vigoda. Adaptive simulated annealing: a near-optimal connection between sampling and counting. *J. ACM*, 56(3):Art. 18, 36, 2009. doi:10.1145/1516512.1516520.

[118] Michel Waldschmidt. *Diophantine approximation on linear algebraic groups*, volume 326 of *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical*

Sciences]. Springer-Verlag, Berlin, 2000. Transcendence properties of the exponential function in several variables. doi:10.1007/978-3-662-11569-5.

[119] Dror Weitz. Combinatorial criteria for uniqueness of Gibbs measures. *Random Structures Algorithms*, 27(4):445–475, 2005. doi:10.1002/rsa.20073.

[120] Dror Weitz. Counting independent sets up to the tree threshold. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 140–149. ACM, New York, 2006. doi:10.1145/1132516.1132538.

[121] Dominic J. A. Welsh. *Complexity: knots, colourings and counting*, volume 186 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1993. doi:10.1017/CBO9780511752506.

[122] Chen N. Yang and T. D. Lee. Statistical theory of equations of state and phase transitions. I. Theory of condensation. *Phys. Rev. (2)*, 87:404–409, 1952.

[123] Chee Keng Yap. *Fundamental problems of algorithmic algebra*. Oxford University Press, New York, 2000.

[124] Lenka Zdeborová. Statistical physics of hard optimization problems. *arXiv preprint*, 2008. arXiv:0806.4112.