UNIVERSITY OF CALIFORNIA
Los Angeles

Counting Linear Extensions and Contingency Tables

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

Samuel John Dittmer

2019

ABSTRACT OF THE DISSERTATION

Counting Linear Extensions
and Contingency Tables

by

Samuel John Dittmer
Doctor of Philosophy in Mathematics
University of California, Los Angeles, 2019
Professor Igor Pak, Chair

This dissertation investigates the difficulty of counting two classes of combinatorial objects, linear extensions of posets and contingency tables. For linear extensions of posets, we prove a number of hardness results. We show that computing the parity of the number of linear extensions of dimension two is $\oplus\mathsf{P}$-complete. We extend this result to show that counting linear extensions of dimension two posets is $\#\mathsf{P}$-complete, answering a question posed by Möhring[Möh89] and by Felsner and Wernisch [FW97]. We also show that counting linear extensions of height two posets is $\#\mathsf{P}$-complete, resolving a conjecture of Brightwell and Winkler [BW91]. We extend this result to show that counting linear extensions of incidence posets is $\#\mathsf{P}$-complete.

For the results about posets of dimension two we employed a computer search to construct families of permutations that behave as logic gates in a certain setting. The results about height two posets and incidence posets rely instead on gadgets that were constructed by hand.

For contingency tables, we work from the opposite direction, proving results about the feasibility of approximately counting and sampling tables. We give a new algorithm for approximating the number of contingency tables with fixed margins, which we call the SHM algorithm. We prove that the SHM algorithm is a fully polynomial random approximation scheme (FPRAS) for the number of tables for certain families of sparse margins. We then

use this result to establish a polynomial mixing time for the Diaconis-Gangolli chain with sparse margins.

Using our SHM algorithm and techniques in discrete probability, we present experimental and theoretical evidence in support of answers to a number of questions Barvinok posed about the distribution of individual entries in contingency tables [Bar10b]. In particular, we show that for a certain set of margins considered by Barvinok, and under certain assumptions about weak independence of entries, the distribution of the corner table entry exhibits a phase transition in mean and distribution.

The dissertation of Samuel John Dittmer is approved.

Rafail Ostrovsky

Artem Chernikov

Bruce L. Rothschild

Igor Pak, Committee Chair

University of California, Los Angeles

2019

To the lantern out of doors

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I am grateful for my advisor, Igor Pak and his unfailing guidance, warmth and humor. He taught me more about math than I will ever know. I would also like to thank Bruce Rothschild, Artem Chernikov, Rafail Ostrovsky, Alejandro Morales, Hanbaek Lyu, John Zhang, Bon-Soon Lin, Mike Miller and Zach Boyd for many insights and delightful conversations.

I am grateful for my family and my friends and their love.

Much of the work in this thesis is from my joint research papers. Chapters 2 and 3 are from [DP19+a], joint with Igor Pak. Chapters 4, 5 and 7 are from [DP19+b, DP19+d] and [DP19+c], respectively, also with Igor Pak. Chapter 6 is from the joint work [DLP19+b] with Hanbaek Lyu and Igor Pak.

VITA

2014 B.S. (Mathematics), Brigham Young University, Provo, Utah.

PUBLICATIONS

(with Igor Pak) Counting linear extensions of dimension two posets, submitted (2019).

(with Igor Pak) Counting linear extensions of restricted posets, submitted (2019).

(with Igor Pak) Random sampling and approximate counting of sparse contingency tables, submitted (2019).

(with Hanbaek Lyu and Igor Pak) Phase transition in random contingency tables with non-uniform margins, submitted (2019) arXiv:1903.08743.

(with Hanbaek Lyu and Igor Pak) Phase transition in dense contingency tables, in preparation (2019).

(with Alexander J. Diesl and Pace P. Nielsen) Idempotent lifting and ring extensions, J. Algebra Appl., vol. 15 issue 6, (2016) 16 pp.

Spoof odd perfect numbers, Math. Comp. vol. 83 (2014), 2575-2582.

(with Pace Nielsen) On a question of Hartwig and Luh, Bull. Aust. Math. Soc. vol. 89 (2014), 271-278.

(with Michael Proulx and Stephanie Seybert) Some arithmetic problems related to class group L-functions, Ramanujan Jour. vol. 37, issue 2 (2015), 257-268.

Part I

Linear extensions

CHAPTER 1

Introduction

1.1 Linear Extensions

Counting *linear extensions* (#LE) of a finite poset is a fundamental problem in both Combinatorics and Computer Science, with connections and applications ranging from Statistics to Optimization, to Social Choice Theory. It is primarily motivated by the following basic question: given partial information about preferences between various objects, what are the chances of other comparisons?

In 1991, Brightwell and Winkler showed that #LE is #P-complete [BW91]. This resolved the 1986 conjecture by Linial [Lin86]. The #P-completeness of the following natural extension was first posed in 1988 by Möhring [Möh89, p. 163], and then again in 1997 by Felsner and Wernisch [FW97] motivated by different applications.

> #D2LE (*Number of linear extensions of dimension-2 posets*)
>
> **Input:** A partially ordered set P of dimension two.
>
> **Output:** The number $\boldsymbol{e}(P)$ of linear extensions of P.

Here the poset P is said to have *dimension two* if it can be represented by a finite set of points $\{(x_1, y_1), \ldots, (x_n, y_n)\} \subset \mathbb{R}^2$, with the inequalities $(x_i, y_i) \preccurlyeq (x_j, y_j)$ if $x_i \leq x_j$ and $y_i \leq y_j$, $i \neq j$. Equivalently, poset P has dimension two if and only if its comparability graph $\Gamma(P)$ has complement $\overline{\Gamma(P)} \simeq \Gamma(P^*)$, for a *dual poset* P^* (see e.g. [Tro92]).

In Chapter 3, we prove:

Theorem 1.1.1. #D2LE *is* #P*-complete.*

We also consider the parity of $\boldsymbol{e}(P)$.

> $\oplus$D2LE (*Parity of linear extensions of dimension-2 posets*)
>
> **Input:** A partially ordered set P of dimension two.
>
> **Output:** The parity of the number $\boldsymbol{e}(P)$ of linear extensions of P.

In Chapter 2, we prove:

Theorem 1.1.2. $\oplus$D2LE *is* $\oplus\mathsf{P}$*-complete.*

Remark 1.1.3. This result is independent of Theorem 1.1.1, though the arguments are closely related. The proof of Theorem 1.1.1 relies on mod p reductions, for a collection of primes p whose size depends on the number of clauses in the corresponding 3SAT formula. The complexity of computing #D2LE mod p for any prime greater than 2 remains open. We conjecture that the #D2LE mod p problem is mod p-complete for every prime p.

As a motivation, Felsner and Wernisch [FW97] show that #D2LE is equivalent to the following problem on the number of possible bubble sorted permutations τ from a given $\sigma \in S_n$ (see also [BjW91, Reu96]).

> #Bruhat (*Size of principal ideal in the weak Bruhat order*)
>
> **Input:** A permutation $\sigma \in S_n$.
>
> **Output:** The number $\boldsymbol{e}(\sigma)$ of permutations $\tau \in S_n$ with $\tau \leq \sigma$.

Here we write $\tau \leq \sigma$ if τ can be obtained from σ by a *bubble sorting*: repeated application of adjacent transpositions which the minimal possible number of inversions:

$$\sigma \;=\; \tau \cdot (i_1, i_1+1) \cdots (i_\ell, i_\ell+1), \quad \text{where} \quad inv(\sigma) \;=\; inv(\tau) + \ell.$$

The *weak Bruhat order* B_n is defined to be $(S_n, \leq)$. In #Bruhat, we consider the principal ideal $P_\sigma = B_n \cap \{\omega \leq \sigma\}$, so in the notation above $\boldsymbol{e}(\sigma) = \boldsymbol{e}(P_\sigma)$. Note that #Bruhat is

in #P. Because the reduction is parsimonious, it likewise implies that ⊕Bruhat is equivalent to ⊕D2LE.

We include a quick proof of the reduction of #D2LE to #Bruhat in §2.1.2, both for completeness and to introduce the framework for the proof of the main result.

Theorem 1.1.4. #Bruhat *is* #P*-complete.*

The proof of Theorem 1.1.4 is presented in two stages. First, we will describe a combinatorial problem #RigidCircuit. In Lemma 2.1.3 we give a parsimonious reduction from #3SAT, which is #P-complete, to #RigidCircuits. Then, in Lemma 3.1.2, we use a more complicated set of reductions from #RigidCircuits to #Bruhat to show that #Bruhat is #P-complete.

Let us emphasize that the proof of Lemma 3.1.2 is *computer assisted*, i.e. it has gates found by computer, but which in principle can be checked directly. See §3.4.1 for a detailed discussion of computational aspects of the proof.

Remark 1.1.5. The proof in [BW91] uses a modulo p argument and the Chinese Remainder Theorem, which we also employ for our result (cf. §3.4.2). In fact, this approach is one of the few applicable for these problems, since the existence of FPRAS (see below) strongly suggests the impossibility of a parsimonious reduction of #3SAT and its relatives.

Brightwell and Winkler's proof that #LE is #P-complete [BW91] showed further that counting linear extensions for posets of height 3 is #P-complete. They conjectured that the following problem is #P-complete:

> #H2LE (*Number of linear extensions of height-2 posets*)
>
> **Input:** A partially ordered set P of height 2.
>
> **Output:** The number $\boldsymbol{e}(P)$ of linear extensions.

Here *height two* means that P has two levels, i.e. no chains of length 3. This problem has been open for 27 years, most recently reiterated in [Hub14, LS17]. We resolve it in Chapter 4.

Theorem 1.1.6. #H2LE *is* #P*-complete.*

Our next result in Chapter 4 is an extension of Theorem 1.1.6. It was proposed recently by Lee and Skipper in [LS17], motivated by the optimization of nonlinear functions over the much-studied correlation polytope (see e.g. [DL97, LSS18]).

#IPLE (*Number of linear extensions of incidence posets*)

Input: A graph $G = (V, E)$.

Output: The number $\boldsymbol{e}(I_G)$ of linear extensions of the incidence poset I_G.

Here the incidence poset I_G is defined as a height 2 posets with vertices V on one level, edges E on another level, and the inequalities defined by adjacencies in G.

Theorem 1.1.7. #IPLE *is* #P*-complete.*

Theorem 1.1.7 implies Theorem 1.1.6, of course. Formally, the proofs of both results are independent, but use the same technical ideas of using number theory to to obtain targeted reductions modulo primes. Since the proof Theorem 1.1.6 is both technically and conceptually simpler, we chose to include both proofs.

Historical review

The notion of #P*-completeness* was introduced by Valiant [Val79] a way to characterize the class of computationally hard counting problems; see [MM11, Pap94] for a modern treatment. The #LE problem is related to the problem counting order ideals in a poset, known to be #P-complete [PB83]. In contrast with the latter problem, #LE has FPRAS which allows $(1+\varepsilon)$-approximation of $\boldsymbol{e}(P)$, see e.g. [KK91, Mat91].

There are several classes of posets for which the counting is known to be polynomial: the dimension-2 posets given by Young diagrams of skew shape (see e.g. [MPP18, Sta97]), the *series-parallel posets* (also dimension 2, see [Möh89, §2.4]), a larger class of posets with *bounded decomposition diameter* [Möh89, §4.2], *sparse posets* [EGKO16, KHNK16], posets

whose covering graphs have disjoint cycles (see [Atk89]), and *N-free posets* with bounded width and spread [FM14].

The study of posets of a given dimension is an important area, and the dimension 2 is both the first interesting dimension and special due to the duality property. See monograph [Tro92] for a comprehensive treatment. Posets of dimension 2 have a sufficiently rigid combinatorial structure to make various computational problems tractable. For example, the decision problem whether a poset has dimension 2 is in P (see e.g. [Tro95]), as is the above mentioned problem of counting ideals of dimension-2 posets, see [Möh89, p. 163].

The *weak Bruhat order* is a fundamental object in both Representation Theory and Algebraic Combinatorics, well studied in much greater generality, see e.g. [BjB05, MS05, Sta97]. In fact, it plays a key role in several areas such as *Schubert calculus* (see e.g. [Mac91, Man01]), and *Kazhdan–Lusztig theory* (see e.g. [BjB05, Lus03]). In the context of bubble sorting, counting Bruhat order sizes is discussed by Knuth [Knu98, §5.2.1], among others. As we mentioned above, the connection between #D2LE and #Bruhat has been rediscovered a number of times in varying degree of generality, see [BjW91, FW97, Reu96].

The height-2 posets is an important and well studied class of posets. Brightwell and Winkler write: "We strongly suspect that Linear Extension Count for posets of height 2 is still #P-complete, but it seems that an entirely different construction is required to prove this" [BW91]. Incidence posets have also been studied quite intensely. We refer to recent papers [LS17, TW14] for an overview of the area and further references.

The *correlation polytopes* were introduced by Padberg [Pad89] and studied intensely in a number of papers (see e.g. [DL97, Pit91]). The connection to linear extensions of incidence posets was given in [LS17] based on [Sta86] and other related earlier work (see [LSS18] for a detailed overview).

Proof structure

Chapters 2 and 3 contain a highly technical proof of theorems 1.1.1 and 1.1.4. We begin with basic definition and notation in Section 1.2. In sections 2.1 and 2.2 we present the

construction. In Section 3.3 we gives proofs of technical lemmas. The list systems of algebraic equations defining parameters of the logical gates is in Section 3.5. We conclude this portion of the discussion with final remarks and open problems in Section 3.4.

1.2 Basic definitions and notation

1.2.1 Posets

We assume the reader is familiar with basic definitions on posets, see e.g. [Tro95] and [Sta97, Ch. 3]. We describe a *linear extension* of a poset $\mathcal{P} = (X, <)$ on a set X with n elements as an *assignment* of the values $\{1, 2, \ldots, n\}$ to X, or as a *labeling* of X by the values $\{1, 2, \ldots, n\}$.

Let $\ell : X \to \{1, 2, \ldots, n\}$ be a linear extension of $\mathcal{P}$, and let X be given a default ordering, say $X = \{x_1, \ldots, x_n\}$. Then the function $i \mapsto \ell(x_i)$ is a permutation in S_n. We call this the permutation *induced* by the linear extension.

1.2.2 Permutations

For the technical constructions in Section 3.2 we express all permutations in one-line notation, in other words as a sequence where the integers from 1 to n occur exactly once. For several of these constructions, we wish to generalize permutations by either omitting or repeating numbers. We can treat an arbitrary sequence of n integers as a permutation in S_n by relabeling the elements from 1 to n, from smallest to largest, and, when a number is repeated, from left to right. For example, we would relabel the sequence

$$(7, 7, 5, 3, 3, 5)$$

by replacing the two 3's with a 1 and a 2, the two 5's with a 3 and a 4, and the two 7's with a 5 and 6, giving the permutation

$$(5, 6, 3, 1, 2, 4).$$

We will describe this relabeling explicitly where it helps to clarify the presentation, and talk about *shifting* elements up or down.

We use the term *block* exclusively to refer to a sequence of consecutive integers in consecutive position, and write it by replacing the sequence with an integer encased in a box: $\boxed{3}$.

1.2.3 Other notation

We write $\mathbb{N} = \{0, 1, 2, \ldots\}$ for the set of nonnegative integers, and $\mathbb{F}_q$ to denote the finite field with q elements. Let $[n] = \{1, 2, \ldots, n\}$ and $\binom{[n]}{k}$ to denote k-subsets of $[n]$. To make our notation more readable, when writing vectors in $\mathbb{F}_q^d$, we omit parentheses and commas, so that $(0, 1)$ becomes 01.

We refer to [MM11, Pap94] for notation, basic definitions and results in computational complexity. We use ϕ for logical gates. We introduce a new notation $\phi \rtimes (v_1, v_2)$ to be a result of a certain operation corresponding to (v_1, v_2) applied to ϕ, see §2.2.2.

1.3 Contingency tables

1.3.1 Counting sparse contingency tables

Random generation and approximate counting contingency tables is a classical problem in statistics, discrete probability, combinatorics and theoretical computer science (see below). The MCMC algorithms have been introduced over 20 years ago and are known to work in case of *dense* tables (i.e. with large margins). The MCMC approach in this chapter is the first that provably works for *sparse* tables.

Formally, let $\mathbf{a} = (a_1, \ldots, a_m)$, $a_1 \geq \ldots \geq a_m > 0$, and $\mathbf{b} = (b_1, \ldots, b_n)$, $b_1 \geq \ldots \geq b_n > 0$, be two integer sequences with equal sum:

$$\sum_{i=1}^{m} a_i \;=\; \sum_{j=1}^{n} b_j \;=\; N.$$

A *contingency table* with *margins* $(\mathbf{a}, \mathbf{b})$ is an $m \times n$ matrix of non-negative integers whose i-th row sums to a_i and whose j-th column sums to b_j. We denote by $\mathcal{T}(\overline{a}, \overline{b})$ the set of all such matrices, and let $\mathrm{T}(\mathbf{a}, \mathbf{b}) := |\mathcal{T}(\overline{a}, \overline{b})|$.

In this chapter we give a new polynomial time algorithm for approximating $\mathrm{T}(\mathbf{a}, \mathbf{b})$ for

sparse matrices. More precisely, we prove FPRAS in the following two cases:

(1) for **s**mall margins $a_1, b_1 = O(n^{\alpha})$, $\alpha < 1/4$, and $m = \Theta(n)$,

(2) for **s**mooth margins $a_1, b_1 = O(n^{1-\epsilon})$, $\epsilon > 0$, s.t. $a_1/a_m, b_1/b_n = O(1)$, and $m = \Theta(n)$.

These results are new, conjectured for 20 years, and have been long sought in both theory and practice.

1.3.2 Brief summary of prior work

Since the literature is large and diverse, below we give a brief summary of several approaches to approximating $\mathrm{T}(\mathbf{a}, \mathbf{b})$. More details and precise references are given later in the introduction.

- Exact counting in polynomial time for bounded m and n, via Barvinok's algorithm.
- Exact counting in polynomial time for bounded a_1 or b_1 via dynamic programming.
- Exact counting is #P-complete even for $m = 2$ or $n = 2$.
- Approximate counting is *self-reducible*.
- Exact asymptotic formulas in the *uniform* case $a_1 = \ldots = a_m$, $b_1 = \ldots = b_n$.
- Estimates in the non-uniform case whose quality depends on a_1/a_m and b_1/b_n.
- Lower and upper bounds for *smooth margins*, i.e. for bounded $a_1/a_m, b_1/b_n$ and $m = \Theta(n)$.
- Quasi-poly time algorithm for approximate counting for $a_1/a_m, b_1/b_n < 1.6$ and $m = \Theta(n)$.
- MCMC based FPRAS for bounded height m or width n.
- MCMC based FPRAS for large margins $a_m = \Omega(n^{3/2} m \log m)$, $b_n = \Omega(m^{3/2} n \log n)$.

It is obvious but still worth mentioning, that even in the best studied uniform case the exact asymptotic estimates do not give FPRAS since the error terms depend on parameters

a_1, b_1, m and n.

Magic squares are $n \times n$ contingency tables with uniform margins equal to K. This is an especially attractive example because of applications to many different areas (see below). Our results give the **f**irst FPRAS for the number $\boldsymbol{t}(n, K)$ of such magic squares for $K = O(n^{1-\epsilon})$, $\epsilon > 0$. In contrast, the case of $K = \Omega(n^{2.5} \log n)$ was done earlier. This leaves unresolved the intermediate cases $K = n^{\alpha}$, $1 \le \alpha \le 2.5$, a curious open problem.

1.3.3 Complexity background

Counting contingency tables is #P-complete and feasibility of certain 3-dim contingency tables is NP-complete, see [DO04, DKM97, IJ94]. Note that the exact counting is conjectured to be strongly #P-complete (see [DO04, PP86]), but this requires the matrices to have unbounded dimensions $m, n = \omega(1)$.

Barvinok's algorithm [Bar93] allows the exact counting in polynomial time when the table dimensions m, n are fixed. For fixed margins $a_1, b_1 = O(1)$, the exact counting can be done by a dynamic programming (see e.g. [GM77]).

The first rigorous MCMC approach was given in [DG95] based on the Markov chain which chooses random 2×2 submatrix and performs the following one of the following two changes with equal probability (stay put if the change is impossible):

$$\begin{matrix} +1 & -1 \\ -1 & +1 \end{matrix} \qquad \text{or} \qquad \begin{matrix} -1 & +1 \\ +1 & -1 \end{matrix} \tag{1.3.1}$$

This *Diaconis–Gangolli Markov chain* was first rigorously studied by Diaconis and Saloff–Coste [DS95] for fixed row and column sums, but their bounds are exponential in mn. Dyer, Kannan and Mount [DKM97] gave a polynomial time bound is obtained for margins $a_m = \Omega(n^2 m)$ and $b_n = \Omega(m^2 n)$, see also [Mou95]. These bounds were later improved by Morris [Mor02] to $a_m = \Omega(n^{3/2} m \log m)$ and $b_n = \Omega(m^{3/2} n \log n)$. See also [CGY96] for a somewhat weaker earlier bound for a different but related chain.

When the number of rows m is bounded, a polynomial time algorithm was given by Cryan and Dyer [CD03] with no constraints on the margins. See also [C+06] for sharper bounds and a natural MCMC approach in the same setting.

The self-reducibility of approximate counting $\mathrm{T}(\mathbf{a}, \mathbf{b})$ is standard and described in a number of sources (see e.g. [DG95, DKM97]). This follows from the fact that the number of contingency tables $X = (x_{ij}) \in \mathcal{T}(\overline{a}, \overline{b})$ have entry $x_{11} > s$ is equal to the number of contingency tables $\mathrm{T}(\mathbf{a}, \mathbf{b})$ with $a_1 \leftarrow a_1 - s$, $b_1 \leftarrow b_1 - s$. As a consequence, a polynomial mixing time of a Markov chain with uniform distribution on a self-reducibility closed class of matrices implies the FPRAS for that class.

A completely different approach was developed by Barvinok, Luria, Samorodnitsky and Yong in [BLSY10]; they gave a quasi-polynomial algorithm for "smooth margins" s.t. a_1/a_m, $b_1/b_n < \phi$, where $\phi = (1 + \sqrt{5})/2$ is the *golden ratio*. This approach was later extended in [BH12] to an asymptotic bound in some cases.

1.3.4 Statistical background

The problem of sampling contingency tables is fundamental in statistics and applications to natural sciences, see e.g. [Eve92, FLL17, Kat14]. Both general and *binary* (0-1) tables are studied. The latter are somewhat easier to sample in cases of practical interest (see e.g. [DS98, MH13]).

Many authors at different times lamented the difficulty of sampling contingency tables from the uniform distribution, notably Diaconis and Efron [DE85] (see also [Kat14, MH13]). Instead, the *hypergeometric* (*Fisher – Yates*) *distribution* is commonly used as it is easy to sample (see e.g. [DG95, Kat14]), or a variety of ad hoc approaches (especially in the binary case). This is understood to be a major problem in the area. Notably, the authors in [MH13] present a case study of how a biased distribution used in [PA86] gave an apparently wrong conclusion (by several orders of magnitude).

There are various practical sampling algorithms in the literature with different levels of rigor and proven efficiency. Beside the MCMC approach discussed earlier, let us mention

the *sequential importance sampling* [CDHL05] (cf. [BSSV12]), the algebraic approach (see e.g. [DS98, DF03, Sul18+]), and various divide-and-conquer approaches (see e.g. [Mou95, DZ15+]).

1.3.5 Combinatorial background

Contingency tables naturally correspond to adjacency matrices of bipartite multigraphs; sampling such graphs with given degree vectors is important in combinatorics and network analysis (see e.g. [DG95, Wor18]). Exact asymptotic formulas are known in a few cases, notably for general and binary relatively small degrees (see e.g. [Bar09, BC78, GM13]). Note that [BBK72] requires bounded row and column sums, while the more recent [GM13] requires $a_1 b_1 = o(N^{2/3})$. General lower and upper bounds are given in [Bar10a, Bar12]; note that these bounds are off by an exponential factor.

Contingency tables are also integer points of the *transportation polytopes* which play an important role in combinatorial optimization (see e.g. [DK14]). They are a special case (for a bipartite graph) of a more general *integer network flow* problem, see e.g. [BDV04, Bar09, CDR10].

For $m = n$ and equal margins (the uniform case) the transportation polytope is the (scaled) *Birkhoff polytope* $\mathrm{B}_n \subset \mathbb{R}^{n^2}$ of bistochastic matrices. This polytope is of interest in Combinatorics, Optimization, Probability and other areas (see e.g. [DK14, DG04, Pak00]). The integer points in $K \cdot \mathrm{B}_n$ are called *magic squares*; they are $n \times n$ contingency tables with row and column sums equal to K. The number $\boldsymbol{t}(n, K)$ of such magic squares is the evaluation of the *Ehrhart polynomial*, intensely studied both combinatorially, empirically and asymptotically, see [BP03, CM09, CV16]. It is not known, e.g., if $\big\{\boldsymbol{t}(n, n)\big\}$ or $\big\{\mathrm{v}ol(\mathrm{B}_n)\big\}$ can be computed in $\mathrm{poly}(n)$ time; see [Pak18] for the context on computability of sequences.

1.3.6 Random graphs and contingency tables

There is a thematic connection between questions about contingency tables and questions about random graph. For sparse simple graphs the classical questions going back to Erdős and Rényi, intensely studied in the last several decades [Bo01, 4] have been: How many graphs are there with given degrees? What do random graphs look like? How do their properties change when the parameters change? The case of fixed degrees is fundamental in the network and internet sciences, as large scale real-world networks have a power-law degree distribution [Ba99, 1, 5]. The asymptotics, MCMC random generation and approximate counting are obtained under various degree constraints [Wor18], [3, §11].

Binary (0-1) and general contingency tables are adjacency matrices of bipartite graphs and multi-graphs, respectively; they are also standard models in network theory [2]. Like sparse simple graphs with fixed degrees, sparse contingency tables with sublinear $o(n)$ margins have easier structure when it comes to asymptotic counting [GM13] and random generation [DP19+d]. For the *linear margins* $\Theta(n)$, the existing techniques for the number of contingency tables either do not apply, or produce bounds which are off by an exponential factor [Bar12, BLSY10].

This background motivates Chapter 6. We explore random models for contingency tables and demonstrate in Theorem 6.4.2 the existence of a phase transition in individual entry distribution under certain assumptions of weak independence.

CHAPTER 2

$\oplus$D2LE is $\oplus$P-complete

2.1 The setup for #D2LE, $\oplus$D2LE, #Bruhat and $\oplus$Bruhat

2.1.1 #D2LE and $\oplus$D2LE

This chapter is devoted to a proof of Theorem 1.1.2. There is a substantial area of overlap between this proof and the proof of Theorem 1.1.1. Both rely on the equivalence between #Bruhat and #D2LE, and both use the same notion of rigid circuits. Both proofs also encode rigid circuits in permutations in an almost identical fashion. One unfortunate difference is that the permutations representing True and False in the two constructions are opposite of each other.

Other noteworthy differences are in the technical constructions to initialize and test wires (§ 2.2.4 vs § 3.2.3) and the construction of parameterized gates introduced in § 3.2.4. We will defer the proof of some of the more technical lemmas to the next chapter whenever translating the general argument mod p to the mod 2 case would be unenlightening.

2.1.2 Linear extensions and the Bruhat order

We begin with a known result that #D2LE is equivalent to #Bruhat.

Lemma 2.1.1 ([FW97])**.** *For every $\sigma \in S_n$, there exists a poset P_σ of dimension two with n elements such that $\boldsymbol{e}(P_\sigma) = \boldsymbol{e}(\sigma)$. Conversely, for every poset P of dimension two with n elements, there exists $\sigma \in S_n$ such that $\boldsymbol{e}(P) = \boldsymbol{e}(\sigma)$.*

Proof. Given a permutation $\sigma \in S_n$, we form a poset P_σ of dimension 2 by taking the

points $p_i = (i, \sigma^{-1}(i)) \in \mathbb{R}^2$, with the standard product ordering. A linear extension of P_σ is a function from the p_i's to $\{1, 2, \ldots, n\}$, which induces a permutation τ as described in Section 1.2. Then τ is a linear extension of P_σ if and only if $\tau(i) < \tau(j)$ whenever $i < j$ and $\sigma^{-1}(i) < \sigma^{-1}(j)$. When this holds, for $\omega = \tau^{-1}$ we have $\omega \leq \sigma$ in the weak Bruhat order, so that $\boldsymbol{e}(P_\sigma) = \boldsymbol{e}(\sigma)$.

Conversely, given a poset P of dimension two, it can be represented as a collection of points $p_i \in \mathbb{R}^2$ with the product ordering. We translate the points of p_i so that they are all in the first quadrant, and then, for some sufficiently small $\varepsilon > 0$, perform the affine transformation:

$$p_i \mapsto \begin{pmatrix} 1 & \varepsilon \\ \\ \varepsilon & 1 \end{pmatrix} p_i.$$

This transformation ensures that no two points are in the same row or column without changing the ordering on P. Label the points from 1 to n, reading from left to right, and replace the x-coordinates with these labels. Similarly, replace the y-coordinates with the labels 1 through n, read from bottom to top. The points now represent the poset P_σ, for some $\sigma \in S_n$. We thus have $\boldsymbol{e}(P) = \boldsymbol{e}(P_\sigma) = \boldsymbol{e}(\sigma)$. □

2.1.3 Rigid circuits

In this subsection, we define rigid circuits, which will be the principal gadget in this proof and the proof of Theorem 1.1.4. Visually, a rigid circuit consists of a collection of wires laid out in the plane. The wires run horizontally, from left to right. They carry a binary signal, with a 1 representing TRUE, and a 0 representing FALSE. Adjacent wires can feed into logic gates, where they interact in some way; wires can only cross by using a certain logic gate called a SWAP gate[1].

At the far left of the picture, the wires represent binary inputs. The bottom wire at the

[1]Note that the construction we give here is a little different from the usual definition of Boolean circuits (see e.g. [Vol99]). In particular, the TESTEQ and TESTNEQ gates are relations but not functions. We call our circuits *rigid* to emphasize that wires are ordered and only adjacent wires can interact.

far right is the output wire. The circuit is satisfied by a choice of inputs if the output wire reads TRUE. Formally, we give the following definitions:

A *circuit state* with k wires is a vector $v \in \mathbb{F}_2^k$. A *general rigid circuit* with m circuit states[2] and k wires is a sequence of m circuit states $(v_1, \ldots, v_m)$, each with k wires, together with a list of relations $(L_1, \ldots, L_{m-1})$ on $\mathbb{F}_2^k$, such that $(v_i, v_{i+1}) \in L_i$, for $1 \leq i \leq m-1$. The relations L_i we call *logic gates*.

We next define *specialized rigid circuits*, which are the circuits we will use throughout the paper, by restricting our choice of logic gates. We define five *simple logic gates* as follows.

IDENTITY gate L_1: The identity function from $\mathbb{F}_2 \to \mathbb{F}_2$.

SWAP gate L_2: A function from $\mathbb{F}_2^2 \to \mathbb{F}_2^2$ that sends $ab \to ba$, for $a, b \in \mathbb{F}_2$.

ANDOR gate L_3: A function from $\mathbb{F}_2^2 \to \mathbb{F}_2^2$ that sends $ab \to (a \text{ AND } b)(a \text{ OR } b)$,

where AND and OR are bitwise operations, for $a, b \in \mathbb{F}_2$.

TESTEQ gate L_4: A relation on $\mathbb{F}_2^2$ that contains $\{(11, 11), (00, 00)\}$.

TESTNEQ gate L_5: A relation on $\mathbb{F}_2^2$ that contains $\{(10, 10), (01, 01)\}$.

Note that the TESTEQ gate merely copies the signal when both wires share the same truth value. If the wires contain different truth values, there is no acceptable next circuit state. In this case, we say the circuit *shorts out*. Likewise the TESTNEQ gate copies the signal when two two wires have different truth values, and otherwise shorts out.

For #D2LE we use gates L_1 through L_4, while for $\oplus$D2LE we use gates L_1, L_2, L_3 and L_5. This difference between the two constructions is a consequence of the relative ease of constructing a TESTEQ gate versus a TESTNEQ gate in the two settings. We note that, in the usual language of computer science, gates L_1, L_2, L_3 and L_5 are functionally complete because the TESTNEQ gate could be used as a NOT gate, while gates L_1 through L_4 are not complete. We address this by grouping the input wires into pairs and requiring exactly one wire in each pair to be TRUE. This construction plays the role of a NOT gate. We state this requirement formally below and prove a notion of completeness in Lemma 2.1.3.

[2]In the usual language of circuit complexity, a circuit with m circuit states has depth $m-1$.

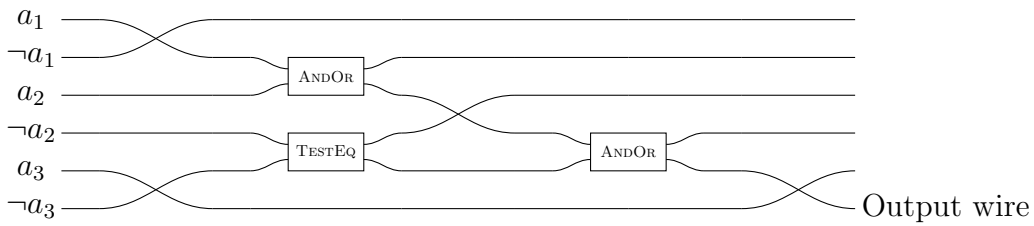


Figure 2.1: A specialized rigid circuit C with $\boldsymbol{e}(C) = 4$. We force $\neg a_2 = \neg a_3$, and the output wire carries the value of the clause $(a_1 \vee a_2 \vee \neg a_2)$.

We also observe that if we instead adopted the convention that a 1 represented False, and a 0 represented True, each gate would have the same behavior except for L_3, which would become an OrAnd gate. This flexibility is part of our motivation for using this set of gates.

Let v and v' be circuit states with k and k' wires, respectively. We define the *coupling* of v and v', which we write as $v \wedge v'$, by concatenating the entries of v and v' to give a circuit state with $k + k'$ wires. Let L and L' be logic gates on k and k' wires, respectively. We define the *coupling* of L and L', which we write as $L \wedge L'$, to be the relation on $\mathbb{F}_2^{k+k'}$ where $(v_1 \wedge v_1', v_2 \wedge v_2') \in L \wedge L'$ precisely when $(v_1, v_2) \in L$ and $(v_1', v_2') \in L'$. A *compound logic gate* is a logic gate made by coupling together copies of the four simple logic gates.

Note that a compound logic gate on (v_i, v_{i+1}) determines v_{i+1} from v_i as long as the circuit does not short out. In our construction, it is sufficient to use compound logic gates where all but one of the gates coupled together are Identity gates. By abuse of notation, we still generally refer to such a compound logic gate by the one simple gate in the coupling that is not an Identity gate. So, for example, a compound logic gate that swaps the wires in positions i and $i+1$ and otherwise is made up of Identity gates we will call a Swap gate.

A *specialized rigid circuit* is a general rigid circuit with m circuit states and $2k$ wires, such that each logic gate is a compound logic gate and the initial circuit state $v_1 = (a_1, \ldots, a_{2k})$ has exactly one of each pair a_{2i-1}, a_{2i} set to true. We therefore relabel the entries of v_1 as $(a_1, \neg a_1, \ldots, a_k, \neg a_k)$, where $\neg$ denotes bitwise not. A *satisfying assignment* of a circuit C is a choice of v_1 such that the circuit does not short out and the last term of v_m is set to

TRUE.

We refer to circuits by the capital letter C, and call the number of satisfying assignments $\boldsymbol{e}(C)$. We can now state the following:

$\oplus$RIGIDCIRCUIT

Input: A specialized rigid circuit C.

Output: The parity of the number $\boldsymbol{e}(C)$ of satisfying assignments of C.

Throughout this paper, we will refer to specialized rigid circuits simply as rigid circuits or as circuits when our meaning is clear. Before moving on, we observe the following:

Lemma 2.1.2. *For every rigid circuit C with a satisfying assignment v_1, there will be exactly k wires set to* TRUE *in each circuit tate.*

Proof. There must be k wires set to TRUE in v_1. We note that none of the five simple gates can change the number of TRUE wires, which completes the proof. □

Our reduction in both chapters relies on the following.

Lemma 2.1.3. *There is a parsimonious reduction from* #3SAT *to* #RIGIDCIRCUIT.

Proof. Let I be an instance of #3SAT with u variables and v clauses. We form a rigid circuit with $6v$ wires, so that there is one pair of wires for each time a variable or its negation appears in a clause.

We label these $6v$ wires as $(a_1, \neg a_1, a_2, \neg a_2, \ldots, a_{3v}, \neg a_{3v})$. We want some of these wires to represent multiple instances of the same variable. To force $a_i = a_j$, we can use SWAP gates to move a_i and a_j next to each other, and then run them through a TESTEQ gate. Equivalently, we can move a_i and $\neg a_j$ next to each other and then run them through a TESTNEQ gate. The circuit will then short out unless both $a_i = a_j$ and $\neg a_i = \neg a_j$.

We then use SWAP gates to re-arrange the variables so that the order of variables in the first $3v$ wires match the clauses of I. We use two ANDOR gates on each clause to produce the desired disjunctions. At this point in the construction, the $3i$-th wire carries the value of

the i-th clause, for i between 1 and k. Now, we use more SWAP gates to move these k wires to the far left of the circuit state, and use $(k-1)$ ANDOR gates to compute the conjunction of all of the clauses, which ends up in the first wire. Finally, we swap the first wire into the last position of our circuit state.

It takes $O(v)$ uses of SWAP gates to move any two wires adjacent to each other, so this entire process requires $m = O(v^2)$ circuit states. □

In Section 2.2 we will prove the following:

Main Lemma 2.1.4. *For every rigid circuit C with m circuit states and $2k$ wires, $k > 7$, there is $n = O(mk^{10})$, and $\sigma \in S_n$, such that $\boldsymbol{e}(C) \equiv \boldsymbol{e}(\sigma) \mod 2$.*

We then have:

Proof of Theorem 1.1.2. We construct a polynomial time reduction from #3SAT to $\oplus$D2LE. Given a problem in #3SAT, we first apply Lemma 2.1.3 to obtain a rigid circuit C with m circuit states and $2k$ wires. We next apply Lemma 2.1.4 to find some choice of n and $\sigma \in S_n$ with $\boldsymbol{e}(C) \equiv \boldsymbol{e}(\sigma) \mod 2$. Then, by Lemma 2.1.1, we can compute $\oplus$D2LE, as desired. □

2.2 Circuit constructions

2.2.1 Bruhat circuits

To prove Lemma 2.1.4, we produce a permutation σ that emulates the design in §2.1.3. We build a Bruhat circuit, with Bruhat circuit states, simple Bruhat logic gates, and compound Bruhat logic gates.

We need to modify our circuits as follows. Whenever a TESTEQ or ANDOR gate acts on a pair of wires, we use SWAP gates to move those wires to the first two positions of the circuit state vector. We perform the desired TESTEQ or ANDOR operation, and then use SWAP gates to put the wires back in their previous positions. We make this modification because of the technical requirement of Lemma 2.2.3.

A *Bruhat circuit framework* is a permutation $\sigma \in S_n$ together with a classification of the elements in $\{1, 2, \ldots, n\}$ into one of three categories.

The *separators* are a list of elements $s_1 < s_2 < \cdots < s_m$ with $\sigma^{-1}(s_1) < \cdots < \sigma^{-1}(s_m)$. By convention we let $s_0 = \sigma^{-1}(s_0) = 0$ and $s_{m+1} = \sigma^{-1}(s_{m+1}) = n + 1$ where needed. For each remaining element x, there is some i, with $0 \leq i \leq m$, such that

$$\sigma^{-1}(s_i) < \sigma^{-1}(x) < \sigma^{-1}(s_{i+1}).$$

We require either

$$s_i < x < s_{i+1},$$

in which case we call x a *stable element*, or

$$s_{i-1} < x < s_i,$$

in which case we call x a *variable*.

We label the N variables satisfying $\sigma^{-1}(s_i) < \sigma^{-1}(x) < \sigma^{-1}(s_{i+1})$ as $x_{i1} > x_{i2} > \cdots > x_{iN}$. We require further that $\sigma^{-1}(x_{i1}) < \sigma^{-1}(x_{i2}) < \cdots < \sigma^{-1}(x_{iN})$.

We now make the following essential observations. Let $\tau \in S_n$ be chosen with $\tau \leq \sigma$. Let x be a stable element and x_{ij} be a variable satisfying

$$\sigma^{-1}(s_i) < \sigma^{-1}(x), \sigma^{-1}(x_{ij}) < \sigma^{-1}(s_{i+1}).$$

Then:

$$\tau^{-1}(s_1) < \cdots < \tau^{-1}(s_m) \qquad \text{and} \qquad \tau^{-1}(s_i) < \tau^{-1}(x) < \tau^{-1}(s_{i+1}),$$

and either

$$\tau^{-1}(s_i) < \tau^{-1}(x_{ij}) < \tau^{-1}(s_{i+1}) \qquad \text{or} \qquad \tau^{-1}(s_{i-1}) < \tau^{-1}(x_{ij}) < \tau^{-1}(s_i).$$

Given a Bruhat circuit framework σ and some $\tau \leq \sigma$, for $1 \leq i \leq m$ we assign to τ a *Bruhat circuit state* $v_i \in \mathbb{F}_2^N$ as follows. Write $v_i = (a_{i1}, a_{i2}, \ldots, a_{iN})$, with $a_{ij} \in \mathbb{F}_2$. Then take $a_{ij} = 0$ if $\tau^{-1}(s_i) < \tau^{-1}(x_{ij}) < \tau^{-1}(s_{i+1})$, and $a_{ij} = 1$ otherwise. Note that we will adopt the opposite convention in the next chapter.

Since the y_{ij}'s are arranged in strictly decreasing order, they can be re-arranged arbitrarily in τ, so that every possible circuit state can be realized as a Bruhat circuit state. In particular, for every possible circuit assignment $(v_1, \ldots, v_m)$, there is a unique permutation τ with Bruhat circuit state equal to $(v_1, \ldots, v_m)$ maximal in the Bruhat order. It is obtained by moving each variable which takes the value FALSE in circuit state v_i immediately to the left of s_i, keeping those FALSE variables in descending order. Call this permutation $\tau|v_1, \ldots, v_m$.

In summary, we have:

$$\boldsymbol{e}(\sigma) = \sum_{(v_1,\ldots,v_m)} \boldsymbol{e}(\tau|v_1, \ldots, v_m), \tag{2.2.1}$$

where the sum is taken over every possible set of circuit states $(v_1, \ldots, v_m)$. In the next four subsections, we will show how to control the value of $\boldsymbol{e}(\tau|v_1, \ldots, v_m)$ to encode the logic of our circuit.

2.2.2 Bruhat logic gates

A *Bruhat logic gate* with k wires is a sequence ϕ of distinct integers such that the smallest k terms are in decreasing order, the last k terms are in decreasing order, and these two sets do not overlap. We refer to these elements as the input and output variables, respectively, of the logic gate.

For technical reasons, we also require that immediately preceding the last k terms is a stable element less than the last k terms. We refer to this element as the *penultimate element*.

If $|\phi| = \ell$, we do not require ϕ to take values strictly in the set $\{1, \ldots, \ell\}$, but we still treat ϕ as a member of S_ℓ, as described in Section 1.2.

The *evaluation* of a Bruhat logic gate ϕ with k wires at some pair of circuit states $(v_1, v_2) \in \mathbb{F}_2^k$ is given by deleting from ϕ each of the input variables corresponding to a 0 in v_1 and each of the output variables corresponding to a 1 in v_2. We write this as $\phi \rtimes (v_1, v_2)$.

Given a Bruhat circuit framework σ, we write down a collection of sequences $(\sigma_1, \ldots, \sigma_{m+1})$ as follows. For σ_i, write down all the elements of σ (taken in one line notation) between s_{i-1}

and s_i, and then write down only the variables that occur between s_i and s_{i+1}.

Note that, for all i satisfying $2 \leq i \leq m$, the sequence σ_i is a Bruhat logic gate with N wires. Also note that the choice of $(\sigma_1, \ldots, \sigma_{m+1})$ determines our original Bruhat circuit framework σ uniquely.

For a given set of circuit states $(v_1, \ldots, v_m)$, we similarly define the sequence $(\tau_1, \ldots, \tau_{m+1})$, by writing $\tau | v_1, \ldots, v_m$ in one-line notation and breaking it apart at each separator s_i. Note that $\tau_i = \sigma_i \rtimes (v_{i-1}, v_i)$, for $2 \leq i \leq m$. By abuse of notation, we set $v_0 = v_{m+1} = \varnothing$, and let $\sigma_1 \rtimes (v_0, v_1) = \tau_1$ and $\sigma_{m+1} \rtimes (v_m, v_{m+1}) = \tau_{m+1}$.

Though the sequences τ_i are not permutations, we can treat them as permutations as described in Section 1.2, and so compute $\boldsymbol{e}(\tau_i)$. We can now rewrite (2.2.1) as

$$\boldsymbol{e}(\sigma) = \sum_{(v_1,\ldots,v_m)} \prod_{i=1}^{m+1} \boldsymbol{e}(\tau_i) \;=\; \sum_{(v_1,\ldots,v_m)} \prod_{i=1}^{m+1} \boldsymbol{e}\big(\sigma_i \rtimes (v_{i-1}, v_i)\big), \tag{2.2.2}$$

where the sum is taken over every possible set of circuit states $(v_1, \ldots, v_m)$.

We must have this product take the value 0 modulo 2 whenever $(v_1, \ldots, v_m)$ is not a satisfying assignment of C, and to take the value 1 otherwise.

As with rigid circuits, we say that a Bruhat circuit ϕ *shorts out* at v if, for every choice of v', we have

$$\boldsymbol{e}\big(\phi \rtimes (v, v')\big) \equiv 0 \mod 2.$$

2.2.3 Bruhat compound logic gates

In this subsection, we explain how to couple two Bruhat logic gates ϕ with k wires and ϕ' with k' wires to produce a new Bruhat logic gate $\phi \wedge \phi'$ with $k + k'$ wires, emulating the behavior of coupled logic gates defined in §2.1.3. First, we give a technical construction. Given a Bruhat logic gate ϕ with k wires, we say ϕ is *parity balanced* if $|\phi| - k \equiv 0 \mod 2$. To construct the *restriction* of ϕ, which we write $\phi_\circ$, we append to the beginning of ϕ the element $\max(\phi) + 1$. We have:

Lemma 2.2.1. *For a Bruhat logic gate ϕ that is parity balanced, we have*

$$\boldsymbol{e}\left(\phi_\circ \rtimes (v, v')\right) \equiv 0 \mod 2$$

when v and v' do not have the same parity of the number of wires carrying the value TRUE. *When v and v' do have an equal number of wires carrying the value* TRUE, *we have*

$$\boldsymbol{e}\left(\phi_\circ \rtimes (v, v')\right) \equiv \boldsymbol{e}\left(\phi \rtimes (v, v')\right) \mod 2.$$

Proof. The leading element $\max(\phi) + 1$ in $\phi_\circ$ can be rearranged freely, so we have

$$\boldsymbol{e}\left(\phi_\circ \rtimes (v, v')\right) \equiv \left(|\phi| - k + 1 + |v| - |v'|\right) \boldsymbol{e}\left(\phi \rtimes (v, v')\right) \mod 2.$$

The desired result follows immediately. □

We restrict to the case where ϕ is one of our simple Bruhat gates, and where ϕ' is a compound gate made up of IDENTITY and SWAP gates, since the construction in §2.2.1 requires us to place every TESTNEQ or ANDOR gate at the top of our circuit. We require further that ϕ and ϕ' be parity balanced.

The construction of $\phi \wedge \phi'$ involves inserting $\phi'_\circ$ in place of the penultimate element of ϕ, shifting elements appropriately. To explain these shifts, we replace each of the elements of ϕ and $\phi'_\circ$ with ordered pairs of integers.

Let y be the penultimate element of ϕ. Replace each of the elements x in ϕ with the ordered pair $(x, 0)$. Replace the input variables x of $\phi'_\circ$ with $(0, x)$, and all other elements x of $\phi'_\circ$ with (y, x). Then delete the penultimate block of ϕ and insert the relabeled $\phi'_\circ$ in its place.

Now relabel the entries from 1 to $|\phi| + |\phi'|$, going in order from smallest to largest with respect to the lexicographical order on $\mathbb{Z}^2$. Call the result $\phi \wedge \phi'$. Note that $\phi \wedge \phi'$ is a Bruhat logic gate with $k + 1$ wires when ϕ is the IDENTITY gate, and $k + 2$ wires otherwise, and that $\phi \wedge \phi'$ is parity balanced.

We define the following operations on logic gates:

Definition 2.2.2. *Left insertion*, *middle insertion* and *right insertion*, denoted $L(\phi)$, $M(\phi)$ and $R(\phi)$, respectively, are operators on Bruhat logic gates defined as follows. The terms left, middle and right are all oriented with respect to the penultimate block. Left insertion inserts the element 1 into ϕ immediately to the left of the penultimate block, and shifts all other elements up by 1. Middle insertion increases the length of the penultimate block by 1. Right insertion inserts an element one larger than the largest element in the penultimate block to the very end of ϕ, and shifts all larger elements up by 1.

Also, let $M^{-1}(\phi)$ denote the inverse operation to M where we decrease the length of the penultimate block by 1. Note that we will not use middle insertion until the following chapter.

The following lemma gives the set of conditions required for the coupling of logic gates to behave as desired.

Lemma 2.2.3. *Given ϕ and ϕ' as above, if ϕ is not the identity gate, $(\phi \wedge \phi')_\circ$ behaves as the coupling of the logic gates associated to $\phi_\circ$ and $\phi'_\circ$ when the following six equations are satisfied:*

$$|\phi| \equiv 0 \mod 2, \tag{1}$$

$$\boldsymbol{e}\big(L(\phi_\circ) \rtimes (10,11)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (10,11)\big) \equiv 0 \mod 2, \tag{2}$$

$$\boldsymbol{e}\big(L(\phi_\circ) \rtimes (01,11)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (01,11)\big) \equiv 0 \mod 2, \tag{3}$$

$$\boldsymbol{e}\big(L(\phi_\circ) \rtimes (00,01)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (00,01)\big) \equiv 0 \mod 2, \tag{4}$$

$$\boldsymbol{e}\big(L(\phi_\circ) \rtimes (00,10)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (00,10)\big) \equiv 0 \mod 2, \tag{5}$$

When ϕ is the identity gate, we need two equations to be satisfied:

$$|\phi| - 1 \equiv 0 \mod 2, \tag{7}$$

$$\boldsymbol{e}\big(L(\phi_\circ) \rtimes (0,1)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (0,1)\big) \equiv 0 \mod 2. \tag{8}$$

Proof. This is immediate from the more general Lemma 3.2.2. Note that, while the statement

of Lemma 2.2.1 is slightly weaker than that of Lemma 3.2.1, the fact that we are working mod 2 means we do not have to be concerned about $\ell(\tau') > 1$ or $r(\tau') > 1$ in § 3.3.2. □

2.2.4 Initializing and testing wires

We now give explicitly the construction of σ_1 and σ_{m+1}, to initialize wires at the beginning of our circuit and test the value of the output wire at the end.

For σ_1, we begin with ψ, a compound Bruhat logic gate consisting of $N/2$ copies of the identity wire, and then take $\sigma_1 = \psi \rtimes (\vec{0}, \vec{0})$, where $\vec{0}$ represents a circuit state with all wires set to FALSE. The identity gate construction given in Lemma 3.2.6 is simple enough that we can state what σ_1 looks like explicitly. It contains a sequence of $N/2 + 1$ terms, followed by the variables. The blocks themselves decrease by 2 with each block step, and the variables fill in the missing terms.

For example, when $N = 4$, σ_1 has 2 wires, and we have

$$\sigma_1 = 5\ 3\ 1\ 4\ 2.$$

We then modify σ_1 by duplicating each of the $N/2$ variables at the end, in their respective positions. Our example above becomes

$$\sigma_1 = 5\ 3\ 1\ 4\ 4\ 2\ 2.$$

Finally, we shift all elements of σ_1 up so that all of the elements are distinct and the final sequence is in strictly decreasing order. Our example now reads

$$\sigma_1 = 7\ 4\ 1\ 6\ 5\ 3\ 2.$$

Note that σ_1 now has N output wires, as required.

Recall that, by the abuse of notation introduced in §2.2.2, for a vector

$$v = (a_1, \ldots, a_N) \in \mathbb{F}_2^N,$$

we let $\sigma_1(v_0, v)$ represent σ_1 after deleting every element corresponding to a 1 in v, where we take $v_0 = \varnothing$.

Having given the details of our construction, we now prove the following:

Lemma 2.2.4. *We have* $\sigma_1 \rtimes (v_0, v) \equiv 1 \mod 2$ *precisely when exactly one of each pair* $\{a_{2i-1}, a_{2i}\}$ *is equal to* 1*, and* $\sigma_1 \rtimes (v_0, v) \equiv 0 \mod 2$ *otherwise.*

Proof. By construction, when exactly one of each pair $\{a_{2i-1}, a_{2i}\}$ is equal to 1, then $\sigma_1 \rtimes (v_0, v) = \psi_\circ \rtimes (\vec{0}, \vec{0})$, so $\boldsymbol{e}(\sigma_1(v)) = \boldsymbol{e}(\psi_\circ \rtimes (\vec{0}, \vec{0})) \equiv 1 \mod 2$. If the condition of this lemma is false. In this case, there must be some i for which both a_{2i-1} and a_{2i} are equal to 1 or both equal to 0. But then σ_1 will contain either two adjacent stable elements or two adjacent variables without any intervening term. The rearrangement of these terms are therefore independent of the rearrangement of the rest of σ_1, so that $\boldsymbol{e}(\sigma_1 \rtimes (v_0, v))$ is even, as desired. □

Rather than construct σ_{m+1} in a single step, we introduce a new gate:

XOR gate L_6: A function from $\mathbb{F}_2^2 \to \mathbb{F}_2$ that sends $ab \to (a \text{ XOR } b)$,

where XOR is the exclusive-or bitwise operation, for $a, b \in \mathbb{F}_2$.

Since we are only interested in the output wire, we can use XOR gates repeatedly to remove all but two wires: the desired output wire and another wire. Since no gate can change the parity of the number of TRUE wires, we now know the value that both wires should carry.

We can test those values by taking either

$$\sigma_{m+1} = 7\ 2\ 5\ 1\ 3\ 6\ 4,$$

or

$$\sigma_{m+1} = 7\ 2\ 5\ 1\ 3\ 4,$$

depending on the parity of k, the number of pairs of wires in our construction. We can verify directly that these have the desired properties, or note that σ_{m+1} is two copies of the identity wire coupled together.

Lemma 2.2.5. *There are* IDENTITY, SWAP, ANDOR, TESTNEQ *and* XOR *gates that can act on the top wires and match the conditions of Lemma 3.2.2.*

Proof. By the preceding argument and by construction. See Lemma 2.2.3. We enclose the penultimate element in a box for readability. The computer code to generate these gates was a much simpler version of the code described in § 3.4.1.

1. IDENTITY gate:

$$\phi = 1\ \boxed{2}\ \ 3.$$

2. SWAP gate:

$$11\ 2\ 7\ 3\ 6\ 9\ 1\ 10\ \boxed{4}\ 8\ 5$$

.

3. ANDOR gate:

$$11\ 2\ 7\ 4\ 1\ 6\ 9\ 10\ \boxed{3}\ 8\ 5.$$

4. XOR gate:

$$10\ 2\ 3\ 6\ 8\ 9\ 1\ 7\ \boxed{4}\ 5$$

.

5. TESTNEQ gate:

$$11\ 2\ 4\ 1\ 7\ 10\ 3\ 8\ \boxed{5}\ 9\ 6$$

.

□

CHAPTER 3

Linear extensions of dimension two posets

3.1 Primes and circuits in the Bruhat order

The proof of Theorem 1.1.1 employs the same general constructions as the proof of Theorem 1.1.2, with additional construction. There is no parsimonious reduction from #RIGIDCIRCUIT to #BRUHAT, because there exist rigid circuits with no satisfying assignments, but every element $\sigma \in S_n$ has $\boldsymbol{e}(\sigma) \geq 1$. Instead, we will use a collection of permutations σ that allow us to compute the residue of #RIGIDCIRCUIT modulo enough primes that we can then use the Chinese Remainder Theorem to compute #RIGIDCIRCUIT.

We need the following number theory result:

Proposition 3.1.1 (see e.g. [BW91, p. 4])**.** *For $k \geq 4$, the product of primes between k and k^2 is at least $2^k k!$.*

In Section 3.2 we will prove the following:

Main Lemma 3.1.2. *For every rigid circuit C with m circuit states and $2k$ wires, $k > 7$, and every prime p between k and k^2, there is $n = O(mk^{10})$, and $\sigma \in S_n$, such that $\boldsymbol{e}(C) \equiv -\boldsymbol{e}(\sigma)$ mod p.*

Proof of Theorem 1.1.4. We construct a polynomial time reduction from #3SAT to #BRUHAT. Given a problem in #3SAT, we first apply Lemma 2.1.3 to obtain a rigid circuit C with m circuit states and $2k$ wires. We next apply Lemma 3.1.2 to find, for each prime p between k and k^2, some choice of n and $\sigma \in S_n$ with $\boldsymbol{e}(C) \equiv -\boldsymbol{e}(\sigma) \mod p$. Then, as in [BW91], we use the Chinese Remainder Theorem to compute the residue of $\boldsymbol{e}(C)$ modulo the product of primes between k and k^2.

Since there are at most 2^k satisfying assignments of a particular rigid circuit, applying Proposition 3.1.1 completes the proof. □

3.2 Circuit constructions

3.2.1 Mod-p parallel Bruhat circuits

We must further modify our circuits from § 2.2.1. Let C be a specialized rigid circuit with $2k$ wires and m circuit states. First, we add $(p-k)$ pairs of wires and use TestEq gates to set the value of $a_{k+1}, a_{k+2}, \ldots, a_p$ equal to the initial value of a_1. We then stack $(p-1)$ copies of this modified circuit together, and use TestEq gates to ensure that each copy of the circuit will have the same initial assignment as every other copy.

As in § 2.2.1, whenever a TestEq or AndOr gate acts on a pair of wires (in any copy of the circuit), we use Swap gates to move those wires to the first two positions of the circuit state vector. We perform the desired TestEq or AndOr operation, and then use Swap gates to put the wires back in their previous positions.

Finally, we use Swap gates to bring the last wire of each circuit copy into the final $(p-1)$ positions of our final circuit state. We write C_p for the resulting circuit, and call it a *mod-p parallel circuit*.

The motivation for these modifications comes later, in the technical requirements of Lemma 3.2.2 and the constructions in §3.2.3 and §3.2.5. For now, though, we note that $\boldsymbol{e}(C) = \boldsymbol{e}(C_p)$, and, by Lemma 2.1.2, in every valid circuit assignment, each circuit state of C_p will contain exactly $(p^2 - p)$ wires set to TRUE.

We use Bruhat circuit frameworks as described in §2.2.1. We now require that there be exactly

$$N = 2p^2 - 2p$$

variables.

Note that we take $a_{ij} = 0$ if $\tau^{-1}(s_i) < \tau^{-1}(x_{ij}) < \tau^{-1}(s_{i+1})$, and $a_{ij} = 1$ otherwise in this chapter.

For technical reasons, we also require that immediately preceding the last k terms is a block of $(p^3 - 1)$ consecutive elements, all less than the last k terms. We refer to this block, appropriately, as the *penultimate block*.

Since we are now working modulo p for some unknown prime p, we must have the product (2.2.2) take the value 0 modulo p whenever $(v_1, \ldots, v_m)$ is not a satisfying assignment of C_p, and to take some nonzero constant value otherwise.

The simplest way to do this would be to construct Bruhat logic gates σ_i so that

$$\boldsymbol{e}\big(\sigma_i \rtimes (v_{i-1}, v_i)\big) \equiv 0 \mod p, \quad \text{when } (v_{i-1}, v_i) \notin L_{i-1},$$

and

$$\boldsymbol{e}\big(\sigma_i \rtimes (v_{i-1}, v_i)\big) \equiv 1 \mod p, \quad \text{when } (v_{i-1}, v_i) \in L_{i-1}.$$

To keep computations manageable, we weaken the condition by sometimes only requiring

$$\boldsymbol{e}\big(\sigma_i \rtimes (v_{i-1}, v_i)\big) \not\equiv 0 \mod p, \quad \text{when } (v_{i-1}, v_i) \in L_{i-1}.$$

In §3.2.5, we explain how to complete the proof of Main Lemma 3.1.2 under these weakened conditions. We postpone the construction of σ_1 and σ_{m+1} until §3.2.3.

3.2.2 Bruhat compound logic gates

In this subsection, we explain how the technical requirements to couple together two Bruhat logic gates ϕ with k wires and ϕ' with k' wires has changed from §2.2.3.

First, we give a technical construction to ensure that the number of wires carrying the value TRUE remains constant through logic gates. This matches the statement for rigid circuits given in Lemma 2.1.2. Given a Bruhat logic gate ϕ with k wires, we now say ϕ is *balanced* modulo p if $|\phi| - k \equiv 0 \mod p^3$.

To construct the *restriction* of ϕ, which we write $\phi_\circ$, we append to the beginning of ϕ the block $(\max(\phi) + 1, \max(\phi) + 2, \ldots, \max(\phi) + p^3 - 1)$, which we call the *initial block* of $\phi_\circ$. We have:

Lemma 3.2.1. *For a Bruhat logic gate ϕ that is balanced modulo p, we have*

$$\boldsymbol{e}\left(\phi_\circ \rtimes (v, v')\right) \equiv 0 \mod p$$

when v and v' do not have an equal number of wires carrying the value TRUE. *When v and v' do have an equal number of wires carrying the value* TRUE, *we have*

$$\boldsymbol{e}\left(\phi_\circ \rtimes (v, v')\right) \equiv \boldsymbol{e}\left(\phi \rtimes (v, v')\right) \mod p.$$

The proof of this lemma is given in Subsection 3.3.1.

The construction of $\phi \wedge \phi'$ now proceed as in § 2.2.3. We restrict to the case where ϕ is one of our simple Bruhat gates, and where ϕ' is a compound gate made up of IDENTITY and SWAP gates, since the construction in §3.2.1 requires us to place every TESTEQ or ANDOR gate at the top of our circuit. We require further that ϕ and ϕ' be balanced modulo p.

The construction of $\phi \wedge \phi'$ involves inserting $\phi'_\circ$ in place of the penultimate block of ϕ, shifting elements appropriately. The relabeling procedure is exactly the same as in § 2.2.3, except we now take y to be the value of the first entry of the penultimate block of ϕ. Note that $\phi \wedge \phi'$ is balanced modulo p.

Recall the left, middle and right insertion operations given in Definition 2.2.2. The following lemma is the more general version of Lemma 2.2.3. These equations produce the polynomials given in Section 3.5 that are used in §3.2.5 to complete the proof of Main Lemma 3.1.2.

Lemma 3.2.2. *Given ϕ and ϕ' as above, if ϕ is not the identity gate, $(\phi \wedge \phi')_\circ$ behaves as the coupling of the logic gates associated to $\phi_\circ$ and $\phi'_\circ$ when the following six equations are satisfied:*

$$|\phi| - 2 \equiv 0 \mod p^3, \tag{1}$$

$$-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (10, 11)\big) + \boldsymbol{e}\big(L(\phi_\circ) \rtimes (10, 11)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (10, 11)\big) \equiv 0 \mod p, \tag{2}$$

$$-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (01, 11)\big) + \boldsymbol{e}\big(L(\phi_\circ) \rtimes (01, 11)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (01, 11)\big) \equiv 0 \mod p, \tag{3}$$

$$-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (00, 01)\big) + \boldsymbol{e}\big(L(\phi_\circ) \rtimes (00, 01)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (00, 01)\big) \equiv 0 \mod p, \tag{4}$$

$$-2\boldsymbol{e}\big(M(\phi_\circ)\rtimes(00,10)\big)+\boldsymbol{e}\big(L(\phi_\circ)\rtimes(00,10)\big)+\boldsymbol{e}\big(R(\phi_\circ)\rtimes(00,10)\big)\equiv 0 \mod p, \quad (5)$$

$$2\boldsymbol{e}\big(M^2(\phi_\circ)\rtimes(00,11)\big)-4\boldsymbol{e}\big(LM(\phi_\circ)\rtimes(00,11)\big)-4\boldsymbol{e}\big(RM(\phi_\circ)\rtimes(00,11)\big)+$$
$$\boldsymbol{e}\big(L^2(\phi_\circ)\rtimes(00,11)\big)+2\boldsymbol{e}\big(LR(\phi_\circ)\rtimes(00,11)\big)+\boldsymbol{e}\big(R^2(\phi_\circ)\rtimes(00,11)\big)\equiv 0 \mod p. \quad (6)$$

When ϕ is the identity gate, we need two equations to be satisfied:

$$|\phi|-1\equiv 0 \mod p^3, \quad (7)$$

$$-2\boldsymbol{e}\big(M(\phi_\circ)\rtimes(0,1)\big)+\boldsymbol{e}\big(L(\phi_\circ)\rtimes(0,1)\big)+\boldsymbol{e}\big(R(\phi_\circ)\rtimes(0,1)\big)\equiv 0 \mod p. \quad (8)$$

The proof of this lemma is given in Subsection 3.3.2.

3.2.3 Initializing and testing wires

We now give explicitly the construction of σ_1 and σ_{m+1}, to initialize wires at the beginning of our circuit and test the value of the output wire at the end. The construction of σ_1 is similar to the construction in 2.2.4, while the construction of σ_{m+1} is entirely different.

For σ_1, we begin with ψ, a compound Bruhat logic gate consisting of p^2-p copies of the identity wire, and then take $\sigma_1=\psi_\circ\rtimes(\vec{0},\vec{0})$, where $\vec{0}$ represents a circuit state with all wires set to FALSE. The identity gate construction given in Lemma 3.2.6 is simple enough that we can state what σ_1 looks like explicitly. It contains a sequence of p^2-p+1 blocks, each of size p^3-1, followed by the variables. The blocks themselves decrease by p^3 with each block step, and the variables fill in the missing terms.

For example, when $p=2$, σ_1 has 2 wires, and we have

$$\sigma_1=\boxed{17}\ \boxed{9}\ \boxed{1}\ 16\ 8,$$

where each of the numbers in boxes represent blocks of size p^3-1, using the $\boxed{x}$ notation described in § 1.2.2. For ease in notation, we shift elements down and write instead

$$\sigma_1=\boxed{5}\ \boxed{3}\ \boxed{1}\ 4\ 2.$$

We then modify σ_1 by duplicating each of the $p^2 - p$ terms at the end, in their respective positions. Our example above becomes

$$\sigma_1 = \boxed{5}\ \boxed{3}\ \boxed{1}\ 4\ 4\ 2\ 2.$$

Finally, we shift all elements of σ_1 up so that all of the elements are distinct and the final sequence is in strictly decreasing order. Our example now reads

$$\sigma_1 = \boxed{7}\ \boxed{4}\ \boxed{1}\ 6\ 5\ 3\ 2.$$

Note that σ_1 now has $N = 2p^2 - 2p$ output wires, as required.

Recall that, by the abuse of notation introduced in §2.2.2, for a vector

$$v = (a_1, \ldots, a_N) \in \mathbb{F}_2^N,$$

we let $\sigma_1(v_0, v)$ represent σ_1 after deleting every element corresponding to a 1 in v, where we take $v_0 = \varnothing$.

Having given the details of our construction, we now prove the following:

Lemma 3.2.3. *We have $\sigma_1 \rtimes (v_0, v) \equiv 1 \mod p$ precisely when exactly one of each pair $\{a_{2i-1}, a_{2i}\}$ is equal to 1, and $\sigma_1 \rtimes (v_0, v) \equiv 0 \mod p$ otherwise.*

Proof. By construction, when exactly one of each pair $\{a_{2i-1}, a_{2i}\}$ is equal to 1, then $\sigma_1 \rtimes (v_0, v) = \psi_\circ \rtimes (\vec{0}, \vec{0})$, so $\boldsymbol{e}(\sigma_1(v)) = \boldsymbol{e}(\psi_\circ \rtimes (\vec{0}, \vec{0})) \equiv 1 \mod p$. And, by Lemma 3.2.1, $\boldsymbol{e}(\sigma_1 \rtimes (v_0, v)) \equiv 0 \mod p$ unless $|v| = p^2 - p$.

The only case left to consider is when $|v| = p^2 - p$, but the condition of this lemma is false. In this case, there must be some i for which both a_{2i-1} and a_{2i} are equal to 1. However, σ_1 will then contain two adjacent blocks of size $p^3 - 1$ with no other elements whose values lie between those two blocks. In our example above, deleting both the 6 and the 5 leaves adjacent blocks $\boxed{7}$ and $\boxed{4}$ in σ_1.

The rearrangement of these two blocks are therefore independent of the rearrangement of the rest of σ_1, so that $\boldsymbol{e}(\sigma_1 \rtimes (v_0, v))$ is divisible by

$$\binom{2p^3 - 2}{p^3 - 1}.$$

This is $p^3 C_{p^3-1}$, where C_{p^3-1} is the p^3-1-th Catalan number. Thus the binomial coefficient is divisible by p, so $\boldsymbol{e}(\sigma_1 \rtimes (v_0, v)) \equiv 0 \mod p$, as desired. □

We now give the construction of σ_{m+1}. Recall, by the construction given in §3.2.1, that the final $p-1$ wires should all carry the value of the output wire, while the other $N-(p-1) = 2p^2-3p+1$ wires are grouped into $p-1$ sets of $2p-1$ wires.

We begin with

$$\sigma_{m+1} = (N, \ldots, 2, 1).$$

This choice is forced on us, since the input variables always must be in decreasing order and the smallest elements of the permutation. To finish our construction, we insert the sequence

$$\big(N+1, N+2, \ldots, N+(p-1)\big)$$

into σ_1 so as to divide the variables into the sets described above, i.e. first $p-1$ sets of $2p-1$ wires, followed by a final set of $p-1$ wires.

We call the sequence $\big(N+1, N+2, \ldots, N+(p-1)\big)$ *dividers*. Again, by abuse of notation, given a vector $v \in \mathbb{F}_2^N$, we write $\sigma_{m+1} \rtimes (v, \varnothing) = \sigma_{m+1} \rtimes (v, v_{m+1})$ for the sequence obtained by deleting from σ_{m+1} each variable corresponding to a 0 in v.

Lemma 3.2.4. *If the last element of v is 1, i.e. if the output wire contains the value* TRUE*, then $\boldsymbol{e}(\sigma_{m+1} \rtimes (v, v_{m+1})) \equiv -1 \mod p$. Otherwise, $\boldsymbol{e}(\sigma_{m+1} \rtimes (v, v_{m+1})) \equiv 0 \mod p$.*

Proof. By construction, the entire final set of $(p-1)$ wires will either all be TRUE or all be FALSE. Before all the swapping we did at the end of our circuit in §3.2.1, each of the original $p-1$ circuits contained $2p$ wires, with p wires set to TRUE. So, after the swapping, if the final set of wires is TRUE, each of the other sets will have $p-1$ wires set to TRUE, while if the final set of wires is FALSE, each of the other sets will have p wires set to TRUE.

We treat each case separately:

Case 1: The desired wire carries the value FALSE. Then the p wires set to TRUE in each of the other $p-1$ sets correspond in $\sigma_{m+1}(v, v_{m+1})$ to a sequence of p consecutive decreasing elements. There are $p!$ rearrangements of each of those sets, and the rearrangements of those

sets are independent of the rearrangement of the rest of the elements in the permutation, so that $\boldsymbol{e}(\sigma_{m+1}(v, v_{m+1})) \equiv 0 \mod p$.

Case 2: The desired wire carries the value TRUE. Then the $p-1$ wires set to TRUE in each of the other $p-1$ sets correspond in $\sigma_{m+1}(v, v_{m+1})$ to a sequence of $p-1$ consecutive decreasing elements. The final set of wires also contains $p-1$ decreasing elements. We count the number of rearrangements $\tau \leq \sigma_{m+1}(v, v_{m+1})$ based on the position of the dividers in τ.

When the dividers remain in exactly the same position, then no other element can move out of its set either, since the variables in τ must remain to the left of every divider they were already to the left of in σ. This leaves only rearrangements within the p sets of $p-1$ strictly decreasing elements, for a total count of $\big((p-1)!\big)^p \equiv (-1)^p \equiv -1 \mod p$ by Wilson's theorem.

For all other choices of positions for the dividers, there will be some gap of size at least p between dividers. The number of ways to rearrange the at least p elements that fill this gap will be divisible by $p!$ These rearrangements are independent of the rearrangement of the rest of the sequence, and so the contribution from every other choice of positions for the dividers is 0 modulo p. Thus the total number of rearrangements is congruent to $-1 \mod p$, as desired. □

3.2.4 Parametrized gates

We now describe how to construct a parametrized family of logic gates whose count of rearrangements is a polynomial in the parameters. We use this construction in §3.2.5 to give SWAP, ANDOR and TESTEQ gates for arbitrary primes p.

Let ϕ be a logic gate containing an increasing sequence $(x_1, \ldots, x_t)$. We form the *parametrization* of ϕ with respect to (x_i) by replacing each of the elements x_i with a block of consecutive elements of length $z_i \in \mathbb{N}$ and shifting the other elements of ϕ up appropriately, and denote this as $\boxed{\phi}\,(x_i, z_i)$.

We require that the sequence of x_i's conclude prior to the penultimate block of ϕ, and by convention write x_{t+1} for the penultimate block of ϕ, with $z_{t+1} = p^3 - 1$.

Lemma 3.2.5. *For every parametrization* $\boxed{\phi}(x_i, z_i)$ *of a logic gate* ϕ*, there is a polynomial* $g(z_1, \ldots, z_{t+1})$ *over* $\mathbb{Q}$ *such that* $g(z_1, \ldots, z_t, p^3 - 1) = \boldsymbol{e}\left(\boxed{\phi}(x_i, z_i)\right)$ *for every* p*.*

The proof of this lemma is given in Subsection 3.3.3.

3.2.5 Mod-p modification

Recall that for a logic gate L, we say $(v_1, v_2) \in L$ whenever (v_1, v_2) satisfies the logic gate, and $(v_1, v_2) \notin L$ otherwise. For computational reasons, it is easier to find simple logic gates if we relax the condition that $\boldsymbol{e}(\sigma \rtimes (v_1, v_2)) \equiv 1 \mod p$ whenever (v_1, v_2) satisfies the logic gate. We never alter the set of conditions $\boldsymbol{e}(\sigma \rtimes (v_1, v_2)) \equiv 0 \mod p$ when (v_1, v_2) fails to satisfy the logic gate. We describe the modified conditions below.

IDENTITY gate L_1: $\boldsymbol{e}\big(\sigma \rtimes (v_1, v_2)\big) \equiv 1 \mod p$ whenever $(v_1, v_2) \in L_1$.

SWAP gate L_2: $\boldsymbol{e}\big(\sigma \rtimes (v_1, v_2)\big) \equiv 1 \mod p$ whenever $(v_1, v_2) \in L_2$.

ANDOR gate L_3: $\boldsymbol{e}\big(\sigma \rtimes (v_1, v_2)\big) \not\equiv 0 \mod p$ whenever $(v_1, v_2) \in L_3$.

TESTEQ gate L_4: $\boldsymbol{e}\big(\sigma \rtimes (v_1, v_2)\big) \not\equiv 0 \mod p$ whenever $(v_1, v_2) \in L_4$.

We now have all of the conditions on our simple Bruhat logic gates, allowing us to state and prove the following:

Lemma 3.2.6. *For every prime* $p \geq 2$*, there is an* IDENTITY *gate satisfying the condition of Lemma 3.2.2. In addition, for each of* SWAP*,* ANDOR *and* TESTEQ*, there are parametrized Bruhat logic gates* ϕ *such that the conditions above together with the conditions in Lemma 3.2.2 give a system of polynomial equations in the parameters* $\{z_i\}$ *that has solutions modulo* p *for all primes* $p \geq 11$*.*

Proof. We prove the statement for each gate by giving an explicit construction. The following permutations represent our four desired gates before restriction. The IDENTITY gate works correctly modulo p without parametrization. The other three gates are parametrized with respect to the sequence $\{3, 4, 5, 6, 7\}$. The $\boxed{2}$ in the IDENTITY gate and the $\boxed{8}$ in the other three gates represent the penultimate blocks of size $p^3 - 1$.

We treat each of the equations in Lemma 3.2.2 first over $\mathbb{Q}$, so that the equations correspond to some algebraic variety over $\mathbb{Q}$. For each gate, we are able to give explicitly a rational point on that variety, and the rational nonzero values taken by the AndOr and TestEq gates. For every prime $p \geq 11$, this corresponds to a solution modulo p.

1. Identity gate:

$$\phi = 1\ \boxed{2}\ 3.$$

The equations that ϕ must satisfy are (7) and (8) from Lemma 3.2.2, and the following four equations:

$$\boldsymbol{e}\big(\phi \rtimes (0,0)\big) = \boldsymbol{e}\big(\phi \rtimes (1,1)\big) = 1, \quad \boldsymbol{e}\big(\phi \rtimes (0,1)\big) = \boldsymbol{e}\big(\phi \rtimes (1,0)\big) = 0.$$

The last two equations are guaranteed to be satisfied by Lemma 3.2.1. Since $|\phi| = p^3 + 1$, we have that (7) is satisfied. Note that

$$\begin{aligned}\boldsymbol{e}\big(\phi \rtimes (0,0)\big) &= \boldsymbol{e}\big(\phi \rtimes (1,1)\big) = \boldsymbol{e}\big(L(\phi) \rtimes (0,1)\big) \\ &= \boldsymbol{e}\big(M(\phi) \rtimes (0,1)\big) = \boldsymbol{e}\big(R(\phi) \rtimes (0,1)\big) = 1,\end{aligned}$$

since these permutations are all just strictly increasing sequences of length p^3. Thus the remaining two equations given here and (8) are satisfied, as desired.

For each of the remaining three gates, there are 6 equations from Lemma 3.2.2, and an additional 16 equations for every possible pair of input and output wires. However, applying Lemma 3.2.1 as above, we see that 10 of these equations will be satisfied automatically. For the remaining 12 equations, we use Lemma 3.2.5 to compute the corresponding polynomials in $\{z_i\}$, for $1 \leq i \leq 5$. We write out these polynomials explicitly in Appendix 3.5.

2. Swap gate:

$$\phi = 2\ \boxed{3}\ 12\ \boxed{4}\ 1\ \boxed{5}\ 10\ \boxed{6}\ 13\ \boxed{7}\ \boxed{8}\ 11\ 9.$$

The system of equations in §3.5.1 has a unique solution over $\mathbb{Q}$:

$$(z_1, z_2, z_3, z_4, z_5) = (-1, -2, 0, 1, -2),$$

so the system of equations is solvable mod p, for every prime $p \geq 2$.

3. AndOr gate:

$$\phi = 2\ \boxed{3}\ 13\ \boxed{4}\ 11\ \boxed{5}\ 1\ \boxed{6}\ 10\ \boxed{7}\ \boxed{8}\ 12\ 9.$$

The system of equations in §3.5.2 reduces to a two-dimensional variety over $\mathbb{Q}$ of degree 2, with infinitely many rational points, including the point

$$(z_1, z_2, z_3, z_4, z_5) = (-2, 1, -3, 1, -1).$$

The nonzero values $\boldsymbol{e}(\sigma(v, v'))$ takes are 2 and 4, so we require $p \neq 2$, and the system of equations is solvable mod p for every prime $p \geq 3$.

4. TestEq gate:

$$\phi = 2\ \boxed{3}\ 12\ \boxed{4}\ 10\ \boxed{5}\ 1\ \boxed{6}\ 13\ \boxed{7}\ \boxed{8}\ 11\ 9.$$

The system of equations in §3.5.3 reduces to a one-dimensional variety over $\mathbb{Q}$ of degree 1, with infinitely many rational points, including the point

$$(z_1, z_2, z_3, z_4, z_5) = (-2, -\tfrac{8}{3}, \tfrac{5}{3}, -3, 2),$$

with nonzero values of $\frac{7}{3}$ and $\frac{-8}{3}$ for $\boldsymbol{e}(\sigma(v, v'))$, so that the system of equations is solvable mod p for every prime $p \geq 11$.

□

3.2.6 Proof of Main Lemma 3.1.2.

Given a rigid circuit C, we construct the mod-p parallel circuit C_p with $\boldsymbol{e}(C) = \boldsymbol{e}(C_p)$. We then construct a Bruhat circuit σ that mirrors the behavior of C_p.

By Lemma 3.2.3, our choices of variable assignments v_1 in the sum in (2.2.2) are restricted to those with $N = 2p^2 - 2p$ wires grouped in pairs, with exactly one wire set to TRUE and one wire set to FALSE in each pair. By lemmas 3.2.2 and 3.2.6, the inside product in (2.2.2) is congruent to 0 mod p except when $(v_1, \ldots, v_m)$ is a set of circuit states satisfying C_p.

By the parallel circuit construction, every time an AndOr gate operation occurs in C, it occurs $p-1$ times in C_p, acting on the same set of truth values, which gives a contribution of $1 \mod p$ to the product in (2.2.2).

The same is true for the TestEq operations that occur after the parallel circuit has already been constructed. For the TestEq operations that occur in the construction of the parallel circuit (i.e. the TestEq operations used to force each of the copies of the circuit to have the same initial truth values), just repeat them $p-1$ times, which has no impact on the operation of the circuit.

The Identity and Swap operations all also give a contribution of $1 \mod p$ to the product, by construction.

In summary, the contribution to the product from the operation of each of the gates is 1. The only contribution left comes from σ_{m+1}, which, by Lemma 3.2.4, multiplies the product by -1 if the output wire is true and 0 otherwise. □

3.3 Proof of lemmas

3.3.1 Proof of Lemma 3.2.1.

Write $|v|$ for the number of wires carrying the value true in v. Since the elements in the initial block of $\phi_\circ$ are larger than every other element of ϕ, the Bruhat order gives no restriction on the position of these elements relative to the position of the elements of ϕ in a rearrangement $\tau \leq \phi_\circ$. Thus:

$$\begin{aligned} \boldsymbol{e}\left(\phi_\circ \rtimes (v, v')\right) &= \binom{|\phi \rtimes (v, v')| + p^3 - 1}{p^3 - 1} \boldsymbol{e}\left(\phi \rtimes (v, v')\right) \\ &= \binom{|\phi| - (k - |v|) - |v'| + p^3 - 1}{p^3 - 1} \boldsymbol{e}\left(\phi \rtimes (v, v')\right). \end{aligned}$$

Since ϕ is balanced, we know $|\phi| - k \equiv 0 \mod p^3$. Write $|\phi| - k = ap^3$ and $|v| - |v'| = b$. We have $|b| \leq 2p^2 + 2p$. Observe that, for integers a, b with $a > 0$ and $0 < |b| \leq 2p^2 + 2p$, as

long as $p \geq 3$ we have

$$\binom{ap^3 + p^3 - 1 + b}{p^3 - 1} \equiv 0 \mod p.$$

On the other hand, for $a > 0$ and $b = 0$, we have:

$$\binom{ap^3 + p^3 - 1 + b}{p^3 - 1} \equiv 1 \mod p.$$

We thus have $\boldsymbol{e}(\phi_\circ \rtimes (v, v')) \equiv 0 \mod p$ whenever $|v| \neq |v'|$. When $|v| = |v'|$, the binomial coefficient evaluates to 1 modulo p, so that we have $\boldsymbol{e}(\phi_\circ \rtimes (v, v')) \equiv \boldsymbol{e} \rtimes (\phi(v, v'))$, as desired. □

3.3.2 Proof of Lemma 3.2.2.

Equations (1) and (8) follow immediately from the requirement that ϕ be balanced.

We prove the lemma in the case where ϕ is the SWAP, ANDOR or TESTEQ gate, and then explain how to adjust the proof when ϕ is the IDENTITY gate. Write the input and output wires of $\phi \wedge \phi'$ as $v_1 \wedge v_1'$ and $v_2 \wedge v_2'$, where here $\wedge$ denotes concatenation, $v_i \in \mathbb{F}_2^2$, and $v_i' \in \mathbb{F}_2^k$, so that there are a total of $k + 2$ input and output wires in $\phi \wedge \phi'$.

For every rearrangement τ of $(\phi \wedge \phi') \rtimes (v_1 \wedge v_1', v_2 \wedge v_2')$, we can restrict to a rearrangement of $\phi_\circ'$ by looking only at the elements that come from $\phi_\circ'$ in τ (and shifting the elements back down to their previous values). Write this new rearrangement as $\tau|_{\phi_\circ'}$. Then, writing $\tau|_{\phi_\circ'}$ in one-line notation, $\tau|_{\phi_\circ'}$ begins with some sequence (possibly of length zero) of input variables, and ends with some sequence (possibly of length zero) of output variables. Call the length of the first sequence $\ell(\tau)$ and the length of the second sequence $r(\tau)$.

Now we wish to begin with a permutation $\tau' \leq \phi' \rtimes (v_1', v_2')$, and count the number of $\tau \leq (\phi \wedge \phi') \rtimes (v_1 \wedge v_1', v_2 \wedge v_2')$ with $\tau|_{\phi_\circ'} = \tau'$. Since we have fixed τ', we need to consider only the possible ways that the elements of ϕ can be rearranged with respect to each other or shuffled among the elements of ϕ'.

By Lemma 3.2.1, we have:

$$\boldsymbol{e}\big((\phi \wedge \phi') \rtimes (v_1 \wedge v_1', v_2 \wedge v_2')\big) \equiv 0 \mod p$$

unless $|v_1|+|v_1'| \equiv |v_2|+|v_2'| \mod p^3$. Thus, we restrict our attention to the case where that holds. Then, by the same argument used in the proof of Lemma 3.2.1, we have:

$$|\phi' \rtimes (v_1', v_2')| = |\phi'| + (|v_1'| - k) - |v_2'| \equiv |v_1'| - |v_2'| \equiv |v_2| - |v_1| \mod p^3.$$

Since $\phi_\circ'$ adds a block of size $p^3 - 1$, we have $|\phi_\circ' \rtimes (v_1', v_2')| \equiv |v_2| - |v_1| - 1 \mod p^3$.

We then note that the number of $\tau \leq (\phi \wedge \phi') \rtimes (v_1 \wedge v_1', v_2 \wedge v_2')$ with $\tau|_{\phi_\circ'} = \tau'$ is equal to:

$$\frac{1}{\ell(\tau')!\, r(\tau')!} \cdot \boldsymbol{e}\big(L^{\ell(\tau')} R^{r(\tau')} M^{|v_2|-|v_1|-\ell(\tau')-r(\tau')}(\phi) \rtimes (v_1, v_2)\big).$$

With the $\frac{1}{\ell(\tau')!\, r(\tau')!}$ term because repeated left and right insertion gives sets of consecutive decreasing elements that can be rearranged in $\ell(\tau')!$ and $r(\tau')!$ ways, respectively, but exactly one of these arrangements actually corresponds to τ'.

Next, we group permutations τ' based on the value of $\ell(\tau')$ and $r(\tau')$. Let $g(\ell, r)$ be the number of $\tau' \leq \phi' \rtimes (v_1', v_2')$ with $\ell(\tau') = \ell$ and $r(\tau') = r$. Of course, the value $g(\ell, r)$ also depends on $\phi' \rtimes (v_1', v_2')$. We omit this dependence from our notation for the sake of readability. We then have:

$$\boldsymbol{e}\big((\phi \wedge \phi') \rtimes (v_1 \wedge v_1', v_2 \wedge v_2')\big) = \sum_{\ell, r \geq 0} \frac{g(\ell, r)}{\ell!\, r!} \cdot \boldsymbol{e}\big(L^\ell R^r M^{|v_2|-|v_1|-\ell-r}(\phi) \rtimes (v_1, v_2)\big). \quad (3.3.1)$$

It remains to compute $g(\ell, r)$ for an arbitrary choice of $v_1 \wedge v_1'$, $v_2 \wedge v_2'$, ℓ, and r. Beginning or ending τ' with a particular sequence of elements is the same as counting the number of permutations of τ' with those elements removed. Thus we have:

$$\ell!\, r! \sum_{|v_1'|-|w_1'|=\ell} \sum_{|w_2'|-|v_2'|=r} \boldsymbol{e}\big(\phi_\circ' \rtimes (w_1', w_2')\big) = \sum_{\ell \leq h \leq k} \sum_{r \leq j \leq k} g(h, j). \quad (3.3.2)$$

Here the left hand sum is taken over circuit states w_1' obtained from v_1' by flipping ℓ wires from TRUE to FALSE and w_2' obtained from v_2' by flipping r from FALSE to TRUE. The $\ell!$ and $r!$ terms account for the ways to arrange the initial and final sequences, each strictly decreasing, of length ℓ and r respectively.

Either of the wire flips just described reduces $|v_1'| - |v_2'|$ by one, so that we have:

$$|v_2| - |v_1| = |v_1'| - |v_2'| = \ell + r + |w_1'| - |w_2'|.$$

Recall that, by Lemma 3.2.1, for the left hand side of (3.3.2) to be nonzero modulo p we must have $|w_1'| - |w_2'| = 0$.

Thus, on one hand, if $\ell + r > |v_1'| + |v_2'|$, the left hand side of (3.3.2) is always 0 modulo p, so that we conclude $g(h, j) \equiv 0 \mod p$ for every h, j with $h + j > \ell + r$.

On the other hand, if the left hand side of (3.3.2) is nonzero modulo p, we have

$$|v_2| - |v_1| = |v_1'| - |v_2'| = \ell + r \geq 0.$$

Since $v_1, v_2 \in \mathbb{F}_2^2$, we have $|v_2| - |v_1| \leq 2$, and so we conclude $0 \leq |v_1'| - |v_2'| \leq 2$.

Since ϕ' is composed entirely of IDENTITY and SWAP gates, each input wire in v_1' can be matched with one output wire in v_2' whose signal state it controls. If we require $\boldsymbol{e}(\phi_\circ' \rtimes (w_1', w_2'))$ to be nonzero, then all the input-output wire pairs in w_1', w_2' match, and somewhere between zero and two input-output pairs of wires in v_1', v_2' have an input wire reading TRUE and an output wire reading FALSE. We refer to such a pair as a (TRUE, FALSE) pair and note that the number of (TRUE, FALSE) pairs is equal to $|v_1'| - |v_2'|$.

Of course, whenever $|v_1'| - |v_2'| > 0$, the output wires do not correspond correctly to the input wires, so we need the count

$$\boldsymbol{e}\big((\phi \wedge \phi') \rtimes (v_1 \wedge v_1', v_2 \wedge v_2')\big) \equiv 0 \mod p$$

for this choice of $(v_1 \wedge v_1', v_2 \wedge v_2')$. We will now use (3.3.1) and (3.3.2) to do a computation which will produce the remaining equations given in the statement of this lemma.

Note that, because ϕ' is made up of IDENTITY and SWAP gates, the technical restrictions in §3.2.5 are enough to ensure that $\boldsymbol{e}(\phi_\circ' \rtimes (w_1', w_2')) \equiv 1 \mod p$ whenever $\boldsymbol{e}(\phi_\circ' \rtimes (w_1', w_2'))$ is nonzero modulo p.

Case 1: Zero (TRUE, FALSE) pairs in (v_1', v_2'). Then $\boldsymbol{e}(\phi_\circ' \rtimes (w_1', w_2'))$ is nonzero modulo p precisely when $w_1' = v_1'$ and $w_2' = v_2'$, so that (3.3.2) gives $g(0, 0) \equiv 1 \mod p$ and $g(h, j) \equiv 0 \mod p$ otherwise. Then (3.3.1) becomes:

$$\boldsymbol{e}\big((\phi \wedge \phi') \rtimes (v_1 \wedge v_1', v_2 \wedge v_2')\big) \equiv \boldsymbol{e}\big(\phi \rtimes (v_1, v_2)\big) \mod p.$$

So that $\boldsymbol{e}(\phi \wedge \phi')$ behaves like the Bruhat logic gate ϕ on the top two wires.

Case 2: One (TRUE, FALSE) pair in (v_1', v_2'). Then for $\ell = 1$, $r = 0$, there is exactly one choice of w_1', w_2' with $\boldsymbol{e}(\phi_\circ' \rtimes (w_1', w_2'))$ nonzero modulo p; this choice corresponds to switching the TRUE input wire in the (TRUE, FALSE) pair to FALSE. Likewise, for $\ell = 0$, $r = 1$, there is exactly one choice. Thus (3.3.2) gives $g(1,0) \equiv g(0,1) \equiv 1 \mod p$.

Now taking $\ell = r = 0$, we have $(w_1', w_2') = (v_1', v_2')$, and $\boldsymbol{e}(\phi_\circ' \rtimes (w_1', w_2')) \equiv 0 \mod p$ by Lemma 3.2.1. Then (3.3.2) gives $g(0,0) \equiv -2 \mod p$. There are four possible choices of v_1 and v_2 satisfying $|v_2| - |v_1| = 1$, which give (2), (3), (4), and (5).

Case 3: Two (TRUE, FALSE) pairs in (v_1', v_2'). We proceed with a calculation similar to the one above. For $\ell = 2$, $r = 0$ and $\ell = 0$, $r = 2$, there is exactly one choice of w_1', w_2', while for $\ell = 1$, $r = 1$, there are two choices. Then (3.3.2) gives:

$$g(2,0) \equiv g(1,1) \equiv g(0,2) \equiv 2 \mod p.$$

For $\ell + r = 1$ and $\ell + r = 0$, Lemma 3.2.1 tells us $\boldsymbol{e}(\phi_\circ' \rtimes (w_1', w_2')) \equiv 0 \mod p$, and we compute:

$$g(1,0) \equiv g(0,1) \equiv -4 \mod p \quad \text{and} \quad g(0,0) \equiv 2 \mod p.$$

There is only one choice of v_1 and v_2 satisfying $|v_2| - |v_1| = 2$, which gives (6).

This completes the proof when ϕ is the SWAP, ANDOR or TESTEQ gate. When ϕ is the IDENTITY gate, we follow the same argument and find that we must only consider the cases when $0 \leq |v_2| - |v_1| \leq 1$. Working through Case 1 and Case 2 above gives (8). □

3.3.3 Proof of Lemma 3.2.5

We describe a function that sends a permutation τ with $\tau \leq \boxed{\phi}(x_i, z_i)$ to a permutation $\tau^* \leq \phi$. Note that since the blocks are in increasing order in $\boxed{\phi}(x_i, z_i)$, they will still be in increasing order in τ. The only elements that can lie "within" one of these blocks (where "within" means to the right of some element from the block and to the left of another element of the block) are elements that were originally larger than the block and to its left, or smaller than the block and to its right.

To produce τ^*, push the elements of τ that have moved within blocks out of their blocks, either to the left or right, back to the side they came from. We treat the penultimate block the same way. Then replace the blocks with the old x_i's and shift everything back down.

Since ϕ has finite length, there are only finitely many choices of τ^*. For each τ^* we count the number of possible $\tau \leq \boxed{\phi}\,(x_i, z_i)$ in the pre-image of the function described above. To do this computation, we consider the number of elements in τ immediately to the left of an x_i and larger than it, or immediately to the right of an x_i and smaller than it. Call the first number ℓ and the second r. Then we are counting rearrangements of the block sequence

$$\boxed{3}\,\boxed{2}\,\boxed{1}$$

with blocks of lengths ℓ, z_i and r, respectively, such that none of the elements from the $\boxed{3}$ or $\boxed{1}$ block cross the entire $\boxed{2}$ block. We sum over the number $h \leq \ell$ of elements that move from the $\boxed{3}$ block into the $\boxed{2}$ block, and the number $j \leq r$ of elements that move from the $\boxed{1}$ block into the $\boxed{2}$ block:

$$\sum_{0\leq h\leq \ell}\sum_{0\leq j\leq r}\binom{z_i+h+j-2}{h}\binom{z_i+j-2}{j}.$$

We thus obtain:

$$\boldsymbol{e}\left(\boxed{\phi}\,(x_i, z_i)\right) = \sum_{\tau^*\leq\phi}\prod_{i=1}^{t+1}\sum_{0\leq h\leq \ell}\sum_{0\leq j\leq r}\binom{z_i+h+j-2}{h}\binom{z_i+j-2}{j},$$

which is a polynomial in the z_i's and p^3-1, as desired. □

3.4 Final remarks

3.4.1

Our *computer assisted proof* is both technical and innovative. The computer algebra solution of large algebraic systems used to encode logical gates is novel as until recently such computations remained far out of reach.

The equations in Section 3.5 are nonhomogeneous polynomials in 5 variables, with a maximum total degree of 5. The coefficients are nonnegative integers ≤ 400. Before inserting

parameters, the gates were permutations of length 8, so there were $8! = 40320$ possibilities. In fact, the requirement that the variables be in strictly decreasing order restricts the possibilities significantly. After some experimentation, we added the further restriction that the first variable be in the first position of the permutation. After these restrictions, only 96 possibilities remain.

For each gate, we generated the system of 12 polynomials in `C++`, for each of these 96 possible permutations. We then computed which systems had solutions over $\mathbb{C}$; the systems were tested with `Macaulay2`.[1] Generating the systems took 314.4 seconds, or an average of 3.3 seconds per system. Testing all 96 systems took less than ten seconds for each of the three gates. If we had needed to extend our search to 6 variables, the cost in computing time would have increased significantly, as shown in Figure 3.1.

Here is the result of our computation. For each gate, at least one of the of the 96 possible permutations produced systems of equations with nontrivial solutions over $\mathbb{C}$. To be precise:

- ◇ For the Swap gate, this worked for 2 of the 96 possible permutations.
- ◇ For the AndOr gate, this worked for 47 of the 96 possible permutations.
- ◇ For the TestEq gate, this worked for 4 of the 96 possible permutations.

Variables	Candidate gates	Computation time per candidate gate (sec.)
4	6	≤ 0.1
5	96	3.3
6	1200	1618 ($\sim$ 27 minutes)

Figure 3.1: Candidate permutations and computation time.

[1] Computations were made with an `Intel®` Core™ i7-3610QM CPU with 2.30GHz, 4 cores and 8Gb of RAM.

3.4.2

Let us quickly mention complexity implications of our results for people unfamiliar with modern Complexity Theory. Roughly, when a counting problem is #P-complete, this is an extremely strong evidence against it being computable in polynomial time, much stronger than $\mathsf{P} \neq \mathsf{NP}$, for example. Indeed, otherwise Toda's theorem $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$ implies that every problem in *polynomial hierarchy* PH can be solved in polynomial time [Tod91].

3.4.3

We are also curious about variations on Theorem 1.1.4. For example, is computing the size of the principal ideal of the *strong Bruhat order* #P-complete? What about other finite Coxeter groups? We refer [BjB05] for definitions and the background.

Finally, we conjecture that computing the number $R(\sigma)$ of reduced factorizations of a permutation $\sigma \in S_n$ into adjacent transpositions is #P-complete. Recall that $R(\sigma)$ can be computed in polynomial time in several special cases, see e.g. [MPP17]. Note that such factorizations can be viewed as saturated chains $1 \to \sigma$ in the weak Bruhat order $B_n = (S_n, \leq)$.

3.5 Gate equations

We print here the systems of polynomial equations for the parametrized Swap, AndOr, and TestEq gates given in Lemma 3.2.6. For each of these gates, there are six equations from Lemma 3.2.2 and six equations from the requirements for the logical operation of the gate itself, for a total of twelve equations.

3.5.1 Swap gate.

1. $|\phi| - 2 \equiv 0 \mod p^3$:

   $z_1 + z_2 + z_3 + z_4 + z_5 + 4 = 0$

2. $\boldsymbol{e}\big(\phi \rtimes (11,11)\big) \equiv 1 \mod p$:

   $2z_2z_5^3+2z_1z_5^3+4z_5^3+6z_2z_4z_5^2+6z_1z_4z_5^2+12z_4z_5^2+3z_2z_3z_5^2+3z_1z_3z_5^2+6z_3z_5^2+3z_2^2z_5^2+3z_1z_2z_5^2+15z_2z_5^2+6z_1z_5^2+15z_5^2+6z_2z_4^2z_5+6z_1z_4^2z_5+12z_4^2z_5+6z_2z_3z_4z_5+6z_1z_3z_4z_5+12z_3z_4z_5+6z_2^2z_4z_5+6z_1z_2z_4z_5+30z_2z_4z_5+12z_1z_4z_5+30z_4z_5+3z_2z_3z_5+3z_1z_3z_5+6z_3z_5+3z_2^2z_5+3z_1z_2z_5+13z_2z_5+4z_1z_5+11z_5 = 6$

3. $\boldsymbol{e}\big(\phi \rtimes (10,01)\big) \equiv 1 \mod p$:

   $2z_5^3 + 6z_4z_5^2 + 3z_3z_5^2 + 3z_2z_5^2 + 12z_5^2 + 6z_4^2z_5 + 6z_3z_4z_5 + 6z_2z_4z_5 + 24z_4z_5 + 9z_3z_5 + 9z_2z_5 + 16z_5 + 6z_4^2 + 6z_3z_4 + 6z_2z_4 + 12z_4 = 6$

4. $\boldsymbol{e}\big(\phi \rtimes (10,10)\big) \equiv 0 \mod p$:

   $2z_5^3 + 6z_4z_5^2 + 3z_3z_5^2 + 3z_2z_5^2 + 12z_5^2 + 6z_4^2z_5 + 6z_3z_4z_5 + 6z_2z_4z_5 + 24z_4z_5 + 9z_3z_5 + 9z_2z_5 + 22z_5 + 6z_4^2 + 6z_3z_4 + 6z_2z_4 + 18z_4 + 6z_3 + 6z_2 + 12 = 0$

5. $\boldsymbol{e}\big(\phi \rtimes (01,10)\big) \equiv 1 \mod p$:

   $2z_2z_5^3 + 2z_1z_5^3 + 2z_5^3 + 6z_2z_4z_5^2 + 6z_1z_4z_5^2 + 6z_4z_5^2 + 3z_2z_3z_5^2 + 3z_1z_3z_5^2 + 3z_3z_5^2 + 3z_2^2z_5^2 + 3z_1z_2z_5^2 + 18z_2z_5^2 + 12z_1z_5^2 + 15z_5^2 + 6z_2z_4^2z_5 + 6z_1z_4^2z_5 + 6z_4^2z_5 + 6z_2z_3z_4z_5 + 6z_1z_3z_4z_5 + 6z_3z_4z_5 + 6z_2^2z_4z_5 + 6z_1z_2z_4z_5 + 36z_2z_4z_5 + 24z_1z_4z_5 + 30z_4z_5 + 9z_2z_3z_5 + 9z_1z_3z_5 + 9z_3z_5 + 9z_2^2z_5 + 9z_1z_2z_5 + 40z_2z_5 + 22z_1z_5 + 31z_5 + 6z_2z_4^2 + 6z_1z_4^2 + 6z_4^2 + 6z_2z_3z_4 + 6z_1z_3z_4 + 6z_3z_4 + 6z_2^2z_4 + 6z_1z_2z_4 + 30z_2z_4 + 18z_1z_4 + 24z_4 + 6z_2z_3 + 6z_1z_3 + 6z_3 + 6z_2^2 + 6z_1z_2 + 24z_2 + 12z_1 + 18 = 6$

6. $\boldsymbol{e}(\phi \rtimes (01,01)\big) \equiv 0 \mod p$:

   $2z_2z_5^3+2z_1z_5^3+2z_5^3+6z_2z_4z_5^2+6z_1z_4z_5^2+6z_4z_5^2+3z_2z_3z_5^2+3z_1z_3z_5^2+3z_3z_5^2+3z_2^2z_5^2+3z_1z_2z_5^2+18z_2z_5^2+12z_1z_5^2+15z_5^2+6z_2z_4^2z_5+6z_1z_4^2z_5+6z_4^2z_5+6z_2z_3z_4z_5+6z_1z_3z_4z_5+6z_3z_4z_5+6z_2^2z_4z_5+6z_1z_2z_4z_5+$

$36z_2z_4z_5 + 24z_1z_4z_5 + 30z_4z_5 + 9z_2z_3z_5 + 9z_1z_3z_5 + 9z_3z_5 + 9z_2^2z_5 + 9z_1z_2z_5 + 34z_2z_5 + 16z_1z_5 + 25z_5 + 6z_2z_4^2 + 6z_1z_4^2 + 6z_4^2 + 6z_2z_3z_4 + 6z_1z_3z_4 + 6z_3z_4 + 6z_2^2z_4 + 6z_1z_2z_4 + 24z_2z_4 + 12z_1z_4 + 18z_4 = 0$

7. $\boldsymbol{e}\big(\phi \rtimes (00,00)\big) \equiv 1 \mod p$:

   $2z_5^3 + 6z_4z_5^2 + 3z_3z_5^2 + 3z_2z_5^2 + 18z_5^2 + 6z_4^2z_5 + 6z_3z_4z_5 + 6z_2z_4z_5 + 36z_4z_5 + 15z_3z_5 + 15z_2z_5 + 40z_5 + 12z_4^2 + 12z_3z_4 + 12z_2z_4 + 36z_4 + 6z_3 + 6z_2 + 15 = 3$

8. $-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (10,11)\big) + \boldsymbol{e}\big(L(\phi_\circ) \rtimes (10,11)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (10,11)\big) \equiv 0 \mod p$:

   $2z_5^4 + 8z_4z_5^3 + 5z_3z_5^3 + 5z_2z_5^3 + 2z_1z_5^3 + 12z_5^3 + 12z_4^2z_5^2 + 15z_3z_4z_5^2 + 15z_2z_4z_5^2 + 6z_1z_4z_5^2 + 36z_4z_5^2 + 3z_3^2z_5^2 + 6z_2z_3z_5^2 + 3z_1z_3z_5^2 + 18z_3z_5^2 + 3z_2^2z_5^2 + 3z_1z_2z_5^2 + 18z_2z_5^2 + 6z_1z_5^2 + 22z_5^2 + 6z_4^3z_5 + 12z_3z_4^2z_5 + 12z_2z_4^2z_5 + 6z_1z_4^2z_5 + 30z_4^2z_5 + 6z_3^2z_4z_5 + 12z_2z_3z_4z_5 + 6z_1z_3z_4z_5 + 33z_3z_4z_5 + 6z_2^2z_4z_5 + 6z_1z_2z_4z_5 + 33z_2z_4z_5 + 12z_1z_4z_5 + 40z_4z_5 + 3z_3^2z_5 + 6z_2z_3z_5 + 3z_1z_3z_5 + 13z_3z_5 + 3z_2^2z_5 + 3z_1z_2z_5 + 13z_2z_5 + 4z_1z_5 + 12z_5 = 0$

9. $-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (01,11)\big) + \boldsymbol{e}\big(L(\phi_\circ) \rtimes (01,11)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (01,11)\big) \equiv 0 \mod p$:

   $2z_2z_5^4 + 2z_1z_5^4 + 2z_5^4 + 8z_2z_4z_5^3 + 8z_1z_4z_5^3 + 8z_4z_5^3 + 5z_2z_3z_5^3 + 5z_1z_3z_5^3 + 5z_3z_5^3 + 5z_2^2z_5^3 + 7z_1z_2z_5^3 + 22z_2z_5^3 + 2z_1^2z_5^3 + 16z_1z_5^3 + 17z_5^3 + 12z_2z_4^2z_5^2 + 12z_1z_4^2z_5^2 + 12z_4^2z_5^2 + 15z_2z_3z_4z_5^2 + 15z_1z_3z_4z_5^2 + 15z_3z_4z_5^2 + 15z_2^2z_4z_5^2 + 21z_1z_2z_4z_5^2 + 66z_2z_4z_5^2 + 6z_1^2z_4z_5^2 + 48z_1z_4z_5^2 + 51z_4z_5^2 + 3z_2z_3^2z_5^2 + 3z_1z_3^2z_5^2 + 3z_3^2z_5^2 + 6z_2^2z_3z_5^2 + 9z_1z_2z_3z_5^2 + 30z_2z_3z_5^2 + 3z_1^2z_3z_5^2 + 24z_1z_3z_5^2 + 24z_3z_5^2 + 3z_2^3z_5^2 + 6z_1z_2^2z_5^2 + 27z_2^2z_5^2 + 3z_1^2z_2z_5^2 + 33z_1z_2z_5^2 + 67z_2z_5^2 + 6z_1^2z_5^2 + 37z_1z_5^2 + 43z_5^2 + 6z_2z_4^3z_5 + 6z_1z_4^3z_5 + 6z_4^3z_5 + 12z_2z_3z_4^2z_5 + 12z_1z_3z_4^2z_5 + 12z_3z_4^2z_5 + 12z_2^2z_4^2z_5 + 18z_1z_2z_4^2z_5 + 54z_2z_4^2z_5 + 6z_1^2z_4^2z_5 + 42z_1z_4^2z_5 + 42z_4^2z_5 + 6z_2z_3^2z_4z_5 + 6z_1z_3^2z_4z_5 + 6z_3^2z_4z_5 + 12z_2^2z_3z_4z_5 + 18z_1z_2z_3z_4z_5 + 57z_2z_3z_4z_5 + 6z_1^2z_3z_4z_5 + 45z_1z_3z_4z_5 + 45z_3z_4z_5 + 6z_2^3z_4z_5 + 12z_1z_2^2z_4z_5 + 51z_2^2z_4z_5 + 6z_1^2z_2z_4z_5 + 63z_1z_2z_4z_5 + 124z_2z_4z_5 + 12z_1^2z_4z_5 + 70z_1z_4z_5 + 79z_4z_5 + 3z_2z_3^2z_5 + 3z_1z_3^2z_5 + 3z_3^2z_5 + 6z_2^2z_3z_5 + 9z_1z_2z_3z_5 + 25z_2z_3z_5 + 3z_1^2z_3z_5 + 19z_1z_3z_5 + 19z_3z_5 + 3z_2^3z_5 + 6z_1z_2^2z_5 + 22z_2^2z_5 + 3z_1^2z_2z_5 + 26z_1z_2z_5 + 47z_2z_5 + 4z_1^2z_5 + 23z_1z_5 + 28z_5 = 0$

10. $-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (00,01)\big) + \boldsymbol{e}\big(L(\phi_\circ) \rtimes (00,01)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (00,01)\big) \equiv 0 \mod p$:

    $2z_5^4 + 8z_4z_5^3 + 5z_3z_5^3 + 5z_2z_5^3 + 2z_1z_5^3 + 20z_5^3 + 12z_4^2z_5^2 + 15z_3z_4z_5^2 + 15z_2z_4z_5^2 + 6z_1z_4z_5^2 + 60z_4z_5^2 + 3z_3^2z_5^2 + 6z_2z_3z_5^2 + 3z_1z_3z_5^2 + 33z_3z_5^2 + 3z_2^2z_5^2 + 3z_1z_2z_5^2 + 33z_2z_5^2 + 12z_1z_5^2 + 64z_5^2 + 6z_4^3z_5 + 12z_3z_4^2z_5 + 12z_2z_4^2z_5 + 6z_1z_4^2z_5 + 54z_4^2z_5 + 6z_3^2z_4z_5 + 12z_2z_3z_4z_5 + 6z_1z_3z_4z_5 + 63z_3z_4z_5 + 6z_2^2z_4z_5 + 6z_1z_2z_4z_5 + 63z_2z_4z_5 + 24z_1z_4z_5 + 124z_4z_5 + 9z_3^2z_5 + 18z_2z_3z_5 + 9z_1z_3z_5 + 52z_3z_5 + 9z_2^2z_5 + 9z_1z_2z_5 + 52z_2z_5 + 16z_1z_5 + 64z_5 + 6z_4^3 + 12z_3z_4^2 + 12z_2z_4^2 + 6z_1z_4^2 + 36z_4^2 + 6z_3^2z_4 + 12z_2z_3z_4 + 6z_1z_3z_4 + 36z_3z_4 + 6z_2^2z_4 + 6z_1z_2z_4 + 36z_2z_4 + 12z_1z_4 + 48z_4 = 0$

11. $-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (00,10)\big) + \boldsymbol{e}\big(L(\phi_\circ) \rtimes (00,10)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (00,10)\big) \equiv 0 \mod p$:

    $2z_5^4 + 8z_4z_5^3 + 5z_3z_5^3 + 5z_2z_5^3 + 2z_1z_5^3 + 20z_5^3 + 12z_4^2z_5^2 + 15z_3z_4z_5^2 + 15z_2z_4z_5^2 + 6z_1z_4z_5^2 + 60z_4z_5^2 + 3z_3^2z_5^2 + 6z_2z_3z_5^2 + 3z_1z_3z_5^2 + 33z_3z_5^2 + 3z_2^2z_5^2 + 3z_1z_2z_5^2 + 33z_2z_5^2 + 12z_1z_5^2 + 70z_5^2 + 6z_4^3z_5 + 12z_3z_4^2z_5 + 12z_2z_4^2z_5 + 6z_1z_4^2z_5 + 54z_4^2z_5 + 6z_3^2z_4z_5 + 12z_2z_3z_4z_5 + 6z_1z_3z_4z_5 + 63z_3z_4z_5 + 6z_2^2z_4z_5 + 6z_1z_2z_4z_5 +$

$63z_2z_4z_5 + 24z_1z_4z_5 + 136z_4z_5 + 9z_3^2z_5 + 18z_2z_3z_5 + 9z_1z_3z_5 + 64z_3z_5 + 9z_2^2z_5 + 9z_1z_2z_5 + 64z_2z_5 + 22z_1z_5 + 100z_5 + 6z_4^3 + 12z_3z_4^2 + 12z_2z_4^2 + 6z_1z_4^2 + 42z_4^2 + 6z_3^2z_4 + 12z_2z_3z_4 + 6z_1z_3z_4 + 48z_3z_4 + 6z_2^2z_4 + 6z_1z_2z_4 + 48z_2z_4 + 18z_1z_4 + 84z_4 + 6z_3^2 + 12z_2z_3 + 6z_1z_3 + 36z_3 + 6z_2^2 + 6z_1z_2 + 36z_2 + 12z_1 + 48 = 0$

12. $2\boldsymbol{e}\big(M^2(\phi_\circ) \rtimes (00,11)\big) - 4\boldsymbol{e}\big(LM(\phi_\circ) \rtimes (00,11)\big) - 4\boldsymbol{e}\big(RM(\phi_\circ) \rtimes (00,11)\big) + \boldsymbol{e}\big(L^2(\phi_\circ) \rtimes (00,11)\big) + 2\boldsymbol{e}\big(LR(\phi_\circ) \rtimes (00,11)\big) + \boldsymbol{e}\big(R^2(\phi_\circ) \rtimes (00,11)\big) \equiv 0 \mod p$:

$2z_5^5 + 10z_4z_5^4 + 7z_3z_5^4 + 7z_2z_5^4 + 4z_1z_5^4 + 20z_5^4 + 20z_4^2z_5^3 + 28z_3z_4z_5^3 + 28z_2z_4z_5^3 + 16z_1z_4z_5^3 + 80z_4z_5^3 + 8z_3^2z_5^3 + 16z_2z_3z_5^3 + 10z_1z_3z_5^3 + 50z_3z_5^3 + 8z_2^2z_5^3 + 10z_1z_2z_5^3 + 50z_2z_5^3 + 2z_1^2z_5^3 + 26z_1z_5^3 + 70z_5^3 + 18z_4^3z_5^2 + 39z_3z_4^2z_5^2 + 39z_2z_4^2z_5^2 + 24z_1z_4^2z_5^2 + 114z_4^2z_5^2 + 24z_3^2z_4z_5^2 + 48z_2z_3z_4z_5^2 + 30z_1z_3z_4z_5^2 + 147z_3z_4z_5^2 + 24z_2^2z_4z_5^2 + 30z_1z_2z_4z_5^2 + 147z_2z_4z_5^2 + 6z_1^2z_4z_5^2 + 78z_1z_4z_5^2 + 206z_4z_5^2 + 3z_3^3z_5^2 + 9z_2z_3^2z_5^2 + 6z_1z_3^2z_5^2 + 33z_3^2z_5^2 + 9z_2^2z_3z_5^2 + 12z_1z_2z_3z_5^2 + 66z_2z_3z_5^2 + 3z_1^2z_3z_5^2 + 39z_1z_3z_5^2 + 107z_3z_5^2 + 3z_2^3z_5^2 + 6z_1z_2^2z_5^2 + 33z_2^2z_5^2 + 3z_1^2z_2z_5^2 + 39z_1z_2z_5^2 + 107z_2z_5^2 + 6z_1^2z_5^2 + 50z_1z_5^2 + 100z_5^2 + 6z_4^4z_5 + 18z_3z_4^3z_5 + 18z_2z_4^3z_5 + 12z_1z_4^3z_5 + 54z_4^3z_5 + 18z_3^2z_4^2z_5 + 36z_2z_3z_4^2z_5 + 24z_1z_3z_4^2z_5 + 111z_3z_4^2z_5 + 18z_2^2z_4^2z_5 + 24z_1z_2z_4^2z_5 + 111z_2z_4^2z_5 + 6z_1^2z_4^2z_5 + 66z_1z_4^2z_5 + 160z_4^2z_5 + 6z_3^3z_4z_5 + 18z_2z_3^2z_4z_5 + 12z_1z_3^2z_4z_5 + 60z_3^2z_4z_5 + 18z_2^2z_3z_4z_5 + 24z_1z_2z_3z_4z_5 + 120z_2z_3z_4z_5 + 6z_1^2z_3z_4z_5 + 72z_1z_3z_4z_5 + 185z_3z_4z_5 + 6z_2^3z_4z_5 + 12z_1z_2^2z_4z_5 + 60z_2^2z_4z_5 + 6z_1^2z_2z_4z_5 + 72z_1z_2z_4z_5 + 185z_2z_4z_5 + 12z_1^2z_4z_5 + 92z_1z_4z_5 + 172z_4z_5 + 3z_3^3z_5 + 9z_2z_3^2z_5 + 6z_1z_3^2z_5 + 25z_3^2z_5 + 9z_2^2z_3z_5 + 12z_1z_2z_3z_5 + 50z_2z_3z_5 + 3z_1^2z_3z_5 + 29z_1z_3z_5 + 64z_3z_5 + 3z_2^3z_5 + 6z_1z_2^2z_5 + 25z_2^2z_5 + 3z_1^2z_2z_5 + 29z_1z_2z_5 + 64z_2z_5 + 4z_1^2z_5 + 28z_1z_5 + 48z_5 = 0$

Solution: $(z_1, z_2, z_3, z_4, z_5) = (-1, -2, 0, 1, -2)$.

3.5.2 AndOr gate.

1. $|\phi| - 2 \equiv 0 \mod p^3$:

   $z_1 + z_2 + z_3 + z_4 + z_5 + 4 = 0$

2. $\boldsymbol{e}\big(\phi \rtimes (11,11)\big) \not\equiv 0 \mod p$:

   $z_3z_5^3 + z_2z_5^3 + z_1z_5^3 + 2z_5^3 + 2z_3z_4z_5^2 + 2z_2z_4z_5^2 + 2z_1z_4z_5^2 + 4z_4z_5^2 + 2z_3^2z_5^2 + 3z_2z_3z_5^2 + 2z_1z_3z_5^2 + 9z_3z_5^2 + z_2^2z_5^2 + z_1z_2z_5^2 + 6z_2z_5^2 + 3z_1z_5^2 + 8z_5^2 + z_3z_4^2z_5 + z_2z_4^2z_5 + z_1z_4^2z_5 + 2z_4^2z_5 + 2z_3^2z_4z_5 + 3z_2z_3z_4z_5 + 2z_1z_3z_4z_5 + 9z_3z_4z_5 + z_2^2z_4z_5 + z_1z_2z_4z_5 + 6z_2z_4z_5 + 3z_1z_4z_5 + 8z_4z_5 + z_3^3z_5 + 2z_2z_3^2z_5 + z_1z_3^2z_5 + 7z_3^2z_5 + z_2^2z_3z_5 + z_1z_2z_3z_5 + 8z_2z_3z_5 + 3z_1z_3z_5 + 14z_3z_5 + z_2^2z_5 + z_1z_2z_5 + 6z_2z_5 + 2z_1z_5 + 8z_5 \neq 0$

3. $\boldsymbol{e}\big(\phi \rtimes (10,01)\big) \not\equiv 0 \mod p$:

   $z_5^3 + 2z_4z_5^2 + 2z_3z_5^2 + z_2z_5^2 + 4z_5^2 + z_4^2z_5 + 2z_3z_4z_5 + z_2z_4z_5 + 4z_4z_5 + z_3^2z_5 + z_2z_3z_5 + 4z_3z_5 + z_2z_5 + 3z_5 \neq 0$

4. $\boldsymbol{e}\big(\phi \rtimes (10,10)\big) \equiv 0 \mod p$:

$z_5^3 + 2z_4z_5^2 + 2z_3z_5^2 + z_2z_5^2 + 6z_5^2 + z_4^2z_5 + 2z_3z_4z_5 + z_2z_4z_5 + 7z_4z_5 + z_3^2z_5 + z_2z_3z_5 + 7z_3z_5 + 3z_2z_5 + 11z_5 + z_4^2 + 2z_3z_4 + z_2z_4 + 5z_4 + z_3^2 + z_2z_3 + 5z_3 + 2z_2 + 6 = 0$

5. $\boldsymbol{e}\big(\phi \rtimes (01,10)\big) \equiv 0 \mod p$:

$z_3z_5^3 + z_2z_5^3 + z_1z_5^3 + z_5^3 + 2z_3z_4z_5^2 + 2z_2z_4z_5^2 + 2z_1z_4z_5^2 + 2z_4z_5^2 + 2z_3^2z_5^2 + 3z_2z_3z_5^2 + 2z_1z_3z_5^2 + 10z_3z_5^2 + z_2^2z_5^2 + z_1z_2z_5^2 + 8z_2z_5^2 + 6z_1z_5^2 + 8z_5^2 + z_3z_4^2z_5 + z_2z_4^2z_5 + z_1z_4^2z_5 + z_4^2z_5 + 2z_3^2z_4z_5 + 3z_2z_3z_4z_5 + 2z_1z_3z_4z_5 + 11z_3z_4z_5 + z_2^2z_4z_5 + z_1z_2z_4z_5 + 9z_2z_4z_5 + 7z_1z_4z_5 + 9z_4z_5 + z_3^3z_5 + 2z_2z_3^2z_5 + z_1z_3^2z_5 + 10z_3^2z_5 + z_2^2z_3z_5 + z_1z_2z_3z_5 + 13z_2z_3z_5 + 7z_1z_3z_5 + 28z_3z_5 + 3z_2^2z_5 + 3z_1z_2z_5 + 18z_2z_5 + 11z_1z_5 + 19z_5 + z_3z_4^2 + z_2z_4^2 + z_1z_4^2 + z_4^2 + 2z_3^2z_4 + 3z_2z_3z_4 + 2z_1z_3z_4 + 9z_3z_4 + z_2^2z_4 + z_1z_2z_4 + 7z_2z_4 + 5z_1z_4 + 7z_4 + z_3^3 + 2z_2z_3^2 + z_1z_3^2 + 8z_3^2 + z_2^2z_3 + z_1z_2z_3 + 10z_2z_3 + 5z_1z_3 + 19z_3 + 2z_2^2 + 2z_1z_2 + 11z_2 + 6z_1 + 12 = 0$

6. $\boldsymbol{e}\big(\phi \rtimes (01,01)\big) \not\equiv 0 \mod p$:

$z_3z_5^3 + z_2z_5^3 + z_1z_5^3 + z_5^3 + 2z_3z_4z_5^2 + 2z_2z_4z_5^2 + 2z_1z_4z_5^2 + 2z_4z_5^2 + 2z_3^2z_5^2 + 3z_2z_3z_5^2 + 2z_1z_3z_5^2 + 8z_3z_5^2 + z_2^2z_5^2 + z_1z_2z_5^2 + 6z_2z_5^2 + 4z_1z_5^2 + 6z_5^2 + z_3z_4^2z_5 + z_2z_4^2z_5 + z_1z_4^2z_5 + z_4^2z_5 + 2z_3^2z_4z_5 + 3z_2z_3z_4z_5 + 2z_1z_3z_4z_5 + 8z_3z_4z_5 + z_2^2z_4z_5 + z_1z_2z_4z_5 + 6z_2z_4z_5 + 4z_1z_4z_5 + 6z_4z_5 + z_3^3z_5 + 2z_2z_3^2z_5 + z_1z_3^2z_5 + 7z_3^2z_5 + z_2^2z_3z_5 + z_1z_2z_3z_5 + 8z_2z_3z_5 + 4z_1z_3z_5 + 14z_3z_5 + z_2^2z_5 + z_1z_2z_5 + 6z_2z_5 + 3z_1z_5 + 8z_5 \neq 0$

7. $\boldsymbol{e}(\phi \rtimes (00,00)\big) \not\equiv 0 \mod p$:

$2z_5^3 + 4z_4z_5^2 + 4z_3z_5^2 + 2z_2z_5^2 + 12z_5^2 + 2z_4^2z_5 + 4z_3z_4z_5 + 2z_2z_4z_5 + 13z_4z_5 + 2z_3^2z_5 + 2z_2z_3z_5 + 13z_3z_5 + 4z_2z_5 + 18z_5 + z_4^2 + 2z_3z_4 + z_2z_4 + 6z_4 + z_3^2 + z_2z_3 + 6z_3 + 2z_2 + 8 \neq 0$

8. $-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (10,11)\big) + \boldsymbol{e}\big(L(\phi_\circ) \rtimes (10,11)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (10,11)\big) \equiv 0 \mod p$:

$z_5^4 + 3z_4z_5^3 + 3z_3z_5^3 + 2z_2z_5^3 + z_1z_5^3 + 6z_5^3 + 3z_4^2z_5^2 + 6z_3z_4z_5^2 + 4z_2z_4z_5^2 + 2z_1z_4z_5^2 + 12z_4z_5^2 + 3z_3^2z_5^2 + 4z_2z_3z_5^2 + 2z_1z_3z_5^2 + 12z_3z_5^2 + z_2^2z_5^2 + z_1z_2z_5^2 + 7z_2z_5^2 + 3z_1z_5^2 + 11z_5^2 + z_4^3z_5 + 3z_3z_4^2z_5 + 2z_2z_4^2z_5 + z_1z_4^2z_5 + 6z_4^2z_5 + 3z_3^2z_4z_5 + 4z_2z_3z_4z_5 + 2z_1z_3z_4z_5 + 12z_3z_4z_5 + z_2^2z_4z_5 + z_1z_2z_4z_5 + 7z_2z_4z_5 + 3z_1z_4z_5 + 11z_4z_5 + z_3^3z_5 + 2z_2z_3^2z_5 + z_1z_3^2z_5 + 6z_3^2z_5 + z_2^2z_3z_5 + z_1z_2z_3z_5 + 7z_2z_3z_5 + 3z_1z_3z_5 + 11z_3z_5 + z_2^2z_5 + z_1z_2z_5 + 5z_2z_5 + 2z_1z_5 + 6z_5 = 0$

9. $-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (01,11)\big) + \boldsymbol{e}\big(L(\phi_\circ) \rtimes (01,11)\big) + \boldsymbol{e}\big(R(\phi_\circ) \rtimes (01,11)\big) \equiv 0 \mod p$:

$z_3z_5^4 + z_2z_5^4 + z_1z_5^4 + z_5^4 + 3z_3z_4z_5^3 + 3z_2z_4z_5^3 + 3z_1z_4z_5^3 + 3z_4z_5^3 + 3z_3^2z_5^3 + 5z_2z_3z_5^3 + 4z_1z_3z_5^3 + 12z_3z_5^3 + 2z_2^2z_5^3 + 3z_1z_2z_5^3 + 10z_2z_5^3 + z_1^2z_5^3 + 8z_1z_5^3 + 9z_5^3 + 3z_3z_4^2z_5^2 + 3z_2z_4^2z_5^2 + 3z_1z_4^2z_5^2 + 3z_4^2z_5^2 + 6z_3^2z_4z_5^2 + 10z_2z_3z_4z_5^2 + 8z_1z_3z_4z_5^2 + 24z_3z_4z_5^2 + 4z_2^2z_4z_5^2 + 6z_1z_2z_4z_5^2 + 20z_2z_4z_5^2 + 2z_1^2z_4z_5^2 + 16z_1z_4z_5^2 + 18z_4z_5^2 + 3z_3^3z_5^2 + 7z_2z_3^2z_5^2 + 5z_1z_3^2z_5^2 + 21z_3^2z_5^2 + 5z_2^2z_3z_5^2 + 7z_1z_2z_3z_5^2 + 31z_2z_3z_5^2 + 2z_1^2z_3z_5^2 + 21z_1z_3z_5^2 + 44z_3z_5^2 + z_2^3z_5^2 + 2z_1z_2^2z_5^2 + 10z_2^2z_5^2 + z_1^2z_2z_5^2 + 13z_1z_2z_5^2 + 30z_2z_5^2 + 3z_1^2z_5^2 + 19z_1z_5^2 + 26z_5^2 + z_3z_4^3z_5 + z_2z_4^3z_5 + z_1z_4^3z_5 + z_4^3z_5 + 3z_3^2z_4^2z_5 + 5z_2z_3z_4^2z_5 + 4z_1z_3z_4^2z_5 + 12z_3z_4^2z_5 + 2z_2^2z_4^2z_5 + 3z_1z_2z_4^2z_5 + 10z_2z_4^2z_5 + z_1^2z_4^2z_5 + 8z_1z_4^2z_5 + 9z_4^2z_5 + 3z_3^3z_4z_5 + 7z_2z_3^2z_4z_5 + 5z_1z_3^2z_4z_5 + 21z_3^2z_4z_5 + 5z_2^2z_3z_4z_5 + 7z_1z_2z_3z_4z_5 + 31z_2z_3z_4z_5 + 2z_1^2z_3z_4z_5 +$

$21z_1z_3z_4z_5+44z_3z_4z_5+z_2^3z_4z_5+2z_1z_2^2z_4z_5+10z_2^2z_4z_5+z_1^2z_2z_4z_5+13z_1z_2z_4z_5+30z_2z_4z_5+3z_1^2z_4z_5+19z_1z_4z_5+26z_4z_5+z_3^4z_5+3z_2z_3^3z_5+2z_1z_3^3z_5+10z_3^3z_5+3z_2^2z_3^2z_5+4z_1z_2z_3^2z_5+21z_2z_3^2z_5+z_1^2z_3^2z_5+13z_1z_3^2z_5+35z_3^2z_5+z_2^3z_3z_5+2z_1z_2^2z_3z_5+12z_2^2z_3z_5+z_1^2z_2z_3z_5+15z_1z_2z_3z_5+44z_2z_3z_5+3z_1^2z_3z_5+25z_1z_3z_5+50z_3z_5+z_2^3z_5+2z_1z_2^2z_5+9z_2^2z_5+z_1^2z_2z_5+11z_1z_2z_5+26z_2z_5+2z_1^2z_5+14z_1z_5+24z_5=0$

10. $-2\boldsymbol{e}\big(M(\phi_\circ)\rtimes(00,01)\big)+\boldsymbol{e}\big(L(\phi_\circ)\rtimes(00,01)\big)+\boldsymbol{e}\big(R(\phi_\circ)\rtimes(00,01)\big)\equiv 0 \mod p$:

$z_5^4+3z_4z_5^3+3z_3z_5^3+2z_2z_5^3+z_1z_5^3+8z_5^3+3z_4^2z_5^2+6z_3z_4z_5^2+4z_2z_4z_5^2+2z_1z_4z_5^2+16z_4z_5^2+3z_3^2z_5^2+4z_2z_3z_5^2+2z_1z_3z_5^2+16z_3z_5^2+z_2^2z_5^2+z_1z_2z_5^2+9z_2z_5^2+4z_1z_5^2+19z_5^2+z_4^3z_5+3z_3z_4^2z_5+2z_2z_4^2z_5+z_1z_4^2z_5+8z_4^2z_5+3z_3^2z_4z_5+4z_2z_3z_4z_5+2z_1z_3z_4z_5+16z_3z_4z_5+z_2^2z_4z_5+z_1z_2z_4z_5+9z_2z_4z_5+4z_1z_4z_5+19z_4z_5+z_3^3z_5+2z_2z_3^2z_5+z_1z_3^2z_5+8z_3^2z_5+z_2^2z_3z_5+z_1z_2z_3z_5+9z_2z_3z_5+4z_1z_3z_5+19z_3z_5+z_2^2z_5+z_1z_2z_5+7z_2z_5+3z_1z_5+12z_5=0$

11. $-2\boldsymbol{e}\big(M(\phi_\circ)\rtimes(00,10)\big)+\boldsymbol{e}\big(L(\phi_\circ)\rtimes(00,10)\big)+\boldsymbol{e}\big(R(\phi_\circ)\rtimes(00,10)\big)\equiv 0 \mod p$:

$z_5^4+3z_4z_5^3+3z_3z_5^3+2z_2z_5^3+z_1z_5^3+10z_5^3+3z_4^2z_5^2+6z_3z_4z_5^2+4z_2z_4z_5^2+2z_1z_4z_5^2+21z_4z_5^2+3z_3^2z_5^2+4z_2z_3z_5^2+2z_1z_3z_5^2+21z_3z_5^2+z_2^2z_5^2+z_1z_2z_5^2+13z_2z_5^2+6z_1z_5^2+35z_5^2+z_4^3z_5+3z_3z_4^2z_5+2z_2z_4^2z_5+z_1z_4^2z_5+12z_4^2z_5+3z_3^2z_4z_5+4z_2z_3z_4z_5+2z_1z_3z_4z_5+24z_3z_4z_5+z_2^2z_4z_5+z_1z_2z_4z_5+15z_2z_4z_5+7z_1z_4z_5+44z_4z_5+z_3^3z_5+2z_2z_3^2z_5+z_1z_3^2z_5+12z_3^2z_5+z_2^2z_3z_5+z_1z_2z_3z_5+15z_2z_3z_5+7z_1z_3z_5+44z_3z_5+3z_2^2z_5+3z_1z_2z_5+25z_2z_5+11z_1z_5+50z_5+z_4^3+3z_3z_4^2+2z_2z_4^2+z_1z_4^2+9z_4^2+3z_3^2z_4+4z_2z_3z_4+2z_1z_3z_4+18z_3z_4+z_2^2z_4+z_1z_2z_4+11z_2z_4+5z_1z_4+26z_4+z_3^3+2z_2z_3^2+z_1z_3^2+9z_3^2+z_2^2z_3+z_1z_2z_3+11z_2z_3+5z_1z_3+26z_3+2z_2^2+2z_1z_2+14z_2+6z_1+24=0$

12. $2\boldsymbol{e}\big(M^2(\phi_\circ)\rtimes(00,11)\big)-4\boldsymbol{e}\big(LM(\phi_\circ)\rtimes(00,11)\big)-4\boldsymbol{e}\big(RM(\phi_\circ)\rtimes(00,11)\big)+$
$\boldsymbol{e}\big(L^2(\phi_\circ)\rtimes(00,11)\big)+2\boldsymbol{e}\big(LR(\phi_\circ)\rtimes(00,11)\big)+\boldsymbol{e}\big(R^2(\phi_\circ)\rtimes(00,11)\big)\equiv 0 \mod p$:

$z_5^5+4z_4z_5^4+4z_3z_5^4+3z_2z_5^4+2z_1z_5^4+10z_5^4+6z_4^2z_5^3+12z_3z_4z_5^3+9z_2z_4z_5^3+6z_1z_4z_5^3+30z_4z_5^3+6z_3^2z_5^3+9z_2z_3z_5^3+6z_1z_3z_5^3+30z_3z_5^3+3z_2^2z_5^3+4z_1z_2z_5^3+21z_2z_5^3+z_1^2z_5^3+13z_1z_5^3+35z_5^3+4z_4^3z_5^2+12z_3z_4^2z_5^2+9z_2z_4^2z_5^2+6z_1z_4^2z_5^2+30z_4^2z_5^2+12z_3^2z_4z_5^2+18z_2z_3z_4z_5^2+12z_1z_3z_4z_5^2+60z_3z_4z_5^2+6z_2^2z_4z_5^2+8z_1z_2z_4z_5^2+42z_2z_4z_5^2+2z_1^2z_4z_5^2+26z_1z_4z_5^2+70z_4z_5^2+4z_3^3z_5^2+9z_2z_3^2z_5^2+6z_1z_3^2z_5^2+30z_3^2z_5^2+6z_2^2z_3z_5^2+8z_1z_2z_3z_5^2+42z_2z_3z_5^2+2z_1^2z_3z_5^2+26z_1z_3z_5^2+70z_3z_5^2+z_2^3z_5^2+2z_1z_2^2z_5^2+12z_2^2z_5^2+z_1^2z_2z_5^2+15z_1z_2z_5^2+44z_2z_5^2+3z_1^2z_5^2+25z_1z_5^2+50z_5^2+1(z_4^4)z_5+4z_3z_4^3z_5+3z_2z_4^3z_5+2z_1z_4^3z_5+10z_4^3z_5+6z_3^2z_4^2z_5+9z_2z_3z_4^2z_5+6z_1z_3z_4^2z_5+30z_3z_4^2z_5+3z_2^2z_4^2z_5+4z_1z_2z_4^2z_5+21z_2z_4^2z_5+z_1^2z_4^2z_5+13z_1z_4^2z_5+35z_4^2z_5+4z_3^3z_4z_5+9z_2z_3^2z_4z_5+6z_1z_3^2z_4z_5+30z_3^2z_4z_5+6z_2^2z_3z_4z_5+8z_1z_2z_3z_4z_5+42z_2z_3z_4z_5+2z_1^2z_3z_4z_5+26z_1z_3z_4z_5+70z_3z_4z_5+z_2^3z_4z_5+2z_1z_2^2z_4z_5+12z_2^2z_4z_5+z_1^2z_2z_4z_5+15z_1z_2z_4z_5+44z_2z_4z_5+3z_1^2z_4z_5+25z_1z_4z_5+50z_4z_5+z_3^4z_5+3z_2z_3^3z_5+2z_1z_3^3z_5+10z_3^3z_5+3z_2^2z_3^2z_5+4z_1z_2z_3^2z_5+21z_2z_3^2z_5+z_1^2z_3^2z_5+13z_1z_3^2z_5+35z_3^2z_5+z_2^3z_3z_5+2z_1z_2^2z_3z_5+12z_2^2z_3z_5+z_1^2z_2z_3z_5+15z_1z_2z_3z_5+44z_2z_3z_5+3z_1^2z_3z_5+25z_1z_3z_5+50z_3z_5+z_2^3z_5+2z_1z_2^2z_5+9z_2^2z_5+z_1^2z_2z_5+11z_1z_2z_5+26z_2z_5+2z_1^2z_5+14z_1z_5+24z_5=0$

Solution: $(z_1,z_2,z_3,z_4,z_5)=(-2,1,-3,1,-1)$.

The nonzero values $\boldsymbol{e}(\sigma \rtimes (v, v'))$ takes are 2 and 4.

3.5.3 TestEq gate.

1. $|\phi| - 2 \equiv 0 \mod p^3$:

   $z_1 + z_2 + z_3 + z_4 + z_5 + 4 = 0$

2. $\boldsymbol{e}\big(\phi \rtimes (11, 11)\big) \not\equiv 0 \mod p$:

   $2z_3z_5^3+2z_2z_5^3+2z_1z_5^3+4z_5^3+6z_3z_4z_5^2+6z_2z_4z_5^2+6z_1z_4z_5^2+12z_4z_5^2+6z_3^2z_5^2+9z_2z_3z_5^2+6z_1z_3z_5^2+24z_3z_5^2+3z_2^2z_5^2+3z_1z_2z_5^2+15z_2z_5^2+6z_1z_5^2+18z_5^2+6z_3z_4^2z_5+6z_2z_4^2z_5+6z_1z_4^2z_5+12z_4^2z_5+12z_3^2z_4z_5+18z_2z_3z_4z_5+12z_1z_3z_4z_5+48z_3z_4z_5+6z_2^2z_4z_5+6z_1z_2z_4z_5+30z_2z_4z_5+12z_1z_4z_5+36z_4z_5+6z_3^3z_5+12z_2z_3^2z_5+6z_1z_3^2z_5+36z_3^2z_5+6z_2^2z_3z_5+6z_1z_2z_3z_5+39z_2z_3z_5+12z_1z_3z_5+58z_3z_5+3z_2^2z_5+3z_1z_2z_5+19z_2z_5+4z_1z_5+26z_5 \neq 0$

3. $\boldsymbol{e}\big(\phi \rtimes (10, 01)\big) \equiv 0 \mod p$:

   $2z_5^3 + 6z_4z_5^2 + 6z_3z_5^2 + 3z_2z_5^2 + 12z_5^2 + 6z_4^2z_5 + 12z_3z_4z_5 + 6z_2z_4z_5 + 24z_4z_5 + 6z_3^2z_5 + 6z_2z_3z_5 + 24z_3z_5 + 9z_2z_5 + 16z_5 + 6z_4^2 + 12z_3z_4 + 6z_2z_4 + 12z_4 + 6z_3^2 + 6z_2z_3 + 12z_3 = 0$

4. $\boldsymbol{e}\big(\phi \rtimes (10, 10)\big) \equiv 0 \mod p$:

   $2z_5^3 + 6z_4z_5^2 + 6z_3z_5^2 + 3z_2z_5^2 + 12z_5^2 + 6z_4^2z_5 + 12z_3z_4z_5 + 6z_2z_4z_5 + 24z_4z_5 + 6z_3^2z_5 + 6z_2z_3z_5 + 24z_3z_5 + 9z_2z_5 + 22z_5 + 6z_4^2 + 12z_3z_4 + 6z_2z_4 + 18z_4 + 6z_3^2 + 6z_2z_3 + 18z_3 + 6z_2 + 12 = 0$

5. $\boldsymbol{e}\big(\phi \rtimes (01, 10)\big) \equiv 0 \mod p$:

   $2z_3z_5^3 + 2z_2z_5^3 + 2z_1z_5^3 + 2z_5^3 + 6z_3z_4z_5^2 + 6z_2z_4z_5^2 + 6z_1z_4z_5^2 + 6z_4z_5^2 + 6z_3^2z_5^2 + 9z_2z_3z_5^2 + 6z_1z_3z_5^2 + 24z_3z_5^2 + 3z_2^2z_5^2 + 3z_1z_2z_5^2 + 18z_2z_5^2 + 12z_1z_5^2 + 18z_5^2 + 6z_3z_4^2z_5 + 6z_2z_4^2z_5 + 6z_1z_4^2z_5 + 6z_4^2z_5 + 12z_3^2z_4z_5 + 18z_2z_3z_4z_5 + 12z_1z_3z_4z_5 + 48z_3z_4z_5 + 6z_2^2z_4z_5 + 6z_1z_2z_4z_5 + 36z_2z_4z_5 + 24z_1z_4z_5 + 36z_4z_5 + 6z_3^3z_5 + 12z_2z_3^2z_5 + 6z_1z_3^2z_5 + 42z_3^2z_5 + 6z_2^2z_3z_5 + 6z_1z_2z_3z_5 + 51z_2z_3z_5 + 24z_1z_3z_5 + 88z_3z_5 + 9z_2^2z_5 + 9z_1z_2z_5 + 46z_2z_5 + 22z_1z_5 + 52z_5 + 6z_3z_4^2 + 6z_2z_4^2 + 6z_1z_4^2 + 6z_4^2 + 12z_3^2z_4 + 18z_2z_3z_4 + 12z_1z_3z_4 + 42z_3z_4 + 6z_2^2z_4 + 6z_1z_2z_4 + 30z_2z_4 + 18z_1z_4 + 30z_4 + 6z_3^3 + 12z_2z_3^2 + 6z_1z_3^2 + 36z_3^2 + 6z_2^2z_3 + 6z_1z_2z_3 + 42z_2z_3 + 18z_1z_3 + 66z_3 + 6z_2^2 + 6z_1z_2 + 30z_2 + 12z_1 + 36 = 0$

6. $\boldsymbol{e}(\phi \rtimes (01, 01)\big) \equiv 0 \mod p$:

   $2z_3z_5^3+2z_2z_5^3+2z_1z_5^3+2z_5^3+6z_3z_4z_5^2+6z_2z_4z_5^2+6z_1z_4z_5^2+6z_4z_5^2+6z_3^2z_5^2+9z_2z_3z_5^2+6z_1z_3z_5^2+24z_3z_5^2+3z_2^2z_5^2+3z_1z_2z_5^2+18z_2z_5^2+12z_1z_5^2+18z_5^2+6z_3z_4^2z_5+6z_2z_4^2z_5+6z_1z_4^2z_5+6z_4^2z_5+12z_3^2z_4z_5+18z_2z_3z_4z_5+12z_1z_3z_4z_5+48z_3z_4z_5+6z_2^2z_4z_5+6z_1z_2z_4z_5+36z_2z_4z_5+24z_1z_4z_5+36z_4z_5+6z_3^3z_5+12z_2z_3^2z_5+6z_1z_3^2z_5+42z_3^2z_5+6z_2^2z_3z_5+6z_1z_2z_3z_5+51z_2z_3z_5+24z_1z_3z_5+82z_3z_5+9z_2^2z_5+9z_1z_2z_5+40z_2z_5+16z_1z_5+46z_5+6z_3z_4^2+6z_2z_4^2+6z_1z_4^2+6z_4^2+12z_3^2z_4+18z_2z_3z_4+12z_1z_3z_4+36z_3z_4+6z_2^2z_4+6z_1z_2z_4+24z_2z_4+12z_1z_4+24z_4+6z_3^3+12z_2z_3^2+6z_1z_3^2+30z_3^2+6z_2^2z_3+6z_1z_2z_3+30z_2z_3+12z_1z_3+42z_3+6z_2+18=0$

7. $\boldsymbol{e}(\phi \rtimes (00,00)) \not\equiv 0 \mod p$:

$2z_5^3+6z_4z_5^2+6z_3z_5^2+3z_2z_5^2+18z_5^2+6z_4^2z_5+12z_3z_4z_5+6z_2z_4z_5+36z_4z_5+6z_3^2z_5+6z_2z_3z_5+36z_3z_5+15z_2z_5+40z_5+12z_4^2+24z_3z_4+12z_2z_4+36z_4+12z_3^2+12z_2z_3+36z_3+6z_2+15 \neq 0$

8. $-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (10,11)\big)+\boldsymbol{e}\big(L(\phi_\circ) \rtimes (10,11)\big)+\boldsymbol{e}\big(R(\phi_\circ) \rtimes (10,11)\big) \equiv 0 \mod p$:

$2z_5^4+8z_4z_5^3+8z_3z_5^3+5z_2z_5^3+2z_1z_5^3+12z_5^3+12z_4^2z_5^2+24z_3z_4z_5^2+15z_2z_4z_5^2+6z_1z_4z_5^2+36z_4z_5^2+12z_3^2z_5^2+15z_2z_3z_5^2+6z_1z_3z_5^2+36z_3z_5^2+3z_2^2z_5^2+3z_1z_2z_5^2+18z_2z_5^2+6z_1z_5^2+22z_5^2+6z_4^3z_5+18z_3z_4^2z_5+12z_2z_4^2z_5+6z_1z_4^2z_5+30z_4^2z_5+18z_3^2z_4z_5+24z_2z_3z_4z_5+12z_1z_3z_4z_5+60z_3z_4z_5+6z_2^2z_4z_5+6z_1z_2z_4z_5+33z_2z_4z_5+12z_1z_4z_5+40z_4z_5+6z_3^3z_5+12z_2z_3^2z_5+6z_1z_3^2z_5+30z_3^2z_5+6z_2^2z_3z_5+6z_1z_2z_3z_5+33z_2z_3z_5+12z_1z_3z_5+40z_3z_5+3z_2^2z_5+3z_1z_2z_5+13z_2z_5+4z_1z_5+12z_5=0$

9. $-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (01,11)\big)+\boldsymbol{e}\big(L(\phi_\circ) \rtimes (01,11)\big)+\boldsymbol{e}\big(R(\phi_\circ) \rtimes (01,11)\big) \equiv 0 \mod p$:

$2z_3z_5^4+2z_2z_5^4+2z_1z_5^4+2z_5^4+8z_3z_4z_5^3+8z_2z_4z_5^3+8z_1z_4z_5^3+8z_4z_5^3+8z_3^2z_5^3+13z_2z_3z_5^3+10z_1z_3z_5^3+28z_3z_5^3+5z_2^2z_5^3+7z_1z_2z_5^3+22z_2z_5^3+2z_1^2z_5^3+16z_1z_5^3+20z_5^3+12z_3z_4^2z_5^2+12z_2z_4^2z_5^2+12z_1z_4^2z_5^2+12z_4^2z_5^2+24z_3^2z_4z_5^2+39z_2z_3z_4z_5^2+30z_1z_3z_4z_5^2+84z_3z_4z_5^2+15z_2^2z_4z_5^2+21z_1z_2z_4z_5^2+66z_2z_4z_5^2+6z_1^2z_4z_5^2+48z_1z_4z_5^2+60z_4z_5^2+12z_3^3z_5^2+27z_2z_3^2z_5^2+18z_1z_3^2z_5^2+72z_3^2z_5^2+18z_2^2z_3z_5^2+24z_1z_2z_3z_5^2+99z_2z_3z_5^2+6z_1^2z_3z_5^2+60z_1z_3z_5^2+130z_3z_5^2+3z_2^3z_5^2+6z_1z_2^2z_5^2+27z_2^2z_5^2+3z_1^2z_2z_5^2+33z_1z_2z_5^2+76z_2z_5^2+6z_1^2z_5^2+40z_1z_5^2+70z_5^2+6z_3z_4^3z_5+6z_2z_4^3z_5+6z_1z_4^3z_5+6z_4^3z_5+18z_3^2z_4^2z_5+30z_2z_3z_4^2z_5+24z_1z_3z_4^2z_5+66z_3z_4^2z_5+12z_2^2z_4^2z_5+18z_1z_2z_4^2z_5+54z_2z_4^2z_5+6z_1^2z_4^2z_5+42z_1z_4^2z_5+48z_4^2z_5+18z_3^3z_4z_5+42z_2z_3^2z_4z_5+30z_1z_3^2z_4z_5+114z_3^2z_4z_5+30z_2^2z_3z_4z_5+42z_1z_2z_3z_4z_5+165z_2z_3z_4z_5+12z_1^2z_3z_4z_5+108z_1z_3z_4z_5+214z_3z_4z_5+6z_2^3z_4z_5+12z_1z_2^2z_4z_5+51z_2^2z_4z_5+6z_1^2z_2z_4z_5+63z_1z_2z_4z_5+136z_2z_4z_5+12z_1^2z_4z_5+76z_1z_4z_5+118z_4z_5+6z_3^4z_5+18z_2z_3^3z_5+12z_1z_3^3z_5+54z_3^3z_5+18z_2^2z_3^2z_5+24z_1z_2z_3^2z_5+111z_2z_3^2z_5+6z_1^2z_3^2z_5+66z_1z_3^2z_5+166z_3^2z_5+6z_2^3z_3z_5+12z_1z_2^2z_3z_5+60z_2^2z_3z_5+6z_1^2z_2z_3z_5+72z_1z_2z_3z_5+194z_2z_3z_5+12z_1^2z_3z_5+98z_1z_3z_5+206z_3z_5+3z_2^3z_5+6z_1z_2^2z_5+28z_2^2z_5+3z_1^2z_2z_5+32z_1z_2z_5+86z_2z_5+4z_1^2z_5+38z_1z_5+88z_5=0$

10. $-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (00,01)\big)+\boldsymbol{e}\big(L(\phi_\circ) \rtimes (00,01)\big)+\boldsymbol{e}\big(R(\phi_\circ) \rtimes (00,01)\big) \equiv 0 \mod p$:

$2z_5^4+8z_4z_5^3+8z_3z_5^3+5z_2z_5^3+2z_1z_5^3+20z_5^3+12z_4^2z_5^2+24z_3z_4z_5^2+15z_2z_4z_5^2+6z_1z_4z_5^2+60z_4z_5^2+12z_3^2z_5^2+15z_2z_3z_5^2+6z_1z_3z_5^2+60z_3z_5^2+3z_2^2z_5^2+3z_1z_2z_5^2+33z_2z_5^2+12z_1z_5^2+64z_5^2+6z_4^3z_5+18z_3z_4^2z_5+12z_2z_4^2z_5+6z_1z_4^2z_5+54z_4^2z_5+18z_3^2z_4z_5+24z_2z_3z_4z_5+12z_1z_3z_4z_5+108z_3z_4z_5+6z_2^2z_4z_5+6z_1z_2z_4z_5+63z_2z_4z_5+24z_1z_4z_5+124z_4z_5+6z_3^3z_5+12z_2z_3^2z_5+6z_1z_3^2z_5+54z_3^2z_5+6z_2^2z_3z_5+6z_1z_2z_3z_5+63z_2z_3z_5+24z_1z_3z_5+124z_3z_5+9z_2^2z_5+9z_1z_2z_5+52z_2z_5+16z_1z_5+64z_5+6z_4^3+18z_3z_4^2+12z_2z_4^2+6z_1z_4^2+36z_4^2+18z_3^2z_4+24z_2z_3z_4+12z_1z_3z_4+72z_3z_4+6z_2^2z_4+6z_1z_2z_4+36z_2z_4+12z_1z_4+48z_4+6z_3^3+12z_2z_3^2+6z_1z_3^2+36z_3^2+6z_2^2z_3+6z_1z_2z_3+36z_2z_3+12z_1z_3+48z_3=0$

11. $-2\boldsymbol{e}\big(M(\phi_\circ) \rtimes (00,10)\big)+\boldsymbol{e}\big(L(\phi_\circ) \rtimes (00,10)\big)+\boldsymbol{e}\big(R(\phi_\circ) \rtimes (00,10)\big) \equiv 0 \mod p$:

$2z_5^4+8z_4z_5^3+8z_3z_5^3+5z_2z_5^3+2z_1z_5^3+20z_5^3+12z_4^2z_5^2+24z_3z_4z_5^2+15z_2z_4z_5^2+6z_1z_4z_5^2+60z_4z_5^2+12z_3^2z_5^2+15z_2z_3z_5^2+6z_1z_3z_5^2+60z_3z_5^2+3z_2^2z_5^2+3z_1z_2z_5^2+33z_2z_5^2+12z_1z_5^2+70z_5^2+6z_4^3z_5+18z_3z_4^2z_5+$

$12z_2z_4^2z_5+6z_1z_4^2z_5+54z_4^2z_5+18z_3^2z_4z_5+24z_2z_3z_4z_5+12z_1z_3z_4z_5+108z_3z_4z_5+6z_2^2z_4z_5+6z_1z_2z_4z_5+$
$63z_2z_4z_5+24z_1z_4z_5+136z_4z_5+6z_3^3z_5+12z_2z_3^2z_5+6z_1z_3^2z_5+54z_3^2z_5+6z_2^2z_3z_5+6z_1z_2z_3z_5+63z_2z_3z_5+$
$24z_1z_3z_5+136z_3z_5+9z_2^2z_5+9z_1z_2z_5+64z_2z_5+22z_1z_5+100z_5+6z_4^3+18z_3z_4^2+12z_2z_4^2+6z_1z_4^2+$
$42z_4^2+18z_3^2z_4+24z_2z_3z_4+12z_1z_3z_4+84z_3z_4+6z_2^2z_4+6z_1z_2z_4+48z_2z_4+18z_1z_4+84z_4+6z_3^3+$
$12z_2z_3^2+6z_1z_3^2+42z_3^2+6z_2^2z_3+6z_1z_2z_3+48z_2z_3+18z_1z_3+84z_3+6z_2^2+6z_1z_2+36z_2+12z_1+48=0$

12. $2\boldsymbol{e}\big(M^2(\phi_\circ)\rtimes(00,11)\big)-4\boldsymbol{e}\big(LM(\phi_\circ)\rtimes(00,11)\big)-4\boldsymbol{e}\big(RM(\phi_\circ)\rtimes(00,11)\big)+$
$\boldsymbol{e}\big(L^2(\phi_\circ)\rtimes(00,11)\big)+2\boldsymbol{e}\big(LR(\phi_\circ)\rtimes(00,11)\big)+\boldsymbol{e}\big(R^2(\phi_\circ)\rtimes(00,11)\big)\equiv 0 \mod p$:

$2z_5^5+10z_4z_5^4+10z_3z_5^4+7z_2z_5^4+4z_1z_5^4+20z_5^4+20z_4^2z_5^3+40z_3z_4z_5^3+28z_2z_4z_5^3+16z_1z_4z_5^3+80z_4z_5^3+$
$20z_3^2z_5^3+28z_2z_3z_5^3+16z_1z_3z_5^3+80z_3z_5^3+8z_2^2z_5^3+10z_1z_2z_5^3+50z_2z_5^3+2z_1^2z_5^3+26z_1z_5^3+70z_5^3+18z_4^3z_5^2+$
$54z_3z_4^2z_5^2+39z_2z_4^2z_5^2+24z_1z_4^2z_5^2+114z_4^2z_5^2+54z_3^2z_4z_5^2+78z_2z_3z_4z_5^2+48z_1z_3z_4z_5^2+228z_3z_4z_5^2+24z_2^2z_4z_5^2+$
$30z_1z_2z_4z_5^2+147z_2z_4z_5^2+6z_1^2z_4z_5^2+78z_1z_4z_5^2+206z_4z_5^2+18z_3^3z_5^2+39z_2z_3^2z_5^2+24z_1z_3^2z_5^2+114z_3^2z_5^2+$
$24z_2^2z_3z_5^2+30z_1z_2z_3z_5^2+147z_2z_3z_5^2+6z_1^2z_3z_5^2+78z_1z_3z_5^2+206z_3z_5^2+3z_2^3z_5^2+6z_1z_2^2z_5^2+33z_2^2z_5^2+3z_1^2z_2z_5^2+$
$39z_1z_2z_5^2+107z_2z_5^2+6z_1^2z_5^2+50z_1z_5^2+100z_5^2+6(z_4^4)z_5+24z_3z_4^3z_5+18z_2z_4^3z_5+12z_1z_4^3z_5+54z_4^3z_5+$
$36z_3^2z_4^2z_5+54z_2z_3z_4^2z_5+36z_1z_3z_4^2z_5+162z_3z_4^2z_5+18z_2^2z_4^2z_5+24z_1z_2z_4^2z_5+111z_2z_4^2z_5+6z_1^2z_4^2z_5+$
$66z_1z_4^2z_5+160z_4^2z_5+24z_3^3z_4z_5+54z_2z_3^2z_4z_5+36z_1z_3^2z_4z_5+162z_3^2z_4z_5+36z_2^2z_3z_4z_5+48z_1z_2z_3z_4z_5+$
$222z_2z_3z_4z_5+12z_1^2z_3z_4z_5+132z_1z_3z_4z_5+320z_3z_4z_5+6z_2^3z_4z_5+12z_1z_2^2z_4z_5+60z_2^2z_4z_5+6z_1^2z_2z_4z_5+$
$72z_1z_2z_4z_5+185z_2z_4z_5+12z_1^2z_4z_5+92z_1z_4z_5+172z_4z_5+6z_3^4z_5+18z_2z_3^3z_5+12z_1z_3^3z_5+54z_3^3z_5+$
$18z_2^2z_3^2z_5+24z_1z_2z_3^2z_5+111z_2z_3^2z_5+6z_1^2z_3^2z_5+66z_1z_3^2z_5+160z_3^2z_5+6z_2^3z_3z_5+12z_1z_2^2z_3z_5+60z_2^2z_3z_5+$
$6z_1^2z_2z_3z_5+72z_1z_2z_3z_5+185z_2z_3z_5+12z_1^2z_3z_5+92z_1z_3z_5+172z_3z_5+3z_2^3z_5+6z_1z_2^2z_5+25z_2^2z_5+$
$3z_1^2z_2z_5+29z_1z_2z_5+64z_2z_5+4z_1^2z_5+28z_1z_5+48z_5=0$

Solution: $(z_1,z_2,z_3,z_4,z_5)=(-2,-\frac{8}{3},\frac{5}{3},-3,2)$.

The nonzero values $\boldsymbol{e}(\sigma\rtimes(v,v'))$ takes are $\frac{7}{3}$ and $\frac{-8}{3}$.

CHAPTER 4

Linear extensions of height two posets

4.1 Height two posets

In this chapter, we prove Theorem 1.1.6 and Theorem 1.1.7. We begin with a procedure for constructing a height two poset from an arbitrary poset. Let $\mathcal{P} = (X, <)$ be a poset on a set X of n elements $\{x_1, \ldots, x_n\}$. Denote by $\Gamma = (X, E)$ its comparability graph, with oriented edges $(x_i, x_j) \in E$ if $x_i < x_j$ in $\mathcal{P}$. Denote by X' a identical copy of X with elements $\{x'_1, \ldots, x'_n\}$.

Define the poset $\mathcal{Q} = (X \cup X', \prec)$ on $2n$ elements, by having $x_i \prec x'_i$ for all $x_i \in X$, and $x_i \prec x'_j$ for all $x_i < x_j$, with $x_i, x_j \in X$. In particular, the Hasse diagram of $\mathcal{Q}$ consists of $n + |E|$ edges. Note that $\mathcal{Q}$ is a poset of height 2, see Figure 4.2.

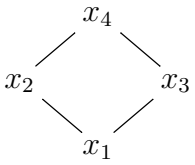


Figure 4.1: The Hasse diagram of a poset $\mathcal{P}$.

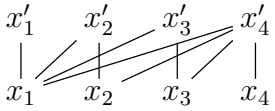


Figure 4.2: Poset $\mathcal{Q}$ associated to poset $\mathcal{P}$.

For every prime p between n and n^2, we construct the modified poset $\mathcal{Q}_p$ by adding, for all i and j satisfying $1 \leq i \leq n$ and $1 \leq j \leq p-2$, the element x_{ij} and the relation $x_{ij} \prec x'_i$. Note that $\mathcal{Q}_p$ is still of height 2 and has pn elements (see Figure 4.3).

We will use the number of linear extensions of $\mathcal{Q}$ and $\mathcal{Q}_p$ to compute the number of linear

extensions of $\mathcal{P}$. Consider first the number $\boldsymbol{e}(\mathcal{Q})$ of linear extensions of $\mathcal{Q}$. Let $A \in \binom{[2n]}{n}$, i.e. A is a n-subset of $[2n] = \{1, 2, \ldots, 2n\}$. Denote by $\boldsymbol{e}_A(\mathcal{Q})$ be the number of linear extensions ℓ of $\mathcal{Q}$ such that $\ell(X') = A$. In other words, $\boldsymbol{e}_A(\mathcal{Q})$ consists of linear extensions that assign values in A to elements of X'. A linear extension ℓ of $\mathcal{Q}$ belongs to exactly one set $\boldsymbol{e}_A(\mathcal{Q})$, so

$$\boldsymbol{e}(\mathcal{Q}) = \sum_{A \in \binom{[2n]}{n}} \boldsymbol{e}_A(\mathcal{Q}).$$

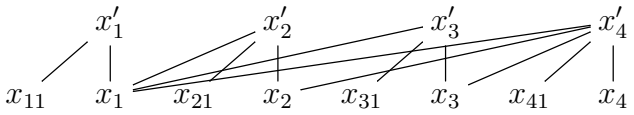


Figure 4.3: $\mathcal{Q}_p$ for $p = 3$.

Lemma 4.1.1. $\boldsymbol{e}(\mathcal{P}) = \boldsymbol{e}_{\{2,4,6,\ldots,2n\}}(\mathcal{Q})$.

Proof. We construct a bijection Φ explictly from $\boldsymbol{e}(\mathcal{P}) \to \boldsymbol{e}_{\{2,4,6,\ldots,2n\}}(\mathcal{Q})$, and give its inverse Ψ. First, given a linear extension ρ of $\mathcal{P}$, let $\Phi(\rho) \in \boldsymbol{e}_{\{2,4,6,\ldots,2n\}}(\mathcal{Q})$ be defined by

$$\Phi(\rho)[x_i] = 2\rho[x_i] - 1$$

and

$$\Phi(\rho)[x'_i] = 2\rho[x_i].$$

Note that $\Phi(\rho)[x_i] < \Phi(\rho)[x'_i]$ for all i, and if $x_i < x_j \in \mathcal{P}$, then

$$\Phi(\rho)[x_i] = 2\rho[x_i] - 1 < 2\rho[x_j] - 1 < 2\rho[x_j] = \Phi(\rho)[x'_j].$$

Thus $\Phi(\rho)$ is indeed a linear extension of $\mathcal{Q}$. Since $\Phi : \boldsymbol{e}(\mathcal{P}) \to \boldsymbol{e}_{\{2,4,6,\ldots,2n\}}(\mathcal{Q})$ is injective, we have $\boldsymbol{e}(\mathcal{P}) \leq \boldsymbol{e}_{\{2,4,6,\ldots,2n\}}(\mathcal{Q})$.

We next give the inverse map Ψ. Given a linear extension $\eta \in \boldsymbol{e}_{\{2,4,6,\ldots,2n\}}(\mathcal{Q})$ we construct a linear extension $\Psi(\eta) \in \boldsymbol{e}(\mathcal{P})$. For every $x_i \in \mathcal{P}$, we set

$$\Psi(\eta)[x_i] = \eta[x'_i]/2.$$

For every $\rho \in \boldsymbol{e}(\mathcal{P})$ we have $\Psi(\Phi(\rho)) = \rho$, by construction. To complete the proof, we must show that $\Phi(\Psi(\eta)) = \eta$ for every $\eta \in \boldsymbol{e}_{\{2,4,6,\ldots,2n\}}(\mathcal{Q})$, and that $\Psi(\eta) \in \boldsymbol{e}(\mathcal{P})$ for every $\eta \in \boldsymbol{e}_{\{2,4,6,\ldots,2n\}}(\mathcal{Q})$, that is, we must show that $\Psi(\eta)$ is a linear extension of $\mathcal{P}$.

Consider $\eta \in \boldsymbol{e}_{\{2,4,6,\ldots,2n\}}(\mathcal{Q})$. We will show that $\eta[x_i] = \eta[x'_i] - 1$ for all i with $1 \leq i \leq n$. Since

$$\eta[X'] = \{2, 4, 6, \ldots, 2n\}$$

we must have

$$\eta[X] = \{1, 3, 5, \ldots, 2n - 1\}.$$

For some $x'_i \in X'$ and for some $x_j \in X$, we have $\eta[x'_i] = 2n$ and $\eta[x_j] = 2n - 1$. For the relation $x_j \prec x'_j$ to be satisfied, we must have $\eta[x_j] < \eta[x'_j]$. We must therefore have $x'_j = 2n$, that is, $i = j$ and $\eta[x_i] = \eta[x'_i] - 1$ for this value of i.

Suppose that for some m we have that $\eta[x_i] = \eta[x'_i] - 1$ for all x_i with $\eta[x_i] > 2m$. We have just proved this statement holds for the case $m = n - 1$. We proceed by induction on $n - m$. There exist $x'_j \in X'$ and $x_k \in X$ with $\eta[x'_j] = 2m$ and $\eta[x_k] = 2m - 1$. Note that we cannot have $\eta[x'_k] > 2m$, since then we would have $\eta[x_k] = \eta[x'_k] - 1 > 2m - 1$ by the induction hypothesis. But $\eta[x'_k] > \eta[x_k] = 2m - 1$, so that we must have $\eta[x'_k] = 2m$ and $\eta[x_j] = \eta[x'_j] - 1$. Thus by induction $\eta[x_i] = \eta[x'_i] - 1$ for all i with $1 \leq i \leq n$, as desired.

Applying this result, we have

$$\Phi(\Psi(\eta))[x'_i] = 2\Psi(\eta)[x_i] = \eta[x'_i]$$

and

$$\Phi(\Psi(\eta))[x_i] = 2\Psi(\eta)[x_i] - 1 = \eta[x'_i] - 1 = \eta[x_i],$$

so that $\Phi(\Psi(\eta)) = \eta$, as desired.

Finally, for every pair $x_i < x_j \in \mathcal{P}$, we have $x_i \prec x'_j \in \mathcal{Q}$. Then $\eta[x_i] < \eta[x'_j]$, so that $\eta[x_i] + 1 = \eta[x'_i] \leq \eta[x'_j]$. Of course η is a bijection so we have $\eta[x'_i] < \eta[x'_j]$. Thus $\Psi(\eta)[x_i] < \Psi(\eta)[x_j]$, and $\Psi(\eta)$ is a linear extension of $\mathcal{P}$. We conclude that Φ and Ψ are inverse maps, which completes the proof. □

The above lemma should be compared with the following result:

Lemma 4.1.2. $\boldsymbol{e}(\mathcal{Q}_p) \equiv (-1)^n \boldsymbol{e}_{\{2,4,6,\ldots,2n\}}(\mathcal{Q}) \mod p$.

Proof. Throughout the proof of this lemma, we will consider colorings of a set of integers. A *coloring* of a set is a function from that set to some list of acceptable colors.

Let $A \in \binom{[2n]}{n}$, and write $A = \{a_1, \ldots, a_n\}$, with $a_1 < a_2 < \cdots < a_n$. A coloring of the set $[pn] = \{1, 2, \ldots, pn\}$ is called A-*compatible* if the following conditions are satisfied:

1. there is a sequence of $2n$ integers $b_1 < \cdots < b_{2n}$ colored black,
2. there are another n colors $C_1, \ldots, C_n$, and $p-2$ integers are colored with each of these colors,
3. all of the elements colored with C_k lie before b_{a_k}.

Let $f_p(A)$ be the number of A-compatible colorings of $[pn]$. Given a linear extension $\eta \in \boldsymbol{e}(\mathcal{Q})$, we write $\boldsymbol{e}_\eta(\mathcal{Q}_p)$ for the number of linear extensions of $\mathcal{Q}_p$ which preserve the ordering on $X \cup X'$ given by η. When $\eta \in \boldsymbol{e}_A(\mathcal{Q})$, we claim that

$$\boldsymbol{e}_\eta(\mathcal{Q}_p) = f_p(A)\big((p-2)!\big)^n.$$

Given a linear extension $\eta \in \boldsymbol{e}_A(\mathcal{Q})$ and a coloring in $f_p(A)$, we can construct $\big((p-2)!\big)^n$ linear extensions $\rho \in \boldsymbol{e}_\eta(\mathcal{Q}_p)$ as follows. Let

$$\rho[x_i] = b_{\eta[x_i]}$$

for all $x_i \in X$, and similarly let

$$\rho[x_i'] = b_{\eta[x_i']}$$

for all $x_i' \in X'$. Thus $\rho[x_i] < \rho[x_j]$ if and only if $\eta[x_i] < \eta[x_j]$. Note that for every integer k, with $1 \le k \le n$, there is some i with $\eta[x_i'] = a_k$ and $\rho[x_i'] = b_{a_k}$. For the $p-2$ elements $x_{ij} \prec x_i'$, assign to $\rho[x_{ij}]$ some permutation of the integers with color C_k. This gives $(p-2)!$ choices for each k, so the total number of linear extensions ρ preserving the ordering η for a fixed coloring is $\big((p-2)!\big)^n$, as desired. Reversing this procedure gives a linear extension for every choice of a linear extension $\eta \in \boldsymbol{e}_A(\mathcal{Q})$ and an A-compatible coloring.

We then have, by Wilson's theorem:

$$\begin{aligned}\boldsymbol{e}(\mathcal{Q}_p) &= \sum_{A\in\binom{[2n]}{n}} \sum_{\eta\in\boldsymbol{e}_A(\mathcal{Q})} \boldsymbol{e}_\eta(\mathcal{Q}_p) \\ &= ((p-2)!)^n \sum_{A\in\binom{[2n]}{n}} \boldsymbol{e}_A(\mathcal{Q})\, f_p(A) \\ &\equiv \sum_{A\in\binom{[2n]}{n}} \boldsymbol{e}_A(\mathcal{Q})\, f_p(A) \mod p.\end{aligned}$$

In an A-compatible coloring of $\{1,2,\ldots,pn\}$, there are $a_k - 1 + k(p-2)$ terms to the left of b_{a_k} colored either black or one of the colors $C_1,\ldots,C_k$. Among these terms, we can choose the position of the elements colored C_k arbitrarily. This gives

$$f_p(A) = \prod_{k=1}^{n} \binom{a_k - 1 + k(p-2)}{p-2}.$$

For $A = \{2,4,6,\ldots,2n\}$, we have $a_k = 2k$, so this becomes

$$f_p\left(\{2,4,6,\ldots,2n\}\right) = \prod_{k=1}^{n} \binom{kp-1}{p-2} \equiv (-1)^n \mod p,$$

by Lucas's theorem. For every other A with $\boldsymbol{e}_A(\mathcal{Q}) \neq 0$, we have $f_p(A) \equiv 0 \mod p$. Indeed, suppose $\boldsymbol{e}_A(\mathcal{Q}) \neq 0$ and consider some $\eta \in \boldsymbol{e}_A(\mathcal{Q})$. Then $\eta[x_i'] = 2n$ for some i, since $\eta[x_i] = 2n$ contradicts $x_i \prec x_i'$. Thus $a_n = 2n$. We proceed by induction on $n-k$. Suppose that

$$(a_{k+1},\ldots,a_n) = (2k+2,\ldots,2n).$$

Then for every integer $j > a_k$ with $j \notin (2k+2,\ldots,2n)$, we have $\eta[x_i] = j$, for some $x_i \in X$. The relation $x_i \prec x_i'$ gives $\eta[x_i'] > \eta[x_i] > a_k$, so that $\eta[x_i'] \in (2k+2,\ldots,2n)$. If we had $a_k < 2k$, we would then have at least $(n-k+2)$ possible values of $j > a_k$, but only $(n-k+1)$ possible values for $\eta[x_i']$ to take in $(2k+2,\ldots,2n)$. Thus either $a_k = 2k$ or $a_k = 2k+1$.

If $a_k = 2k+1$, then

$$\binom{a_k - 1 + kp - 2k}{p-2} = \binom{kp}{p-2}$$

will divide $f_p(A)$. By another application of Lucas's theorem, $\binom{kp}{p-2} \equiv 0 \mod p$, so we have $f_p(A) \equiv 0 \mod p$ unless $a_k = 2k$. This completes the induction, and we conclude $\boldsymbol{e}_A(\mathcal{P}) f_p(A) \equiv 0 \mod p$ unless $A = \{2,4,\ldots,2n\}$. □

We now apply Proposition 3.1.1 to complete the proof.

Proof of Theorem 1.1.6. We make an argument based on the Chinese Remainder Theorem similar to that in [BW91] and Section 3.1. Since $\boldsymbol{e}(\mathcal{P}) \leq n!$, Proposition 3.1.1 together with the Chinese Remainder Theorem shows that computing the residue of $\boldsymbol{e}(\mathcal{P})$ mod p for the primes p with $n \leq p \leq n^2$ is sufficient to determine $\boldsymbol{e}(\mathcal{P})$. The lemmas above show that we can compute the residue of $\boldsymbol{e}(\mathcal{P})$ mod p by computing $\boldsymbol{e}(\mathcal{Q}_p)$. Since #LE is #P-complete, so is #H2LE. □

4.2 Incidence posets

4.2.1 Counting incidence posets

Given a graph $G = (V, E)$, we construct its incidence poset I_G, with elements corresponding to vertices *and* edges of G, with $x < y$ in $\mathcal{P}$ if and only if $x \in E$, $y \in V$ and y is an endpoint of x. We write $\boldsymbol{e}(G)$ for the number of linear extensions of I_G.

Our approach here is similar to our approach in Section 4.1. We produce, given a poset $\mathcal{P}$ and a prime $p > |\mathcal{P}|$, a graph $G_p(\mathcal{P})$ with:

$$\boldsymbol{e}\left(G_p(\mathcal{P})\right) \equiv (-1)^{|\mathcal{P}|} \cdot 8\,\boldsymbol{e}(\mathcal{P}) \mod p.$$

Let $G = (V, E)$ be a graph, with $V = \{x_1, \ldots, x_n\}$, and $\sigma \in S_n$ a permutation. Denote by $\boldsymbol{e}_\sigma(G)$ the number of linear extensions of I_G, which satisfy the following condition: when restricted to V, induce the permutation σ, so that $x_{\sigma^{-1}(1)} \leq x_{\sigma^{-1}(2)} \leq \cdots \leq x_{\sigma^{-1}(n)}$. We have:

$$\boldsymbol{e}(G) = \sum_{\sigma \in S_n} \boldsymbol{e}_\sigma(G).$$

Informally, to compute $\boldsymbol{e}_\sigma(G)$ we visit the vertices of G in the order dictated by σ, accounting for the new edges we meet at each step.

Formally, given a permutation $\sigma \in S_n$, we produce the sequence $\{t_1, \ldots, t_n\}$, where t_i is the number of edges in E with $x_{\sigma^{-1}(i)}$ as an endpoint, and no endpoint $x_{\sigma^{-1}(j)}$ for $j < i$. Let

$\{u_1, \ldots, u_n\}$ be the sequence of partial sums of the t_i's, so that

$$u_k = t_1 + \ldots + t_k.$$

Note that u_k is the total number of edges incident to the set of vertices $x_{\sigma^{-1}(1)}, \ldots, x_{\sigma^{-1}(k)}$.

Let $|E| = m$. Then we call a coloring of the set $\{1, 2, \ldots, m+n\}$ (G, σ)-compatible if the following conditions are satisfied:

1. there is a sequence of n integers $b_1 < \cdots < b_n$ colored black,
2. there are another n colors $C_1, \ldots, C_n$, and t_k integers are colored with the color C_k,
3. all of the elements colored with C_k lie before b_k.

Let $f(G, \sigma)$ be the number of (G, σ)-compatible colorings. In such a coloring, there are $u_k + k - 1$ numbers to the left of b_k colored either black or one of the colors $C_1, \ldots, C_k$. Among these terms, we can choose the position of the elements colored C_k arbitrarily. This gives:

$$f(G, \sigma) = \prod_{k=1}^{n} \binom{u_k + k - 1}{t_k}.$$

A (G, σ)-compatible coloring corresponds to a collection of linear extensions of I_G counted by $\boldsymbol{e}_\sigma(G)$. The values assigned to the t_k new edges at $x_{\sigma^{-1}(k)}$ are given by the numbers colored with C_k, and these values can be assigned in $(t_k)!$ ways, so that we have:

$$\boldsymbol{e}(G) = \sum_{\sigma \in S_n} f(G, \sigma) \prod_{k=1}^{n} (t_k)! = \sum_{\sigma \in S_n} \prod_{k=1}^{n} (t_k)! \binom{u_k + k - 1}{t_k}. \tag{4.2.1}$$

In particular, when we are counting modulo p we can restrict our attention to permutations σ, which have corresponding sequences $\{t_1, \ldots, t_n\}$ with $t_i < p$ for all i. Informally, we want to visit each vertex of G in the order given by σ, deleting the edges incident to each vertex after we visit it, and ensure that no vertex has at least p edges by the time we visit it.

Now we give the actual construction of $G_p(\mathcal{P})$. The first step is to construct a gadget J_p, which is a graph defined as follows. Start with the complete bipartite graph $K_{p-1,p-1}$ on $2p-2$ vertices. Call these vertices $y_1, \ldots, y_{p-1}$ and $z_1, \ldots, z_{p-1}$ and add an additional $p-2$ edges from z_{p-1} to z_i for $1 \leq i < p-1$. Note that each of the y_i's has degree $p-1$ and the z_i's have degree $\geq p$ (see Figure 4.4). We need the following:

Lemma 4.2.1. $\boldsymbol{e}(J_p) \equiv -8 \mod p$.

We defer the proof of this lemma to the end of this section.

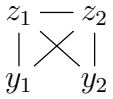


Figure 4.4: J_p for $p = 3$.

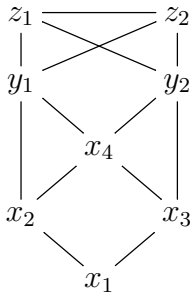


Figure 4.5: $G_p(\mathcal{P})$ for $\mathcal{P}$ as in Figure 4.1 and $p = 3$.

To construct $G_p(\mathcal{P})$, add below J_p the Hasse diagram of $\mathcal{P}$ (treated as an undirected graph). For each element $x \in \mathcal{P}$, let v_x be the number of elements in $\mathcal{P}$ that cover x. Add $p - 1 - v_x$ edges from x to the degree $p - 1$ vertices y_i of J_p in an arbitrarily way (see Figure 4.5).

Theorem 1.1.7 follows immediately from the following:

Lemma 4.2.2. $\boldsymbol{e}(G_p(\mathcal{P})) \equiv (-1)^{|\mathcal{P}|+1} \cdot 8\,\boldsymbol{e}(\mathcal{P}) \mod p$

Proof. Every maximal element of $\mathcal{P}$ has $v_x = 0$, and so is connected to each of the y_i's in J_p. Since $\mathcal{P}$ has at least one maximal element, every element of J_p has degree $\geq p$. Thus every σ which visits a vertex in J_p before visiting every maximal element of $\mathcal{P}$ has a term $t_i \geq p$, so that $\boldsymbol{e}_\sigma(G_p(\mathcal{P})) \equiv 0 \mod p$. Likewise, of these permutations, every permutation σ that visits an element of $\mathcal{P}$ before visiting all of its immediate predecessors has $\boldsymbol{e}_\sigma(G_p(\mathcal{P})) \equiv 0 \mod p$.

Thus we can restrict our count of $\boldsymbol{e}(G_p(\mathcal{P}))$ modulo p to permutations that have as their first n terms a linear extension of $\mathcal{P}$. For these permutations, we have $t_1 = t_2 = \ldots = t_n =$

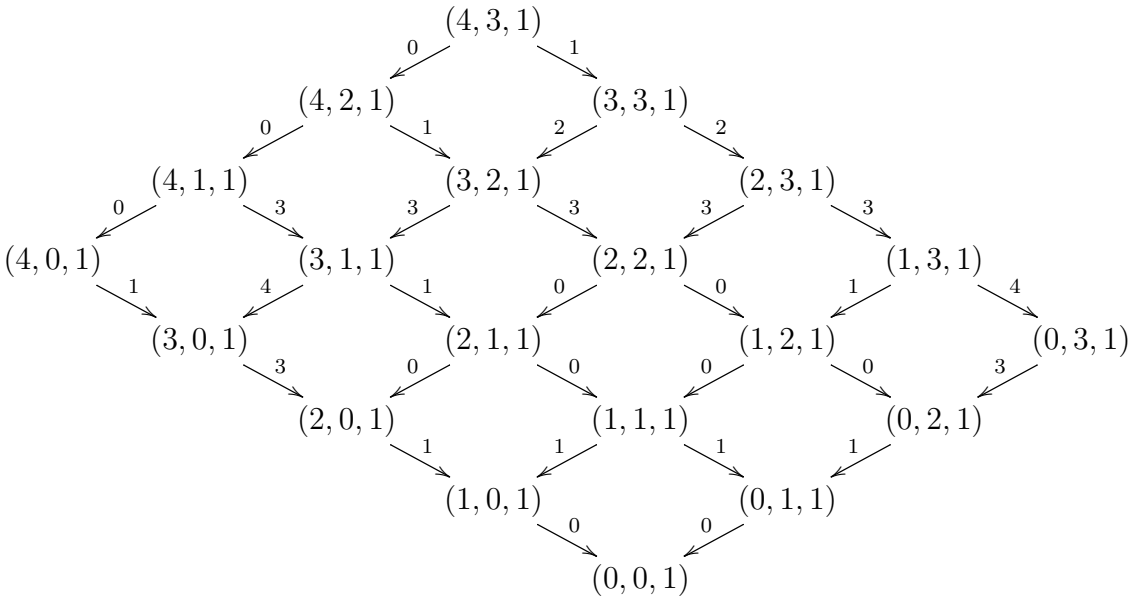


Figure 4.6: The $c = 1$ half of the directed graph $\mathcal{G}'$, with weights, for $p = 5$.

$p-1$, so that $(t_k)! \equiv -1 \mod p$ by Wilson's theorem, and

$$\binom{u_k + k - 1}{t_k} = \binom{kp-1}{p-1} \equiv 1 \mod p.$$

Furthermore, for every $k > n$, we have $t_1 + \ldots + t_k = np - n + (t_{n+1} + \cdots + t_k) + k - 1$, so that

$$\binom{u_k + k - 1}{t_k} \equiv \binom{u_k - u_n + (k-n) - 1}{t_k} \mod p.$$

Now comparing the expressions for $\boldsymbol{e}(G_p(\mathcal{P}))$ and $\boldsymbol{e}(J_p)$ given by (4.2.1), we have

$$\boldsymbol{e}\left(G_p(\mathcal{P})\right) \equiv (-1)^{|\mathcal{P}|}\boldsymbol{e}(\mathcal{P})\boldsymbol{e}(J_p) \mod p,$$

and Lemma 4.2.1 completes the proof. □

Proof of Theorem 1.1.7. Using the same Chinese Remainder Theorem argument we used in Section 3.1 and Section 4.1, the two lemmas above show that computing $\boldsymbol{e}(G_p(P))$ for the primes between $|\mathcal{P}|$ and $|\mathcal{P}|^2$ is sufficient to determine $\boldsymbol{e}(\mathcal{P})$. Since #LE is #P-Complete, so is #IPLE. □

4.2.2 Proof of Lemma 4.2.1

Note that the values t_k and $u_k + k - 1$ in (4.2.1) are both independent of the order in which the previous $k-1$ vertices are visited. They can be computed solely by identifying the vertex $x_{\sigma^{-1}(k)}$ and the collection of vertices $\{x_{\sigma^{-1}(i)}\}_{i<k}$. This motivates the following construction. Recall that the induced subgraphs of a graph G are those formed by deleting some vertices together with all incident edges. Take a directed graph $\mathcal{G}$ whose vertices are the induced subgraphs of J_p and whose edges point from each subgraph to those obtained from it by deleting a single vertex. Attach to each edge the weight

$$(t_k)!\binom{u_k+k-1}{t_k} = (t_k)!\binom{u_k+k-1}{u_k-u_{k-1}}. \tag{4.2.2}$$

Then $\boldsymbol{e}(J_p)$ is equal to the sum of all weighted paths in $\mathcal{G}$ from J_p to the empty subgraph.

Let $J_p(a,b,c)$ be an induced subgraph of J_p with a of the y_i's, b of the z_i's, for $1 \le i < p-1$, and $c = 1$ if $z_{p-1} \in J_p(a,b,c)$, $c = 0$ otherwise, for $0 \le a \le p-1$ and $0 \le b \le p-2$. Since the y_i's, and the z_i's, except for z_{p-1}, are indistinguishable, these subgraphs $J_p(a,b,c)$ are *all* of the induced subgraphs of J_p, up to isomorphism.

We can thus reduce our graph of subgraphs $\mathcal{G}$ to the graph $\mathcal{G}'$ containing only these $2p^2-2p$ vertices. We re-weight the edges from $J_p(a,b,c)$ where a, b or c is reduced by one, by multiplying by a, b or c, respectively. This accounts for the a, b or c choices of vertex to remove. Write $\ell(a,b,c)$ for the value of $u_{k-1}+k-1$ upon reaching $J_p(a,b,c)$, that is, $\ell(a,b,c)$ is the number of vertices and edges that must be deleted from $J_p(p-1,p-2,1)$ to give $J_p(a,b,c)$. Then (4.2.2) gives the weight of the edge from $J_p(a,b,c)$ to $J_p(a-1,b,c)$ in terms of a,b,c and ℓ:

$$a(b+c)!\binom{\ell(a,b,c)+b+c}{b+c} = a(b+c)!\binom{\ell(a-1,b,c)-1}{\ell(a-1,b,c)-\ell(a,b,c)-1}. \tag{4.2.3}$$

The equations for the edges from $J_p(a,b,c)$ to $J_p(a,b-1,c)$ and $J_p(a,b,c-1)$ are the same up to a cyclic permutation of (a,b,c). The total number of edges in J_p is $(p-1)^2+(p-2) = p^2-p-1$. The number of edges in $J_p(a,b,c)$ is $ab+ac+bc$, and we reach $J_p(a,b,c)$ by deleting $(p-1-a)+(p-2-b)+(1-c)$ vertices. We then calculate:

$$\ell(a,b,c) = p^2-p-1-(ab+(a+b)c)+(p-1-a)+(p-2-b)+(1-c)$$

$$\equiv (a+2)(p-b-2) + (c-1)(a+b+2) \mod p.$$

Lemma 4.2.3. *When $c = 1$, $(a+2)(p-b-2) > p$ and $(p-a-2)(b+2) > p$, every path in $\mathcal{G}'$ that visits $J_p(a,b,c)$ has weight zero modulo p.*

Proof. We argue by induction on $(2p-3)-(a+b)$, that is, on the distance in $\mathcal{G}'$ from $J_p(p-1,p-2,1)$ to $J_p(a,b,c)$. When $a = p-1$, $b = p-2$, $c = 1$, the conditions of the lemma are not met, and the statement is true vacuously.

Now suppose that a, b, c satisfy the conditions in this lemma. Then a path that visits $J_p(a,b,c)$ must come from either $J_p(a+1,b,c)$ or $J_p(a,b+1,c)$. If the values $a+1, b, c$ satisfy the conditions in this lemma, we can then apply the induction hypothesis to show that every path through $J_p(a+1,b,c)$ has weight 0 modulo p. In particular, a path that includes the edge from $J_p(a+1,b,c)$ to $J_p(a,b,c)$ has weight 0 modulo p.

On the other hand, suppose that $a+1, b, c$ do not satisfy the conditions in this lemma. Then $(a+3)(p-b-2) > (a+2)(p-b-2) > p$, so we must have $(p-a-3)(b+2) \leq p$. Note that if a or b is greater than or equal to $p-2$, either $(a+2)(p-b-2) \leq 0$ or $(p-a-2)(b+2) \leq 0$. We thus have $a, b < p-2$, so that $(p-a-3)(b+2) = p$ is impossible.

However, when $(p-a-3)(b+2) < p$, since $b < p-2$, we have $(p-a-3)(b-2) >_p (p-a-2)(b+2)$. Thus, $\ell(a+1,b,c) >_p \ell(a,b,c)$, and so by (4.2.3), the edge from $J_p(a+1,b,c)$ to $J_p(a,b,c)$ has weight 0 modulo p. The argument for the edge from $J_p(a,b+1,c)$ to $J_p(a,b,c)$ is the same by symmetry. □

Lemma 4.2.4. *Given a, b with $(b+2)(p-a-2) \leq p$ the edge from $J_p(a,b,1)$ to $J_p(a,b,0)$ has weight 0 unless $a = p-3$ and $b = 0$, $a = p-2$ and $b = 0$ or 1, or $a = p-1$ with b arbitrary. Similarly, given a, b with $(a+2)(p-b-2) \leq p$, the edge from $J_p(a,b,1)$ to $J_p(a,b,0)$ has weight 0 unless $b = p-3$ and $a = 0$, or $b = p-2$ and $a = 0$ or 1.*

Proof. We give the proof of the first statement, since the proof of the second is essentially identical. Permuting (a,b,c) in (4.2.3) to find the weight of the edge from $J_p(a,b,1)$ to $J_p(a,b,0)$, we note that we must have $a+b < p$ and $a+b <_p a+b+\ell(a,b,1)$. Since

$\ell(a,b,1) \equiv (b+2)(p-a-2) \mod p$, this gives:

$$a+b+(b+2)(p-a-2) < p.$$

This implies that

$$p < a+1+\tfrac{3}{b+1} \leq a+4.$$

We conclude that $a > p-4$, and the rest of the lemma follows by elementary case analysis. □

Proof of Lemma 4.2.1. Note that the edges from $J_p(p-1,p-2,1)$ to $J_p(p-1,p-3,1)$ and $J_p(p-1,p-2,0)$ have weight 0 modulo p. Combining this with the previous two lemmas, we conclude that every path in $\mathcal{G}'$ has weight 0 modulo p unless it visits either $J_p(p-2,1,1)$ or $J_p(1,p-2,1)$. We now complete the desired calculation, through repeated applications of (4.2.3), symmetry, and Wilson's theorem:

$$\begin{aligned}
\boldsymbol{e}(J_p(p-1,p-2,1)) &\equiv (p-1)(p-1)!\,\boldsymbol{e}\big(J_p(p-2,p-2,1)\big) \\
&\equiv (p-2)!\,(-1)^{p-3}\Big[\boldsymbol{e}(J_p(p-2,1,1)) + \boldsymbol{e}\big(J_p(p-2,0,1)\big)\Big] \\
&\equiv 2\,\boldsymbol{e}\big(J_p(p-2,1,1)\big) \\
&\equiv 2\,(p-1)!\Big[\boldsymbol{e}\big(J_p(p-2,1,0)\big) + \boldsymbol{e}\big(J_p(p-2,0,1)\big)\Big] \\
&\equiv -4\,\boldsymbol{e}\big(J_p(p-2,0,1)\big) \\
&\equiv -4\,(p-2)!\,\boldsymbol{e}\big(J_p(p-2,0,0)\big) \;-\; 4\,(p-2)\,\boldsymbol{e}\big(J_p(p-3,0,1)\big) \\
&\equiv -4\,\boldsymbol{e}\big(J_p(p-2,0,0)\big) \;+\; 8\binom{p-1}{2}\boldsymbol{e}\big(J_p(p-3,0,0)\big) \\
&\equiv -4\,(p-2)! \;+\; 4\,(p-1)(p-2)(p-3)! \\
&\equiv -8 \mod p.
\end{aligned}$$

This completes the proof. □

4.3 Polytope of modes

Motivated by probabilistic applications, Montúfar and Rauh [MR16] recently defined the *polytope of modes* (G,X), for every simple graph $G=(V,E)$ and independent subset of

vertices $X \subset V$. The polytope (G, X) consists of all functions $p : V \to [0, 1]$ satisfying

$$\sum_{v \in V} p(v) = 1$$

and

$$p(x) \geq p(y)$$

for every pair (x, y) with $x \in X$ and $(x, y) \in E$. From the perspective of probability, the functions p are probability distributions, and the points x are *modes* of the distribution. Montúfar and Rauh proved that

$$\mathrm{vol}\,(G, X) \;=\; \frac{\mathrm{vol}(\Delta^n)}{n!}\, \boldsymbol{e}\big(P_{G,X}\big),$$

where $n = |V|$, $\mathrm{vol}(\Delta^n) = \sqrt{n}/(n-1)!$, and $P_{G,X}$ is a poset constructed from G and X [MR16, Prop. 3] (see also [Sta97] for a strongly related *order polytope*). The poset $P_{G,X}$ is formed by taking the elements of V together with the relation $x < y$ for every pair (x, y) with $x \in X$ and $(x, y) \in E$. This poset has height 2, with vertices in X on one level and $V \setminus X$ on the other. The authors then discuss the problem of computing $\boldsymbol{e}\big(P_{G,X}\big)$.

The following result follows easily from our Theorem 1.1.7.

Proposition 4.3.1. *For every incidence poset I_G of a simple graph $G = (V, E)$, there exists some graph H and some independent set $X \subseteq H$ with $P_{H,X} = I_G$.*

Proof. The desired graph H is the *medial graph* defined as a graph on the set of vertices $V \cup E$. The edges of this graph are pairs (v, e) where v is incident to e in G. This graph is bipartite, so V is an independent set, and we take $X = V$. Now we note that the poset $P_{H,X}$ consists of the set $V \cup E$ together with the relation $v < e$ whenever v is incident to e, so that $P_{H,X} = I_G$, as desired. □

Corollary 4.3.2. *The problem of computing $\boldsymbol{e}\big(P_{G,X}\big)$ is #P-complete.*

Proof. By the proposition, computing $\boldsymbol{e}\big(P_{H,X}\big)$ allows us to compute $\boldsymbol{e}(I_G)$ for any graph G. Applying Theorem 1.1.7 completes the proof. □

Part II

Contingency tables

CHAPTER 5

Contingency tables

5.1 Introduction

Let $\mathbf{a} = (a_1, \ldots, a_m)$, $a_1 \geq \ldots \geq a_m > 0$, and $\mathbf{b} = (b_1, \ldots, b_n)$, $b_1 \geq \ldots \geq b_n > 0$, be two integer sequences with equal sum:

$$\sum_{i=1}^{m} a_i \;=\; \sum_{j=1}^{n} b_j \;=\; N.$$

A *contingency table* with *margins* $(\mathbf{a}, \mathbf{b})$ is an $m \times n$ matrix of non-negative integers whose i-th row sums to a_i and whose j-th column sums to b_j. Recall that we write $\mathcal{T}(\overline{a}, \overline{b})$ for the set of all such matrices, and $\mathrm{T}(\mathbf{a}, \mathbf{b}) := |\mathcal{T}(\overline{a}, \overline{b})|$.

Two central algorithmic questions are sampling from the uniform distribution on $\mathcal{T}(\overline{a}, \overline{b})$ and computing $\mathrm{T}(\mathbf{a}, \mathbf{b})$. We refer to Section 1.3 for a more thorough discussion of the significance and background of these questions.

The algorithm in this chapter is based on a MCMC approach. We define a new *SHM Markov chain*, with three stages which correspond to *splitting* the table into many smaller tables, the *hypergeometric sampling* and *merging* the tables back into one (thus, SHM stands for Split–Hyper-Merge, see below). This Markov chain is substantially different from the Diaconis–Gangolli chain and other Markov chains employed in the literature.

Although more technical, on sparse matrices SHM is by far superior to the Diaconis–Gangolli chain in both theory and practice, as we show in Section 7.5. In both cases (1) and (2) described in § 1.3.1, we obtain $O(\log n)$ mixing times (see below). The proof employs a technical coupling argument. Furthermore, we are able to use the *comparison technique* to obtain polynomial upper bounds of the Diaconis–Gangolli chain which remained open until

now.

We should mention the group theoretic origin of our algorithm. Although it is not transparent from our description and is perhaps cumbersome as a motivation, our algorithm is a special case of the *Burnside process* introduced in [Jer93]. This approach was shown to have rapid mixing in some cases and *torpid mixing* in other cases [GJ02] (cf. Theorem 5.2.8). In fact, one can view our Theorem 5.2.8 below as a more natural example of torpid mixing for the Burnside process; we omit the details.

5.2 Main results

5.2.1 Approximate counting

The following results show that we can approximately count the number of sparse tables:

Theorem 5.2.1 (Small margins)**.** *Fix constants $C_1, C_2, C_3, \alpha > 0$, with $\alpha < \frac{1}{4}$. Let $(\mathbf{a}, \mathbf{b})$ be margins for an $m \times n$ table with $C_1 m < n < C_2 m$ and $0 < a_i, b_j < C_3 n^\alpha$. Then there exists a FPRAS to approximately count the number of tables $T(\mathbf{a}, \mathbf{b})$.*

Here by FPRAS we mean *fully polynomial randomized approximation scheme* (see e.g. [Jer03]). Note that the margins in the theorem are allowed to have unbounded ratios. When this is not allowed, we can approximately count the number of sparse tables with even larger margins:

Theorem 5.2.2 (Smooth margins)**.** *Fix constants $C_1, C_2, C_3, C_4, \alpha > 0$, with $\alpha < 1$. Let $(\mathbf{a}, \mathbf{b})$ be the margins of an $m \times n$ table with $C_1 m < n < C_2 m$ and $C_3 n^\alpha < a_i, b_j < C_4 n^\alpha$. Then there exists a FPRAS to approximately count the number of tables $T(\mathbf{a}, \mathbf{b})$.*

Both theorems are proved by using a new *SHM Markov Chain* we describe in the next section. We conclude with a simple but attractive special case.

Corollary 5.2.3 (Counting magic squares)**.** *Fix $\epsilon, \delta > 0$. The number $\boldsymbol{t}(n, K)$ of $n \times n$ magic squares with row/column sums K, where $K = O\big(n^{1-\epsilon}\big)$, can be approximated within factor $(1 \pm \delta)$ in time polynomial in n and $\log 1/\delta$.*

Remark 5.2.4. Note that if the corollary is extended to all $K < (n-1)^2$, then by Lagrange interpolation one can perhaps approximate the volume of the Birkhoff polytope B_n (see above). This would be of independent interest.

5.2.2 Mixing time of the SHM chain

Let π be the uniform distribution on $\mathcal{T}(\mathbf{a}, \mathbf{b})$ and ω be the hypergeometric distribution on $\mathcal{T}(\mathbf{a}, \mathbf{b})$ defined to be proportional to the inverse of the product of factorials of entries:

$$\omega(X) \,=\, C \cdot \left(\prod_{i,j} (x_{ij})! \right)^{-1} \qquad \text{where} \quad X = \big(x_{ij}\big) \in \mathcal{T}(\mathbf{a}, \mathbf{b}).$$

We denote by $P^t(X)$ the distribution after t steps of the SHM Markov chain starting with the table X. Similarly, denote by $P^t(\omega)$ the distribution after t steps, when we begin our Markov chain with a table chosen at random from the hypergeometric distribution.

Theorem 5.2.5 (Constant margins)**.** *Fix $K > 0$. Let $(\mathbf{a}, \mathbf{b})$ be margins for an $m \times n$ table X with constant margins $a_1, b_1 < K$. Then there exists a constant $C > 0$ independent of m and n, s.t.*

$$\|P^t(X) - \pi\|_{TV} \,\le\, \frac{1}{4} \qquad \textit{for all} \quad t > C.$$

This special case is subsumed by the following two results. We prove it first as it is a stepping stone towards these extensions. The proofs of this and other results are given in Section 5.6.

Theorem 5.2.6 (Small margins)**.** *Fix constants $C_1, C_2, C_3, \alpha > 0$, with $\alpha < \frac{1}{4}$. Let X be an $m \times n$ table with nonzero row and column sums and $C_1 m < n < C_2 m$. Let the row and column sums of X satisfy $a_i, b_j < C_3 n^\alpha$. Then there exists constants $C, C' > 0$ independent of m and n, s.t.*

$$\|P^t(\omega) - \pi\|_{TV} \,\le\, \frac{1}{4} \qquad \textit{for all} \quad t > C \log n$$

and

$$\|P^t(X) - \pi\|_{TV} \,\le\, \frac{1}{4} \qquad \textit{for all} \quad t > C' n^\alpha.$$

The second part of the theorem gives a weaker bound than the first since we are starting the MC from distribution ω rather than the worst case starting contingency table $X \in \mathcal{T}(\overline{a}, \overline{b})$.

Theorem 5.2.7 (Smooth margins). *Fix constants $C_1, C_2, C_3, C_4, \alpha > 0$, with $\alpha < 1$. Let X be an $m \times n$ table with nonzero row and column sums and $C_1 m < n < C_2 m$. Let the row and column sums of X satisfy $C_3 n^\alpha < a_i, b_j < C_4 n^\alpha$. Then there exists a constant C independent of m and n, s.t.*

$$\|P^t(\omega) - \pi\|_{TV} \leq \frac{1}{4} \qquad \text{for all} \quad t > C \log n.$$

The following result shows that for superpolynomial margins the SHM chain does not mix rapidly.

Theorem 5.2.8 (Torpid mixing). *Fix a constant L. Let A be a $2 \times n$ table with column sums $2L$ and row sums Ln. Then*

$$\|P^t(\omega) - \pi\|_{TV} > \frac{1}{4} \qquad \text{for all} \quad t < \frac{L}{18 n \log L}.$$

This result shows that, for tables whose entries are large compared to the size of the table, the mixing time can grow with the size of the entries. For example, for $L = \exp \Omega(n)$, the mixing time is exponential as opposed to polynomial in [C+06].

5.2.3 Mixing time of the Diaconis–Gangolli chain

Here we apply our results to obtain upper bounds on mixing time for the *lazy Diaconis–Gangolli chain* which at each step stays put with probability 1/2. Denote by $Q^t(X)$ the distribution after t steps of this MC starting with the table X.

Theorem 5.2.9 (Constant margins). *Fix $K > 0$. Let $(\mathbf{a}, \mathbf{b})$ be margins for an $m \times n$ table X with constant margins $a_1, b_1 < K$. Then there exists $C > 0$ independent of m and n, s.t.*

$$\|Q^t(X) - \pi\|_{TV} \leq \frac{1}{4} \qquad \text{for all} \quad t > C n^7 \log n.$$

In other words, the mixing time for the Diaconis–Gangolli Markov chain is $O(n^7 \log n)$. No subexponential bounds were known in this case. It is easy to see that the mixing time is $\Omega(n^3 \log n)$ for the case of $m = n$ and $K = 1$, see [DS95]. In fact, Diaconis and Saloff-Coste conjecture an upper bound $O(n^3 \log n)$ for all constant margins.

Theorem 5.2.10 (Small margins)**.** *Fix constants $C_1, C_2, C_3, \alpha > 0$, with $\alpha < \frac{1}{4}$. Let X be an $m \times n$ table with nonzero row and column sums and $C_1 m < n < C_2 m$. Let the row and column sums of X satisfy $a_i, b_j < C_3 n^\alpha$. Then there exists $C > 0$ independent of m and n, s.t.*

$$\|Q^t(X) - \pi\|_{TV} \,\le\, \frac{1}{4} \qquad \text{for all} \quad t > C n^{4\alpha+7} \log n\,.$$

Remark 5.2.11. Viewing contingency tables as integer points in the transportation polytope suggests that the *hit-and-run* version of the Diaconis–Gangolli chain mixes in polynomial time for all margins (cf. [Lov99, LV06]). Torpid mixing Theorem 5.2.8 implies that such results cannot be obtained via comparison with the SHM Markov chain.

Note also that the above results are in sharp contrast with [BBR11] which considers the same chain with 0–1 restrictions on the entries; the authors show torpid mixing in some cases.

5.3 The Algorithm

5.3.1 The setup

Let $X = \big(x_{ij}\big)$, $1 \le i \le m$, $1 \le j \le n$. Denote by $\alpha_i(X)$ and $\beta_j(X)$ the *marginal sums* of X defined as

$$\alpha_i(X) \,:=\, \sum_{j=1}^{n} x_{ij}\,, \qquad \beta_j(X) \,:=\, \sum_{i=1}^{m} x_{ij}\,.$$

We use $\mathbf{a}$ to denote the sequence $(a_1, a_2, \ldots)$. In particular, $\alpha(X)$ and $\beta(X)$ are vectors of margins of X.

Let $\mathbb{S}(n)$ be the permutation group on n elements. For all $s \in \mathbb{S}(n)$ denote by $\mathrm{M}(s)$ the 0-1 matrix of size n corresponding to s.

Each step of the Markov chain consists of three subroutines: Split, Hyper and Merge. We describe the procedure in detail below. The second subroutine samples from the Fisher–Yates (hypergeometric) distribution on tables with fixed row and column sums. This fast hypergeometric sampling algorithm has been known for some time, see e.g. [Eve92]. Our version follows [DS98].

5.3.2 Construction of the SHM Markov chain

In the Split subroutine, we construct a sequence of $m \times n$ tables (Y_k) so that the entrywise sum of kY_k is equal to the original table X. We do this by generating a random permutation for each entry of X and finding the partition associated to the cycle decomposition of that permutation. In the Hyper subroutine, we transform each table Y_k into a table Z_k with the same margins by generating a random permutation matrix and subdividing the matrix into blocks based on the marginal sums. Then in the Merge subroutine we add together kZ_k entrywise to produce a new table X'.

We now formally describe the SHM Markov chain by the following pseudocode. The subroutines Split and Hyper are given in separate blocks while the Merge subroutine is given in the last line of the main block.

Step of SHM

Input: $m \times n$ matrix $X = (x_{ij})$ with marginal sums $\mathbf{a}$, $\mathbf{b}$.

Output: $m \times n$ matrix $X' = (x'_{ij})$ with marginal sums $\mathbf{a}$, $\mathbf{b}$.

begin

Split $X \longrightarrow \big(Y_1, Y_2, Y_3, \ldots\big)$, where Y_i are $m \times n$ matrices

and $X = Y_1 + 2Y_2 + 3Y_3 + \ldots$

for all $k = 1, 2, \ldots$ **do**

Hyper $Y_k \longrightarrow Z_k$, where $\alpha(Y_k) = \alpha(Z_k)$, $\beta(Y_k) = \beta(Z_k)$

end

MERGE $X' \longleftarrow Z_1 + 2Z_2 + 3Z_3 + \ldots$

end

SPLIT

Input: $m \times n$ matrix $X = (x_{ij})$

Output: sequence of $m \times n$ matrices $(Y_1, Y_2, \ldots)$, where $Y_k = (y_{ijk})$

and $X = Y_1 + 2Y_2 + 3Y_3 + \ldots$

begin

for $i = 1$ **to** m **do**

for $j = 1$ **to** n **do**

sample uniform random permutation $s_{ij} \in \mathbb{S}(x_{ij})$

for all $k \geq 1$ **do**

$y_{ijk} \leftarrow$ number of k-cycles in s_{ij}

end

end

end

end

HYPER

Input: $m \times n$ matrix $Y = (y_{ij})$ with marginal sums $\mathbf{p}$, $\mathbf{q}$.

Output: $m \times n$ matrix $Z = (z_{ij})$ with marginal sums $\mathbf{p}$, $\mathbf{q}$.

begin

$N \leftarrow p_1 + \ldots + p_m = q_1 + \ldots + q_n$

sample uniform random permutation $s \in \mathbb{S}(N)$

M$(s) \leftarrow$ permutation matrix of s of size $N \times N$

B $\leftarrow m \times n$ block matrix with blocks B_{ij} of size $p_i \times q_j$

for all $1 \le i \le m$, $1 \le j \le n$

for $i = 1$ **to** m **do**

for $j = 1$ **to** n **do**

$z_{ij} \leftarrow$ sum of entries in $\mathrm{M}(s) \cap B_{ij}$

end

end

end

5.3.3 Analysis

We can now state the foundational result for this chapter. It implies that SHM Markov chain can in principle be used to sample (nearly) uniformly at random from $\mathcal{T}(\overline{a}, \overline{b})$.

Theorem 5.3.1. *The SHM Markov chain defined by the algorithm above converges to the uniform distribution π on $\mathcal{T}(\overline{a}, \overline{b})$.*

To finish the analysis of the SHM chain, we will also need the following technical result.

Lemma 5.3.2. *One step of the SHM Markov chain can be executed in $O(mnN \log N)$ time.*

This is sufficient for our purposes, since $N = o(mn)$ in theorems 5.2.1 and 5.2.2.

5.3.4 Proof of theorems 5.2.1 and 5.2.2

We start with a contingency table X, run the Hyper routine once, and Step of SHM for t steps. The resulting contingency table has distribution $P^t(\omega)$. Applying the first part of Theorem 5.2.6 and Theorem 5.2.7 gives $t = O(\log n)$ mixing time to sample from $\mathcal{T}(\overline{a}, \overline{b})$ in both cases. Since the cost of each step of SHM Markov chain is polynomial in n, we obtain both theorems 5.2.1 and 5.2.2, respectively. □

Remark 5.3.3. The logarithmic mixing time $t = O(\log n)$ is really surprising and more indicative of the actual performance of the algorithm than the theorems are suggesting. We explore the empirical performance of the algorithm in Chapter 7.

Note that formally we prove $O(\log n)$ mixing time only when SHM chain starts from the hypergeometric distribution ω rather than from the worst case starting point. Similar "average case mixing times" have been studied in other context as well (cf. [Gey92, LW98]). As the second part of Theorem 5.2.6 shows, the total variation mixing time is polynomial in the small margin case.

5.4 Proof of the uniform stationary distribution

In this section we prove Theorem 5.3.1. First, observe that choosing the identity permutation throughout in the SPLIT step gives $Y_1 = X$ and $Y_2 = Y_3 = \cdots = \mathbf{0}$. The HYPER step then samples a random contingency table Z_1 from the hypergeometric distribution, where Z_1 has the same margins as X, and the MERGE step sets $X' = Z_1$. It is therefore possible to move from any table to any other table in a single step. Thus, the SHM Markov chain is irreducible and aperiodic.

It remains to show that its stationary distribution is uniform on contingency tables. To do this, we show that the uniform distribution is stationary after one step of the Markov chain, or equivalently that the transition matrix is doubly stochastic. Let X^t be the table that occurs in our Markov chain after t steps, beginning with some unspecified initial distribution. We must show:

$$\sum_{X \in \mathcal{T}(\overline{a},\overline{b})} \mathbf{P}\big(X^{t+1} = W \,|\, X^t = X\big) \;=\; 1 \qquad \text{for every} \quad W \in \mathcal{T}(\overline{a},\overline{b}). \tag{5.4.1}$$

Our proof works by re-writing this probability in terms of the tables produced by the subroutines of the SHM chain and changing the order of summation. To emphasize the intermediate tables produced during the subroutines of the SHM chain, we introduce new notation. We write

$$\mathbf{P}\Big(X \xrightarrow{\text{SHM}} W\Big) := \mathbf{P}(X^{t+1} = W | X^t = X),$$

so that (5.4.1) becomes

$$\sum_{X \in \mathcal{T}(\overline{a},\overline{b})} \mathbf{P}\Big(X \xrightarrow{\text{SHM}} W\Big) = 1. \tag{5.4.2}$$

We write similarly

$$\mathbf{P}\Big(X \xrightarrow{\text{Split}} (Y_k)\Big)$$

for the probability that a table X produces the sequence of tables $(Y_k) = (Y_1, Y_2, \dots)$ during the Split step, and

$$\mathbf{P}\Big((Y_k) \xrightarrow{\text{Hyper}} (Z_k)\Big)$$

for the probability that the sequence (Y_k) produces the sequence $(Z_k) = (Z_1, Z_2, \dots)$ during the Hyper step. We now proceed with the proof.

The probability that a table X produces another table W after one step of the SHM chain can be expressed by summing over all possible intermediate states (Y_k) and (Z_k) that can arise during the Split and Hyper steps.

$$\mathbf{P}\Big(X \xrightarrow{\text{SHM}} W\Big) = \sum_{(Y_k)} \sum_{(Z_k)} \mathbf{P}\Big(X \xrightarrow{\text{Split}} (Y_k)\Big) \cdot \mathbf{P}\Big((Y_k) \xrightarrow{\text{Hyper}} (Z_k)\Big),$$

where the sum is taken over all ordered pairs $\big((Y_k), (Z_k)\big)$ with $\textsc{Merge}(Z_k) = W$. Substituting into (5.4.2), we desire to show

$$\sum_{X} \sum_{(Y_k)} \sum_{(Z_k)} \mathbf{P}\Big(X \xrightarrow{\text{Split}} (Y_k)\Big) \cdot \mathbf{P}\Big((Y_k) \xrightarrow{\text{Hyper}} (Z_k)\Big) = 1. \tag{5.4.3}$$

The sum is taken over triples $\big(X, (Y_k), (Z_k)\big)$ with $\textsc{Merge}(Z_k) = W$. We next re-express these probabilities in terms of the entries of the tables.

For a fixed sequence of tables (Y_k), there is exactly one X that can produce (Y_k) during the Split step, that is, $X = \textsc{Merge}(Y_k)$. Write x_{ij} for the entries of X, and y_{ijk} for the corresponding entries of Y_k. The probability that a random permutation of x_{ij} has y_{ijk} cycles of length k is

$$\prod_{k \geq 1} \frac{1}{k^{y_{ijk}}\,(y_{ijk})!}. \tag{5.4.4}$$

Taking the product of (5.4.4) over i and j and setting

$$N_k := \sum_{i,j} y_{ijk}$$

gives

$$\mathbf{P}\Big(X \xrightarrow{\text{Split}} (Y_k)\Big) = \prod_{k \geq 1} \frac{1}{k^{N_k}} \cdot \prod_{i,j,k} \frac{1}{(y_{ijk})!}. \tag{5.4.5}$$

For every k, the probability $\mathbf{P}\big(Y_k \xrightarrow{\text{HYPER}} Z_k\big)$ is independent of Y_k as long as Y_k and Z_k have the same marginal sums (p_{ik}, q_{jk}). Let Z_k have entries z_{ijk}. By the properties of the hypergeometric distribution,

$$\mathbf{P}\big(Y_k \xrightarrow{\text{HYPER}} Z_k\big) \;=\; \frac{\prod_i (p_{ik})! \cdot \prod_j (q_{jk})!}{(N_k)! \prod_{i,j} (z_{ijk})!}\,. \tag{5.4.6}$$

We write $(\mathbf{p}, \mathbf{q})$ for the sequence of marginal sums of (Y_k), and write $\mathbf{y}\,|(\mathbf{p}, \mathbf{q})$ for the set of all tables (Y_k) with marginal sums $(\mathbf{p}, \mathbf{q})$. We write $\mathbf{z}\,|(\mathbf{p}, \mathbf{q})$ for the set of all tables (Z_k) with marginal sums $(\mathbf{p}, \mathbf{q})$ *and* $\text{MERGE}(Z_k) = W$. Substituting (5.4.5) and (5.4.6) into (5.4.3) gives

$$\begin{aligned}
&\sum_X \sum_{(Y_k)} \sum_{(Z_k)} \mathbf{P}\Big(X \xrightarrow{\text{SPLIT}} (Y_k)\Big) \cdot \mathbf{P}\Big((Y_k) \xrightarrow{\text{HYPER}} (Z_k)\Big) \\
&\quad = \sum_{(\mathbf{p},\mathbf{q})} \sum_{\mathbf{y}\,|\,(\mathbf{p},\mathbf{q})} \sum_{\mathbf{z}\,|\,(\mathbf{p},\mathbf{q})} \left(\prod_{k\geq 1} \frac{1}{k^{N_k}} \cdot \prod_{i,j,k} \frac{1}{(y_{ijk})!} \cdot \frac{\prod_{i,k}(p_{ik})! \prod_{j,k}(q_{jk})!}{\prod_{i,j,k}(z_{ijk})! \prod_k (N_k)!} \right),
\end{aligned}$$

where the outermost sum is over all possible margins $(\mathbf{p}, \mathbf{q})$ compatible with $\mathcal{T}(\overline{a}, \overline{b})$. Rearranging the y_{ijk} and z_{ijk} terms in the product and changing the order of summation gives

$$= \sum_{(\mathbf{p},\mathbf{q})} \left(\sum_{\mathbf{z}\,|\,(\mathbf{p},\mathbf{q})} \prod_{k\geq 1} \frac{1}{k^{N_k}} \cdot \prod_{i,j,k} \frac{1}{(z_{ijk})!} \cdot \sum_{\mathbf{y}\,|\,(\mathbf{p},\mathbf{q})} \frac{\prod_{i,k}(p_{ik})! \prod_{j,k}(q_{jk})!}{\prod_{i,j,k}(y_{ijk})! \prod_k (N_k)!} \right).$$

The innermost sum is the probability of producing a sequence of tables (Y_k) during the HYPER step, summed over all possible sequences (Y_k) that could be produced. It therefore evaluates to 1, which simplifies the expression considerably.

$$= \sum_{(\mathbf{p},\mathbf{q})} \sum_{\mathbf{z}\,|\,(\mathbf{p},\mathbf{q})} \left(\prod_{k\geq 1} \frac{1}{k^{N_k}} \cdot \prod_{i,j,k} \frac{1}{(z_{ijk})!} \right) = \sum_{(Z_k)} \left(\prod_{k\geq 1} \frac{1}{k^{N_k}} \cdot \prod_{i,j,k} \frac{1}{(z_{ijk})!} \right),$$

where the final sum is over all tables (Z_k) with $\text{MERGE}(Z_k) = W$. But, applying (5.4.5), this becomes

$$= \sum_{(Z_k)} \mathbf{P}\Big(W \xrightarrow{\text{SPLIT}} (Z_k)\Big) = 1,$$

since the sequences (Z_k) of tables with $\text{MERGE}(Z_k) = W$ are precisely those that can be produced by applying the SPLIT step to W. This completes the proof of Theorem 5.3.1. □

5.5 Making of a coupling

5.5.1 Notation

Throughout the proofs we let X be an $m \times n$ table with margins $(\mathbf{a}, \mathbf{b})$. We use (Y_k) for tables produced by SPLIT and (Z_k) for tables produced by HYPER. We use $(\mathbf{p}, \mathbf{q})$ for the margins of Y_k or Z_k.

For two tables $X, \widehat{X}$ we write

$$d(X, \widehat{X}) = \sum_{i,j} |x_{ij} - \widehat{x}_{ij}|.$$

We write $\mathbf{0}$ for the table whose entries are all 0, so that

$$d(X, \mathbf{0}) = \sum_{i,j} |x_{ij}| = \sum_{i,j} x_{ij},$$

since all entries are non-negative.

For our coupling results, we write $(P, \widehat{P})$ for the coupled distributions, and X^t and $\widehat{X}^t$ for the tables occurring at $P^t(\omega)$ and $\widehat{P}^t(\omega)$, respectively. We likewise write Y_k^t and $\widehat{Y}_k^t$ for the tables produced by the SPLIT steps and Z_k^t and $\widehat{Z}_k^t$ for the tables produced by the HYPER steps in $P^t(\omega)$ and $\widehat{P}^t(\omega)$, respectively. Similarly, write $P^t(X^0)$ and $\widehat{P}^t(\widehat{X}^0)$ for probability distribution of the Markov chain starting at X^0 and $\widehat{X}^0$, respectively. We write the entries of these tables as y_{ijk}^t, $\widehat{y}_{ijk}^t$, z_{ijk}^t and $\widehat{z}_{ijk}^t$, respectively. We write

$$p_{ik}^t := \sum_{j=1}^{n} y_{ijk}^t$$

for the row sums of Y_k^t,

$$q_{jk}^t := \sum_{i=1}^{m} y_{ijk}^t$$

for the column sums of Y_k^t, and

$$N_k^t := \sum_{i=1}^{m} p_{ik}^t$$

for the total sum of Y_k^t. Finally, we take

$$\gamma_{ijk}^t := \frac{p_{ik}^t q_{jk}^t}{N_k^t},$$

when $N_k^t \neq 0$, and $\gamma_{ijk}^t := 0$ otherwise. We define $\widehat{p_{ik}^t}$, $\widehat{q}_{jk}^t$, $\widehat{N}_k^t$ and $\widehat{\gamma}_{ijk}^t$ similarly. Let a_{ik}^t be the number of k's in column i of $P^t(\omega)$. Define $\widehat{\mathrm{a}}_{ik}^t$, b_{jk}^t and $\widehat{\mathrm{b}}_{jk}^t$ similarly. As lower bounds on the error $d(Y_k^t, \widehat{Y}_k^t)$, we write

$$\mathrm{p}_k^t = \sum_{i=1}^{m} \left| p_{ik}^t - \widehat{p}_{ik}^t \right|,$$

and

$$\mathrm{q}_k^t = \sum_{j=1}^{n} \left| q_{jk}^t - \widehat{q}_{jk}^t \right|.$$

As estimates of the error $d(X^t, \widehat{X}^t)$, we define x_k^t as the number of entries where X^t is equal to k but $\widehat{X}^t$ is not, and define $\widehat{\mathrm{x}}_k^t$ similarly. Note that

$$d(X^t, \widehat{X}^t) \leq \sum_{k=1}^{N} k\mathrm{x}_k^t + k\widehat{\mathrm{x}}_k^t.$$

5.5.2 Coupling construction idea

We adopt a notation from (see e.g. [LPW09, LW98]) that the (total variation) *mixing time* is the smallest t such that

$$\|P^t(X) - \pi\|_{TV} \leq \frac{1}{4} \qquad \text{for all starting points } X.$$

Recall that

$$\|P^t(X) - \pi\|_{TV} \leq \mathbf{P}(\tau > t) \tag{5.5.1}$$

for every t and a coupling τ, see e.g. [LPW09, §5].

We employ a simple *coupling* construction in the proof of Theorem 5.2.5; we show that two copies of the SHM chain will on average quickly produce identical margins during the Split step, and then couple together from that point. For theorems 5.2.6 and 5.2.7 our coupling is more complicated. During the Split step we couple by producing identical cycle decompositions at every entry where X and $\widehat{X}$ match. The coupling we give for the Hyper step does not always cause $d(Z_k, \widehat{Z}_k) < d(Y_k, \widehat{Y}_k)$. We use Lemma 5.5.3 to bound the new error introduced in the Hyper step.

5.5.3 Dispersion lemma

We use the following technical lemma in the proofs of theorems 5.2.5 and 5.2.6. When we begin our algorithm by sampling from the hypergeometric distribution, we use this lemma to show that most of the nonzero values in the table are initially 1. When we compute the (total variation) mixing time, we use this lemma to show that most 1's produced during the Split step disperse into unoccupied space elsewhere in the table during the Hyper step.

Lemma 5.5.1 (Dispersion lemma)**.** *Let Y be an $m \times n$ table satisfying the conditions of Theorem 5.2.5 or of Theorem 5.2.6. Let Z be the table produced by applying a* Hyper *step to (Y). Then for any column i, and any subset S of entries in column i with $|S| = O(n^\alpha)$, the expected number of nonzero entries outside of S in column i is*

$$\sum_{j \notin S} \mathbf{P}\left(z_{ij} \neq 0\right) = q_i - o(1).$$

Proof. The probability that z_{ij} is nonzero is bounded below by

$$\mathbf{P}\left(z_{ij} \neq 0\right) \geq 1 - \left(1 - \frac{p_j}{N}\right)^{q_i} = \frac{p_j q_i}{N} - O\left(\frac{p_j^2 q_i^2}{N^2}\right).$$

The same probability is bounded above by the expectation of the entry, that is

$$\mathbf{P}(z_{ij} \neq 0) \leq \mathbb{E}[z_{ij}] = \frac{p_j q_i}{N}.$$

The marginal sums are all at least 1, so $N \geq n$ and $(p_j q_i)/N = O(n^{2\alpha-1})$. The expected number of nonzero entries outside of S in column i is thus at least

$$\sum_{j=1}^{n} \left(\frac{p_j q_i}{N} - O(n^{4\alpha-2})\right) - \sum_{j \in S} \left(\frac{p_j q_i}{N}\right) = q_i - O(n^{4\alpha-1}) - O(n^{3\alpha-1}) = q_i - o(1),$$

as desired. □

5.5.4 Coupling lemmas

The first lemma in this section gives the coupling we apply during the Hyper step. The second lemma bounds the error introduced during the Hyper step under this coupling.

Lemma 5.5.2 (Coupling lemma)**.** *Let $T, \widehat{T} \subseteq [m]$ and $U, \widehat{U} \subseteq [n]$ be subsets with $|T| = |U|$, $|\widehat{T}| = |\widehat{U}|$, $|T \cap \widehat{T}| = |U \cap \widehat{U}|$, and $|\widehat{T}| \geq |T|$. Then there is a coupling on pairs of permutations (σ, τ) with $\sigma \in S_{|T|}$, $\tau \in S_{|\widehat{T}|}$ such that the marginal distributions on $S_{|T|}$ and $S_{|\widehat{T}|}$ are uniform. Moreover, treating σ and τ as $0-1$ matrices on $T \times U$ and $\widehat{T} \times \widehat{U}$, respectively, this coupling can be constructed so that, restricting both σ and τ to $T \cap \widehat{T} \times U \cap \widehat{U}$, the ℓ_1 distance $\|\sigma - \tau\|_1 \leq |\widehat{T}| - |T|$.*

Proof. First, assume that $T \subseteq \widehat{T}$ and $U \subseteq \widehat{U}$. Choose a random permutation matrix τ on $\widehat{T} \times \widehat{U}$. We construct a permutation σ from τ. Restricting τ to $T \times U$ gives a $0-1$ matrix with at most one 1 in any row or column. If there is exactly one 1 in each row and column we call the resulting $0-1$ matrix σ. Otherwise, there are an equal number of empty rows and empty columns. Call this number k, and let the empty rows and empty columns be indexed by $x_1, \ldots, x_k$ and $y_1, \ldots, y_k$, respectively. Choose a bijection from the x_i's to the y_i's uniformly at random, place 1's at the corresponding entries in $T \times U$, and call the result σ.

The marginal distribution of τ in this coupling is uniform, by construction. The probability of generating a fixed σ is

$$\sum_{k=0}^{|\widehat{T}|-|T|} \frac{1}{k!} \binom{|T|}{k} (|\widehat{T}| - |T| - k)! \left(\frac{(|\widehat{T}| - |T|)!}{(|\widehat{T}| - |T| - k)!} \right)^2,$$

which is independent of σ. The marginal distribution of σ is thus also uniform. The ℓ_1 distance between σ and τ restricted to $T \times U$ is the number k of empty rows or empty columns in τ restricted to $T \times U$, which is at most $|\widehat{T}| - |T|$, as desired.

Next, suppose that $|T| = |\widehat{T}|$ and $|U| = |\widehat{U}|$. Then we can choose bijections $\alpha : T \to \widehat{T}$ and $\beta : U \to \widehat{U}$ such that α and β act as the identity on $T \cap \widehat{T}$ and $U \cap \widehat{U}$, respectively. Choose a random permutation $\tau : \widehat{T} \times \widehat{U}$. Then take $\sigma : T \to U$ to be $\sigma = \beta^{-1} \tau \alpha$. Since $\sigma = \tau$ on $(T \cap \widehat{T}) \times (U \cap \widehat{U})$, the ℓ_1 distance between σ and τ on that set is 0, as desired.

Note that for every collection of subsets $T, \widehat{T} \subseteq [m]$ and $U, \widehat{U} \subseteq [n]$ satisfying the conditions of the lemma, we can choose subsets $\widehat{T}' \subseteq [m]$ and $\widehat{U}' \subseteq [n]$ with $|\widehat{T}'| = |T|$, $|\widehat{U}'| = |U|$, $\widehat{T}' \subseteq \widehat{T}$, and $\widehat{U}' \subseteq \widehat{U}$. We couple permutations $\sigma : T \to U$ with permutations $\omega : \widehat{T}' \to \widehat{U}'$, and then in turn couple those permutations ω with permutations $\tau : \widehat{T} \to \widehat{U}$. This completes

the proof in the general case. □

Lemma 5.5.3. *Let Y and $\widehat{Y}$ be tables with different marginal sums $(\mathbf{p}, \mathbf{q})$ and $(\widehat{\mathbf{p}}, \widehat{\mathbf{q}})$. Let $\mathrm{p} = \sum_{i=1}^{m} |p_i - \widehat{p}_i|$ and define q similarly. Then there is a coupling*

$$(Y, \widehat{Y}) \xrightarrow{\textsc{Hyper}} (Z, \widehat{Z})$$

such that

$$d(Z, \widehat{Z}) \leq \tfrac{3}{2}(\mathrm{p} + \mathrm{q}) \leq 3d(Y, \widehat{Y}).$$

Proof. We treat the pair of tables $(Y, \widehat{Y})$ as a pair of permutation matrices in S_N and apply Lemma 5.5.2. In the language of Lemma 5.5.2, we have $|\widehat{T}| - |T| \leq \min\{\mathrm{p}, \mathrm{q}\} \leq d(Y, \widehat{Y})$. The contribution to $d(Z, \widehat{Z})$ on $(T \cap \widehat{T}) \times (U \cap \widehat{U})$ is thus bounded above by

$$\tfrac{1}{2}(\mathrm{p} + \mathrm{q}) \leq d(Y, \widehat{Y}).$$

We count the contribution to $d(Z, \widehat{Z})$ on $(T/\widehat{T}) \times [N]$ by summing $|p_i - \widehat{p}_i|$ over all indices i where $p_i > \widehat{p}_i$. We count the contribution on $(\widehat{T}/T) \times [N]$ in the same way. Together, this gives a contribution of $\sum_i |p_i - \widehat{p}_i|$. Similarly, the contribution on $[N] \times (U/\widehat{U})$ and $[N] \times (\widehat{U}/U)$ is $\sum_i |q_i - \widehat{q}_i|$. Adding these three bounds gives the desired result. □

5.5.5 Error bounding results

The following results allow us to bound the expected error $\mathbb{E}[d(X^t, \widehat{X}^t)]$ under our coupling.

Proposition 5.5.4. *Let X^t and $\widehat{X}^t$ be sequences of $m \times n$ tables under the coupling $(P, \widehat{P})$ described in §5.5.2 and Lemma 5.5.2. Suppose that there is some $\varkappa > 1$ and some $t_0 \geq 0$, s.t.*

$$\sum_{k > \varkappa} \mathbb{E}\big[N_k^t\big] \;=\; o(1) \qquad \textit{for all} \quad t > t_0 .$$

Then, for $t > t_0$ we have:

$$\sum_{k=1}^{\varkappa} k \Big(\mathbb{E}\left[\mathrm{p}_k^t\right] + \mathbb{E}\left[\mathrm{q}_k^t\right] \Big) \;\leq\; O\left(\Big| \sum_{k=1}^{\varkappa} \mathbb{E}\big[\mathrm{x}_k^t + \widehat{\mathrm{x}}_k^t\big] \Big|^{1/2} \right) + o(1),$$

where the implied constant depends only on $\varkappa$.

Proof. By Lyapunov's inequality, for every random variable S we have

$$\mathbb{E}\big[|S|\big] \le \sqrt{\mathbb{E}[S^2]} = \sqrt{\mathrm{Var}(S) + (\mathbb{E}[S])^2} \le \sqrt{\mathrm{Var}(S)} + \big|\mathbb{E}[S]\big|. \tag{5.5.2}$$

By symmetry,

$$\mathbb{E}[p_{ik}^t - \widehat{p}_{ik}^t] = 0.$$

Thus, taking

$$S = \sum_{i=1}^{m} p_{ik}^t - \widehat{p}_{ik}^t$$

in 5.5.2 gives

$$\mathbb{E}[\mathrm{p}_k^t] \le \sqrt{\mathrm{Var}\left(\sum_{i=1}^{m} p_{ik}^t - \widehat{p}_{ik}^t\right)}. \tag{5.5.3}$$

The expected number of k-cycles in a random permutation S_ℓ is $1/k$ when $\ell \ge k$, so that

$$\mathbb{E}[p_{ik}^t] = \sum_{\ell \ge k} \frac{1}{k} \mathbb{E}[\mathrm{a}_{i\ell}^t].$$

By the coupling described in §5.5.2, the only contribution to p_k^t comes from entries that contribute to x_ℓ^t or $\widehat{\mathrm{x}}_\ell^t$, with $\ell \ge k$. If $\ell < 2k$, the distribution of p_{ik}^t conditioned on X^t is binomial with probability $1/k$. For $\ell \ge 2k$, it is sufficient for us to observe that, conditioned on X^t, the contribution to p_k^t from each ℓ is independent, with mean

$$\frac{1}{k}\left(\mathrm{x}_\ell^t - \widehat{\mathrm{x}}_\ell^t\right)$$

and variance

$$O(\mathrm{x}_\ell^t + \widehat{\mathrm{x}}_\ell^t),$$

where the implied constant depends only on $\varkappa$. Thus, by the law of total variance, we have

$$\mathrm{Var}\left(\sum_{i=1}^{m} p_{ik}^t - \widehat{p}_{ik}^t\right) = O\left(\frac{1}{k}\sum_{k \ge \ell} \mathrm{x}_\ell^t + \widehat{\mathrm{x}}_\ell^t\right). \tag{5.5.4}$$

The same argument for q_{jk}^t gives that

$$\mathbb{E}[\mathrm{q}_k^t] \le \sqrt{\mathrm{Var}\left(\sum_{j=1}^{n} q_{jk}^t - \widehat{q}_{jk}^t\right)}. \tag{5.5.5}$$

and

$$\operatorname{Var}\left(\sum_{j=1}^{n} q_{jk}^t - \widehat{q}_{jk}^t\right) = O\left(\frac{1}{k}\sum_{k\geq \ell} \mathrm{x}_\ell^t + \widehat{\mathrm{x}}_\ell^t\right). \tag{5.5.6}$$

Combining (5.5.3) with (5.5.4) and (5.5.5) with (5.5.6), and absorbing the contribution from every entry larger than $\varkappa$ into the $o(1)$ term, gives the desired result. □

Lemma 5.5.5. *Let X^t and $\widehat{X}^t$ be sequences of $m \times n$ tables under the coupling $(P, \widehat{P})$ described in §5.5.2 and Lemma 5.5.2. Suppose that there is some $\varkappa > 1$ and some $t_0 \geq 0$ such that, for all $t > t_0$, we have $\sum_{k>\varkappa} \mathbb{E}[N_k^t] = o(1)$ and the probability that the pair of values*

$$\{z_{ijk}^t, \widehat{z}_{ijk}^t\} \neq \{0, 1\}$$

is $o(1)$ conditioned on $z_{ijk}^t \neq \widehat{z}_{ijk}^t$. We also require the probability that $z_{ij\ell}^t \neq \widehat{z}_{ij\ell}^t$ to be $o(1)$ conditioned on $z_{ijk}^t \neq \widehat{z}_{ijk}^t$, for $k \neq \ell$. We then have

$$\sum_{k=1}^{\varkappa} k\mathbb{E}\left[\mathrm{x}_k^{t+1} + \widehat{\mathrm{x}}_k^{t+1}\right] = O\left(\left|\sum_{k=1}^{\varkappa} \mathbb{E}\left[\mathrm{x}_k^t + \widehat{\mathrm{x}}_k^t\right]\right|^{1/2}\right),$$

where the implied constant depends only on $\varkappa$.

Proof. Suppose that every z_{ijk}^t takes only the values 0 and 1, and for every entry (i, j) at most one of z_{ijk}^t is nonzero. We would then have

$$k\mathbb{E}\left[\mathrm{x}_k^{t+1} + \widehat{\mathrm{x}}_k^{t+1}\right] = k\mathbb{E}\left[d(Z_k^t, \widehat{Z}_k^t)\right].$$

We now show that removing the conditions we imposed on z_{ijk}^t contribute at most a factor of $(1 + o(1))$. The only contributions to $k\mathrm{x}_k^{t+1} + k\widehat{\mathrm{x}}_k^{t+1}$ occur at entries (i, j) where $z_{ij\ell}^t \neq \widehat{z}_{ij\ell}^t$ for some ℓ. This contribution is bounded by $2\varkappa$ except on the $o(1)$ entries greater than $\varkappa$, so that removing our assumptions on z_{ijk}^t gives

$$\sum_{k=1}^{N} k\mathbb{E}\left[\mathrm{x}_k^{t+1} + \widehat{\mathrm{x}}_k^{t+1}\right] \leq (1 + o(1)) \sum_{k=1}^{N} k\mathbb{E}\left[d(Z_k^t, \widehat{Z}_k^t)\right].$$

Applying Lemma 5.5.3 gives

$$\sum_{k=1}^{N} k\mathbb{E}\left[\mathrm{x}_k^{t+1} + \widehat{\mathrm{x}}_k^{t+1}\right] \leq \left(\tfrac{3}{2} + o(1)\right) \sum_{k=1}^{N} k\mathbb{E}\left[\mathrm{p}_k^t + \mathrm{q}_k^t\right]$$

$$\leq \left(\tfrac{3}{2} + o(1)\right) \sum_{k=1}^{\varkappa} k\mathbb{E}[\mathrm{p}_k^t + \mathrm{q}_k^t] \; + \; o(1).$$

Applying Proposition 5.5.4 gives

$$\begin{aligned}\sum_{k=1}^{N} k\mathbb{E}\big[\mathrm{x}_k^{t+1} + \widehat{\mathrm{x}}_k^{t+1}\big] \;&\leq\; \left(\tfrac{3}{2} + o(1)\right) O\left(\Big|\sum_{k=1}^{\varkappa} \mathbb{E}\big[\mathrm{x}_k^t + \widehat{\mathrm{x}}_k^t\big]\Big|^{1/2}\right) \\ &\leq O\left(\Big|\sum_{k=1}^{\varkappa} \mathbb{E}\big[\mathrm{x}_k^t + \widehat{\mathrm{x}}_k^t\big]\Big|^{1/2}\right).\end{aligned}$$

Finally, observe that the implied constant depends only on $\varkappa$, as desired. □

5.6 Mixing time of the SHM chain for small margins

We refer to §5.5.1 for notation we adopt throughout and §5.5.2 for an overview of the coupling approach used in the proofs in this section.

5.6.1 Proof of Theorem 5.2.5

We prove this theorem by showing that, with high probability, two copies of the SHM chain running in parallel will produce identical margins during the Split step simultaneously, after which we can couple the chains together immediately. We need the following elementary result:

Proposition 5.6.1. *Let Y be an $m \times n$ table with nonzero row and column sums. Then after performing* Split*, the expected number of rows with nonzero sums in Y_1 is at least $\frac{n}{2}$. Moreover, the probability that there are less than $\frac{n}{4}$ such rows is $O(e^{-Cn})$ for some absolute constant C.*

Proof. The probability that a random permutation $\sigma \in S_k$ has at least one fixed point is $\geq 1/2$, for every $k \geq 1$. The outcomes of the given distribution are thus greater than or equal to the outcomes from a binomial distribution with $p = \frac{1}{2}$ and n trials. The exponential bound is a standard Chernoff bound. □

We now show that, when the row and column sums are bounded by an absolute constant K, our Markov chain quickly produces tables where almost every entry is 1. Suppose that column i has sum a_i, and after t steps of our Markov chain there are a_{ik}^t entries equal to k, for $1 \le k \le K$. Let q_{i1}^t be the sum of column i in Y_1^t after the SPLIT step. Since the expected number of fixed points in any permutation is 1, we have

$$\mathbb{E}[q_{i1}^t] = \sum_{k=1}^{K} \mathbb{E}[\mathrm{a}_{ik}^t].$$

The right hand side is at least $\mathrm{a}_{i1}^t + 1$ if $\mathrm{a}_{i1}^t < a_i$, and equal to exactly a_i otherwise, so that we have

$$\sum_{k=1}^{K} \mathbb{E}[\mathrm{a}_{ik}^t] \;\ge\; a_i \mathbf{P}(\mathrm{a}_{i1}^t = a_i) \;+\; \mathbb{E}[\mathrm{a}_{i1}^t + 1]\,\mathbf{P}(\mathrm{a}_{i1}^t \neq a_i).$$

For each column i, choose S to be the set of entries where $2Y_2^t + 3Y_3^t + \cdots + KY_K^t$ is nonzero and apply Lemma 5.5.1 to the subtable of Y_1^t formed by taking all rows and columns with nonzero sum. We then have

$$\mathbb{E}\big[\mathrm{a}_{i1}^{t+1}\big] \;=\; \mathbb{E}\big[q_{i1}^t\big] \;-\; O(1/n) \;\ge\; \min\big\{\mathbb{E}[\mathrm{a}_{i1}^t] + 1, a_i\big\} \;-\; O(1/n).$$

Thus, for $t > K$ and n sufficiently large, we have $\mathbb{E}[\mathrm{a}_{i1}^t] = a_i - O(1/n)$, and so the probability that $Y_2 = Y_3 = \cdots = \mathbf{0}$ is bounded below by some constant. The implied constant is the maximum value of $p_j q_i / N$, at most K^2. Thus for $n > K^2$ and $t > K$, with high probability two different tables, running two different Markov chains in parallel, will, within $O(K)$ steps, have a SPLIT step that gives $Y_2 = Y_3 = \cdots = \mathbf{0}$ for each table simultaneously. After this, we can couple together the two Markov chains immediately. By applying (5.5.1) the mixing time is $O(1)$, where the implied constant depends linearly on K.

For $n \le K^2$, the total number of possible tables is bounded above by $\binom{K^2}{K}^K = O(1)$, so that the mixing time under this condition is also $O(1)$.

5.6.2 Proof of Theorem 5.2.6

We prove this theorem by analyzing the coupling described in §5.5.2 and §5.5.4. First, we show that we can reduce our computation of the (total variation) mixing time to the case

where we begin our algorithm by sampling from the hypergeometric distribution. If we initialize our two Markov chains with random samples from the hypergeometric distribution, we have $|Y_1|, |\widehat{Y}_1| = N - O(n^{4\alpha})$ immediately. If we begin our two Markov chains with fixed tables X and $\widehat{X}$, then applying Lemma 5.5.1 and adjusting the argument from the previous section we have

$$\mathbb{E}[\mathrm{a}_{i1}^{t+1}] = \mathbb{E}[q_{i1}^t] - O(n^{\alpha-1}) \geq \min\{\mathbb{E}[\mathrm{a}_{i1}^t] + 1, a_i\} - O(n^{\alpha-1}).$$

Taking n large enough that the $O(n^{\alpha-1})$ term is less than $1/2$ we have $\mathbb{E}[\mathrm{a}_{i1}^t] = a_i - O(n^{\alpha-1})$ for $t > 2a_i$. Let $t_0 = 0$ when we begin our algorithm by sampling from the hypergeometric distribution and $t_0 = 2a_i$ when we begin with some arbitrary fixed table X. We therefore have $|Y_1|, |\widehat{Y}_1| = N - O(n^{4\alpha})$ for $t > t_0$ in both cases. We now show that the coupling time is $t_0 + O(\log n)$. Extending the argument in Lemma 5.5.1, the expected number of entries in X^t that are at least 4 is

$$n^2 O\left((\max\{p_i q_j / N\})^4\right) = O\left(n^{2+4(2\alpha-1)}\right) = O\left(n^{2\alpha-8}\right) = o(1).$$

The probability that a pair of entries $\{z_{ijk}^t, \widehat{z}_{ijk}^t\}$ is equal to $\{0, 1\}$ is

$$2\frac{p_i q_j}{N} - O\left(\frac{p_i^2 q_j^2}{N^2}\right),$$

and the probability that z_{ijk}^t or $\widehat{z}_{ijk}^t$ is at least 2 is $O\left((p_i q_j / N)^2\right)$, so that the probability that $\{z_{ijk}^t, \widehat{z}_{ijk}^t\} \neq \{0, 1\}$ conditioned on $z_{ijk}^t \neq \widehat{z}_{ijk}^t$ is

$$O\left(\frac{p_i q_j}{N}\right) = O\left(n^{2\alpha-1}\right) = o(1).$$

The conditions of Lemma 5.5.5 are therefore satisfied for $\varkappa = 3$. Applying this lemma, there is some absolute constant C_5 such that

$$d(X^t, \widehat{X}^t) \leq \sum_{k=1}^{N} k\mathbb{E}\left[\mathrm{x}_k^t\right] + k\mathbb{E}\left[\widehat{\mathrm{x}}_k^t\right] = O(1) \quad \text{for} \quad t > t_0 + C_5 \log n.$$

Once the distance between X^t and $\widehat{X}^t$ is bounded, coupling will occur if the Split steps at some bounded number of cells produce matching results, giving $Y_k^t = \widehat{Y}_k^t$ for all k. This will occur with some nonzero probability depending only on $\varkappa$, so the coupling time is $t_0 + O(\log n)$, and applying (5.5.1) completes the proof. $\square$

5.7 Mixing time of the SHM chain for smooth margins

In this section we prove Theorem 5.2.7 with the same coupling used in §5.6.2.

5.7.1 Distribution of entries

We begin with a lemma describing the distribution of values in each entry of the table. This lemma is analogous to Lemma 5.5.1 used in previous proofs.

Lemma 5.7.1. *Let $\varkappa = \lfloor 2/(1-\alpha) \rfloor$. For k satisfying $1 \le k \le \varkappa$, and for all integers $t \ge 0$, there exist constants $c(k,t)$ such that*

$$\mathbb{E}[y_{ijk}^t] = \frac{c(k,t)}{k} \left(\frac{a_i b_j}{N}\right)^k \left(1 + O(n^{\alpha-1}) + O(n^{-\alpha})\right),$$

and

$$\mathbb{E}[\gamma_{ijk}^t] = \frac{c(k,t)}{k} \left(\frac{a_i b_j}{N}\right)^k \left(1 + O(n^{\alpha-1}) + O(n^{-\alpha})\right),$$

for all pairs (i,j) of table entries. We also have

$$\mathrm{Var}[y_{ijk}^t] = \frac{c(k,t)}{k} \left(\frac{a_i b_j}{N}\right)^k \left(1 + O(n^{\alpha-1}) + O(n^{-\alpha})\right),$$

and for any pair (i,j), (ℓ,m) of table entries with $(i,j) \ne (\ell,m)$ we have

$$\mathrm{Cov}[y_{ijk}^t, y_{\ell mk}^t] = O\left(n^{2k(\alpha-1)}(n^{\alpha-1} + n^{-\alpha})\right).$$

Proof. We first verify the claimed bounds on $\mathbb{E}[y_{ijk}^t]$ for $t = 0$. Recall that $t = 0$ corresponds to sampling from the hypergeometric distribution. We set $c(k,0) = 1/k!$. From the bounds on a_i, b_j, and N, we have

$$\frac{C_3^2}{C_2 C_4} n^{\alpha-1} \le \frac{a_i b_j}{N} \le \frac{C_4^2}{C_1 C_3} n^{\alpha-1} = o(1).$$

Our initial table $A(0)$ is sampled from the hypergeometric distribution, so the probability that the (i,j) entry is exactly k is bounded below by

$$\left(1 - \frac{a_i}{N}\right)^{b_j - k} \binom{b_j}{k} \left(\frac{a_i - k}{N}\right)^k = \frac{1}{k!} \left(\frac{a_i b_j}{N}\right)^k \left(1 + O(n^{\alpha-1}) + O(n^{-\alpha})\right).$$

The same probability is bounded above by

$$(1-\frac{a_i}{N-k})^{b_j-k}\binom{b_j}{k}\left(\frac{a_i}{N}\right)^k \;=\; \frac{1}{k!}\left(\frac{a_ib_j}{N}\right)^k\left(1+O(n^{\alpha-1})+O(n^{-\alpha})\right).$$

A similar calculation shows that the probability the (i,j) entry is more than k is $O\left(n^{(k+1)(\alpha-1)}\right)$. When the (i,j) entry is equal to exactly k, the probability that y^0_{ijk} will be set equal to 1 in the SPLIT step is exactly $1/k$. To calculate the expectation and variance of y^0_{ijk}, we can absorb the probability that the (i,j) entry is greater than k into the $(1+O(n^{\alpha-1})+O(n^{-\alpha}))$ error term, giving

$$\mathbb{E}[y^0_{ijk}] \;=\; \frac{c(k,0)}{k}\left(\frac{a_ib_j}{N}\right)^k\left(1+O(n^{\alpha-1})+O(n^{-\alpha})\right),$$

and

$$\mathrm{Var}[y^0_{ijk}] \;=\; \frac{c(k,0)}{k}\left(\frac{a_ib_j}{N}\right)^k\left(1+O(n^{\alpha-1})+O(n^{-\alpha})\right),$$

as desired. A similar calculation shows that

$$\mathbf{P}(y^0_{ijk}=1\cap y^0_{\ell mk}=1) \;=\; \mathbf{P}(y^0_{ijk}=1)\,\mathbf{P}(y^0_{\ell mk}=1)\left(1+O(n^{\alpha-1})+O(n^{-\alpha})\right),$$

which gives the desired bounds on $\mathrm{Cov}[y^0_{ijk},y^0_{\ell mk}]$. We proceed by induction on t to prove the desired bounds on $\mathbb{E}[y^t_{ijk}]$ and $\mathbb{E}[\gamma^t_{ijk}]$ and the variance and covariance of y^t_{ijk}, for all $t\geq 1$. First we show that the correct bounds on $\mathbb{E}[y^t_{ijk}]$, the variance and the covariance imply the correct bounds on $\mathbb{E}[\gamma^t_{ijk}]$. Then we show the correct bounds on $\mathbb{E}[\gamma^t_{ijk}]$ imply the correct bounds on $\mathbb{E}[y^{t+1}_{ijk}]$, the variance and the covariance.

The correct bounds on $\mathbb{E}[y^t_{ijk}]$ give

$$\mathbb{E}[p^t_{ik}] \;=\; \frac{c(k,t)a_i^k}{kN^k}\sum_j b_j^k\left(1+O(n^{\alpha-1})+O(n^{-\alpha})\right),$$

$$\mathbb{E}[q^t_{jk}] \;=\; \frac{c(k,t)b_j^k}{kN^k}\sum_i a_i^k\left(1+O(n^{\alpha-1})+O(n^{-\alpha})\right),$$

and similarly

$$\begin{aligned}\mathbb{E}[N^t_k] \;&=\; \frac{c(k,t)}{kN^k}\sum_{i,j}(a_ib_j)^k\left(1+O(n^{\alpha-1})+O(n^{-\alpha})\right)\\ &=\; \frac{c(k,t)}{kN^k}\left(\sum_i a_i^k\right)\left(\sum_j b_j^k\right)\left(1+O(n^{\alpha-1})+O(n^{-\alpha})\right).\end{aligned}$$

From the bounds on the covariance, we get that

$$\mathbb{E}[p_{ik}^t q_{jk}^t] \;=\; \frac{(c(k,t))^2\, a_i^k b_j^k}{k^2 N^{2k}} \left(\sum_i a_i^k\right)\left(\sum_j b_j^k\right) \left(1 + O(n^{\alpha-1}) + O(n^{-\alpha})\right).$$

From the Taylor series expansion for X/Y, we have

$$\mathbb{E}[X/Y] \;=\; \frac{\mathbb{E}[X]}{\mathbb{E}[Y]} \;+\; O\left(\frac{\mathrm{Cov}(X,Y)}{(\mathbb{E}[Y])^2}\right) \;+\; O\left(\frac{\mathrm{Var}(Y)\mathbb{E}[X]}{(\mathbb{E}[Y])^3}\right)$$

as long as Y is bounded away from 0. Since N_k^t is approximately a binomial random variable, we have

$$\mathbf{P}\left(N_k^t < \tfrac{1}{2}\mathbb{E}[N_k^t]\right) \;=\; 1 - e^{-Cn},$$

for some constant C. We can then apply the previous calculus result to give

$$\mathbb{E}[\gamma_{ijk}^t] \;=\; \frac{c(k,t)}{k}\left(\frac{a_i b_j}{N}\right)^k \left(1 + O(n^{\alpha-1}) + O(n^{-\alpha})\right),$$

as desired. We obtain Z_k^t from Y_k^t by sampling from the hypergeometric distribution, so that

$$\mathbf{P}\left(Z_{ij}(k,t) = \ell\right) \;=\; \frac{\left(\gamma_{ijk}^t\right)^\ell}{\ell!} \left(1 + O(n^{\alpha-1}) + O(n^{-\alpha})\right).$$

Performing our MERGE step gives

$$x_{ij}^{t+1} \;=\; \sum_k k\, z_{ijk}^t\,.$$

Writing a permutation $\lambda \vdash k$ as $\lambda = 1^{a_1} 2^{a_2} \cdots$, we have

$$\mathbf{P}\left(x_{ij}^{t+1} = k\right) \;=\; \sum_{\lambda \vdash k} \prod_{\ell \geq 1} \frac{(\gamma_{ij}(\ell,t))^{a_\ell}}{a_\ell!} \left(1 + O(n^{\alpha-1}) + O(n^{-\alpha})\right)$$

and

$$\mathbf{P}\left(x_{ij}^{t+1} > k\right) \;=\; O\left(n^{(k+1)(\alpha-1)}\right).$$

Thus

$$\mathbb{E}[y_{ij}(k,t+1)] = \frac{1}{k}\mathbb{E}\left[\sum_{\lambda \vdash k} \prod_{\ell \geq 1} \frac{(\gamma_{ij}(\ell,t))^{a_\ell}}{a_\ell!}\right] \left(1 + O(n^{\alpha-1}) + O(n^{-\alpha})\right).$$

One more computation with covariances allows us to move the expectation inside the product and absorb the error into the $(1 + O(n^{\alpha-1}) + O(n^{-\alpha}))$ term. We then have

$$\mathbb{E}[y_{ijk}^{t+1}] = \frac{1}{k}\sum_{\lambda \vdash k} \prod_{\ell \geq 1} \left(\frac{a_i b_j}{N}\right)^{\ell a_\ell} \frac{(d(\ell,t))^{a_\ell}}{a_\ell! \ell^{a_\ell}} \left(1 + O(n^{\alpha-1}) + O(n^{-\alpha})\right)$$

$$= \frac{1}{k}\left(\frac{a_i b_j}{N}\right)^k \sum_{\lambda \vdash k} \prod_{\ell \geq 1} \frac{(d(\ell,t))^{a_\ell}}{a_\ell! \ell^{a_\ell}} \left(1 + O(n^{\alpha-1}) + O(n^{-\alpha})\right).$$

Setting

$$c(k, t+1) = \sum_{\lambda \vdash k} \prod_{\ell \geq 1} \frac{(d(\ell,t))^{a_\ell}}{a_\ell! \ell^{a_\ell}} \tag{5.7.1}$$

completes the induction. The bounds on the variance and covariance of $y_{ij}(k, t+1)$ follow from an argument similar to the argument for $t = 0$, and we omit the details. □

Lemma 5.7.2. *For some constant C_1 depending only on α, and for $t > C_1 \log n$, we have*

$$c(k,t) = 1 - O\big(n^{\alpha-1}\big).$$

Proof. We introduce the generating function

$$G_t(x) = \sum_{k \geq 1} c(k,t) x^k.$$

We have:

$$\int \frac{G_t(x)}{x}\, dx = \sum_{k \geq 1} \frac{c(k,t)}{k}\, x^k.$$

Thus, we can rewrite 5.7.1 as

$$G_{t+1}(x) = \exp \int \frac{G_t(x)}{x} dx - 1.$$

We have $G_0(x) = e^x - 1$, and this recurrence relation has a stable solution of

$$G(x) = \frac{x}{1-x}.$$

Therefore if $c(1,t), c(2,t), \ldots, c(k-1,t)$ are all equal to $1 + O(n^{\alpha-1})$, then (5.7.1) becomes

$$c(k, t+1) = \frac{c(k,t)}{k} + \frac{k-1}{k} + O(n^{\alpha-1}),$$

which converges to $1 + O(n^{\alpha-1})$ in $O(\log n)$ time. □

5.7.2 Proof of Theorem 5.2.7

These two lemmas taken together give us precise information about the distribution of values at each entry of the table. Following the argument in the proof of Lemma 5.7.1, the probability that $\{z_{ijk}^t, \widehat{z}_{ijk}^t\} \neq \{0,1\}$ conditioned on $z_{ijk}^t \neq \widehat{z}_{ijk}^t$ is

$$O\left(\frac{p_i q_j}{N}\right) = O\left(n^{\alpha-1}\right) = o(1),$$

so that the conditions of Lemma 5.5.5 are satisfied for $\varkappa$ as in Lemma 5.7.1 and $t_0 = 0$. Then as in the proof of Theorem 5.2.6, we apply Lemma 5.5.5 to show that there is some absolute constant C_5 such that

$$d(X^t, \widehat{X}^t) \leq \sum_{k=1}^{N} k\mathbb{E}\left[\mathrm{x}_k^t\right] + k\mathbb{E}\left[\widehat{\mathrm{x}}_k^t\right] = O(1) \quad \text{for} \quad t > t_0 + C_5 \log n.$$

As in the proof of Theorem 5.2.6, coupling will occur after this point with some probability depending only on $\varkappa$, and so the coupling time is $O(\log n)$, and applying (5.5.1) completes the proof. □

5.8 Proof of torpid mixing

5.8.1 Torpid mixing lemmas

The proof of torpid mixing is a consequence of two lemmas. First we show that the expected ℓ_1 distance between two tables connected by a single step of the Markov chain is small.

Lemma 5.8.1. *We have:*

$$\mathbb{E}[d(P^{t+1}(\omega), P^t(\omega))] \leq 18n^2 \log L\,.$$

Proof. We write $(Y_k^t)_{k\geq 1}$ for the tables produced during the Split step on $P^t(\omega)$. Then

$$\mathbb{E}\left[d(P^{t+1}(\omega), P^t(\omega))\right] \leq \sum_{k\geq 1} k\mathbb{E}\left[d(Y_k^t, Z_k^t)\right].$$

If the (i,j) entry of $P^t(\omega)$ is greater than or equal to k, then from the theory of random permutations we have

$$\mathbb{E}[y_{ijk}^t] = \frac{1}{k}.$$

For all k we have $d(Y_k^t, \mathbf{0}) = d(Z_k^t, \mathbf{0})$, and so

$$\mathbb{E}\left[d(Y_k^t, Z_k^t)\right] \leq 2\mathbb{E}[d(Y_k^t, \mathbf{0})] \leq 4n.$$

We thus have

$$\begin{aligned}
\mathbb{E}\left[d(P^{t+1}(\omega), P^t(\omega))\right] &\leq \sum_{k\geq 1} k\mathbb{E}\left[d(Y_k^t, Z_k^t)\right] \\
&\leq \sum_{k=1}^{4n} k\mathbb{E}\left[d(Y_k^t, Z_k^t)\right] + \sum_{k>4n} k\mathbb{E}\left[d(Y_k^t, Z_k^t)\right] \\
&\leq 16n^2 + \sum_{k>4n} k\mathbb{E}\left[d(Y_k^t, Z_k^t)\right].
\end{aligned}$$

For $k > 4n$, we refine our analysis of random permutations. The probability that y_{ijk}^t is at least 1 is bounded by

$$\frac{1}{k} - \frac{1}{2k^2} \leq \mathbf{P}\left(y_{ijk}^t \geq 1\right) \leq \frac{1}{k}.$$

Let r_k be the number of entries of $P^t(\omega)$ that are greater than or equal to k. Then

$$\mathbb{E}[d(Y_k^t, \mathbf{0})] = \frac{r_k}{k},$$

and

$$\begin{aligned}
\mathbf{P}(d(Y_k^t, \mathbf{0}) = 1) &\geq \left(1 - \frac{1}{k}\right)^{r_k - 1} \left(\frac{1}{k} - \frac{1}{2k^2}\right) r_k \\
&\geq \left(1 - \frac{2}{k}\right)^{r_k} \frac{r_k}{k}.
\end{aligned}$$

It is easy to see by a direct calculation that for $x > 60$ and $y/x < 1$, we have:

$$\left(1 - \frac{1}{x}\right)^y > 1 - \frac{y}{x}.$$

Taking $x = k/2$ and $y = r_k$ gives

$$\mathbf{P}(d(Y_k^t, \mathbf{0}) = 1) \geq \left(1 - \frac{2}{k}\right)^{r_k} \frac{r_k}{k} \geq \frac{r_k}{k} - \frac{2r_k^2}{k^2}.$$

The HYPER step has no effect when $d(Y_k^t, \mathbf{0}) = 1$, so $(d(Y_k^t, Z_k^t)) = 0$ when that occurs. For $k > 4n$ we thus have

$$k\mathbb{E}\left[d(Y_k^t, Z_k^t)\right] \;\le\; 2k\left(\mathbb{E}[d(Y_k^t, \mathbf{0})] - \frac{r_k}{k} + \frac{2r_k^2}{k^2}\right) \;\le\; \frac{16n^2}{k}\,.$$

Thus

$$\begin{aligned}\mathbb{E}\left[d(P^{t+1}(\omega) - P^t(\omega))\right] &\le 16n^2 + \sum_{k>4n} k\mathbb{E}\left[d(Y_k^t, Z_k^t)\right] \\ &\le 16n^2 + \sum_{k=4n+1}^{L} \frac{16n^2}{k} \;\le\; 16n^2 + 17n^2 \log L \;\le\; 18n^2 \log L,\end{aligned}$$

as desired. □

Our second lemma shows that the ℓ_1 distance between tables sampled from the uniform and hypergeometric distributions is large. We need two results.

Proposition 5.8.2. *Let W be the $2 \times n$ table whose entries are all L, and let V be a random table sampled from the hypergeometric distribution. Fix $\varepsilon > 0$. Then*

$$\mathbf{P}\left(d(V, W) > \varepsilon n L\right) \;\le\; e^{-\varepsilon n}.$$

Proof. We construct a sequence of tables $W = V_0, V_1, \ldots, V_s = V$ satisfying

$$d\big(V_i, V_{i+1}\big) \;=\; 4,$$

and

$$d(V_i, W) \;=\; 4 + d\big(V_{i+1}, W\big).$$

In other words, V_{i+1} is constructed from V_i via a Diaconis–Gangolli step (1.3.1) that increases the ℓ_1 distance from W. The probability that we sample V_{i+1} instead of V_i under the hypergeometric distribution ω is at most

$$\mathbf{P}_\omega\big(V_i\big) \;=\; \left(1 - \frac{1}{2L}\right)^2 \mathbf{P}_\omega\big(V_{i+1}\big).$$

For $d(V, W) > \varepsilon n L$, we have $s \ge \frac{\varepsilon n L}{2}$ such moves. Thus, the probability that such a V is produced by the hypergeometric distribution is bounded above by

$$\left(1 - \frac{1}{L}\right)^{\varepsilon n L} \le e^{-\varepsilon n},$$

as desired. □

Proposition 5.8.3. *Let W be the $2 \times n$ table whose entries are all L, and let U be a random table sampled from the uniform distribution. Fix $\varepsilon > 0$. Then*

$$\mathbf{P}\big(d(U,W) < (1-\varepsilon)nL\big) \;<\; \frac{1}{\varepsilon^2 n}.$$

Proof. We approximate the uniform distribution with rejection sampling. Let $(w_1, w_2, \ldots, w_{n-1})$ be a sequence of $n-1$ independent discrete random variables distributed uniformly on the interval $[0, 2n]$. Define u as

$$u \;:=\; \sum_{i=1}^{n-1} w_i \,.$$

We build a table U from $\mathbf{w}$ by setting $x_{1i} = w_{1i}$, for $1 \le i \le n-1$, $x_{1n} = nL - u$, and $x_{2i} = 2L - x_{1i}$ for $1 \le i \le n$. We reject the resulting table if x_{1n} or x_{2n} is forced to be negative. We have $\mathrm{Var}(u) = (n-1)L^2/6$, so this procedure generates a correct table with probability

$$O\left(1/\sqrt{n}\right).$$

We have $\mathbb{E}\big[|L - w_{1i}|\big] = L/2$ and $\mathrm{Var}\big(|L - w_{1i}|\big) = L^2/12$, for all $i < n$. Bayes' rule gives:

$$\mathbf{P}\big(\,|L - w_{1i}| = k \mid (n-2)L \le u \le nL\big) \;\to\; \frac{1}{2L} \quad \text{as} \quad n \to \infty$$

for every k. Thus

$$\sum_{i=1}^{n-1} \mathbb{E}\big[\,|L - w_i| \mid (n-2)L \le u \le nL\big] \;\to\; \frac{(n-1)L}{2} \quad \text{as} \quad n \to \infty.$$

Another application of Bayes' rule shows that, for $i \neq j$, $\mathrm{Cov}(|L - w_i|, |L - w_j|)$ is small. Therefore,

$$\mathrm{Var}\left(\sum_{i=1}^{n-1} |L - w_i|\right) \;=\; O(nL^2).$$

Chebyshev's inequality now implies the result. □

We can now prove our second lemma.

Lemma 5.8.4. *With high probability, $d(\omega, \pi) > (1-2\varepsilon)nL$.*

Proof. Let X_ε be the set of tables V with

$$d(V, W) \,\le\, \varepsilon n L,$$

and let Y_ε be the set of tables U with

$$d(U, W) \,\ge\, (1-\varepsilon) n L.$$

By the triangle inequality, for $V \in X_\varepsilon$ and $U \in Y_\varepsilon$ we have

$$d(U, V) \,\ge\, (1-2\varepsilon) n L.$$

These sets are disjoint for $\varepsilon < 1/2$. We have for ε fixed as n grows,

$$1 - \mathbf{P}_\omega(X_\varepsilon) - \mathbf{P}_\pi(Y_\varepsilon) \,=\, 1 - e^{-\varepsilon n} - \frac{1}{\varepsilon^2 n} \,=\, 1 - o(1),$$

as desired. □

5.8.2 Proof of Theorem 5.2.8

Comparing Lemma 5.8.4 to Lemma 5.8.1, we see that the mixing time for our MC must be at least

$$\frac{L}{18 n \log L}\,,$$

as desired. □

5.9 Mixing time of the lazy Diaconis–Gangolli chain

5.9.1 The setup

Recall the *comparison theorem* for Markov chains, see e.g. [LPW09, §13.5]. We bound the relaxation time ρ_2 for the lazy Diaconis–Gangolli chain via the relaxation time ρ_1 for the SHM chain. Note that the "lazy" condition allows us to avoid the parity consideration since the chain is reversible and all the eigenvalues are positive.

We will need the following technical lemma estimating relative diameter of the chains.

Lemma 5.9.1. *Every step of the Markov chain SHM can be constructed from at most* $(N-1)$ *steps of the Diaconis–Gangolli chain.*

The proof is postponed until the next section.

5.9.2 Proof of Theorem 5.2.9

Write each step of the SHM chain as a composition of steps of the DG chain. These can be viewed as paths in the graph Γ of the DG chain. The lengths L of all these paths is at most $(N-1) = O(n)$ by Lemma 5.9.1. The degree D of Γ satisfies

$$D \,=\, 2\binom{m}{2}\binom{n}{2} \,=\, O(n^4).$$

The number of contingency tables satisfies:

$$\mathrm{T}(\mathbf{a},\mathbf{b}) \,\le\, \binom{n+K-1}{K-1}^m \,=\, e^{O(n\log n)}.$$

Finally, the (total variation) mixing time of the SHM chain in this case is $O(1)$ by Theorem 5.2.5, implying the relaxation time $\rho_1 = O(1)$. We thus have:

$$\rho_2 \,\le\, (N-1)^2 D\,\rho_1 \,=\, O(n^6).$$

Thus, the mixing time is at most

$$\rho_2 \log \mathrm{T}(\mathbf{a},\mathbf{b}) \,=\, O(n^7 \log n),$$

as desired. □

5.9.3 Proof of Theorem 5.2.10

We follow the notation and steps of the previous proof. We have $D = O(n^4)$, $L \le (N-1) = O\big(n^{\alpha+1}\big)$, and

$$\mathrm{T}(\mathbf{a},\mathbf{b}) \,\le\, \big((2n)^{O(n^\alpha)}\big)^m \,=\, \exp O\big(n^{\alpha+1}\log n\big).$$

Since $\rho_1 = O(n^\alpha)$ by the second part of Theorem 5.2.6, we obtain

$$\rho_2 \,\le\, L^2 D\,\rho_1 \,=\, O\big(n^{3\alpha+6}\big).$$

Thus, the (total variation) mixing time is at most

$$\rho_2 \log \mathrm{T}(\mathbf{a}, \mathbf{b}) \;=\; O\big(n^{4\alpha+7} \log n\big),$$

as desired. □

5.10 Proofs of technical lemmas

5.10.1 Proof of Lemma 5.3.2

We perform the SPLIT step by generating mn random permutations, each on at most N elements, then computing their cycle decompositions and storing them. We can generate a random permutation on N elements in $O(N \log N)$ steps. Determining the cycle decompositions and storing this data takes an additional $O(N \log N)$ steps. Thus the total cost of the SPLIT step is $O(mnN \log N)$.

For each of the HYPER steps $Y_k \to Z_k$, we sample from a permutation on $|Y_k|$ elements, which requires $O(|Y_k| \log N)$ steps. We then transform the resulting permutation matrices back into contingency tables by determining which block matrix each 1 in the permutation matrix corresponds to. This requires another $O(|Y_k| + m + n)$ steps. We must perform at most N HYPER steps. Summing over k, the total cost of the HYPER steps is $O\left((N + mN + nN) \log N\right)$.

Finally, in the MERGE step, we perform N additions for each of the mn entries, so the total cost of the MERGE step is $O(mnN)$. Thus the total cost of the SHM Markov chain is $O(mnN \log N)$. □

5.10.2 Proof of Lemma 5.9.1

For the purposes of this proof, we consider the SPLIT and MERGE steps to be accounting tools that do not change the underlying table X. We use moves from the Diaconis–Gangolli chain to reproduce the HYPER steps $Y_k \to Z_k$, for every k with $Y_k \neq \mathbf{0}$. More formally, we

construct a sequence of tables $X_0 = X, X_1, X_2, \ldots$ with

$$X_i \;=\; \sum_{k=1}^{i} k\, Z_k \;+\; \sum_{k>i} k Y_k\,,$$

and use Diaconis–Gangolli moves to send $X_i \to X_{i+1}$. By the SHM construction, we can treat Y_k and Z_k as permutation matrices of size $|Y_k|$. Every valid Diaconis–Gangolli move on a permutation matrix corresponds to the action of a transposition on the corresponding permutation. It requires at most $|Y_k| - 1$ transpositions to change a permutation on $|Y_k|$ elements into some other fixed permutation on $|Y_k|$ elements. Since one Diaconis–Gangolli move (1.3.1) in Y_k corresponds to k such moves in X, mapping $X_{k-1} \to X_k$ requires at most $k|Y_k| - k$ steps, and performing all the HYPER steps requires

$$\sum_{k\geq 1} k|Y_k| - k \;\leq\; N - 1$$

steps of the Diaconis–Gangolli chain. □

5.11 Conclusions

We constructed a new Split-Hyper-Merge (SHM) Markov Chain which converges to the uniform distribution on contingency tables. We prove that it mixes in time $O(\log n)$ time for near-square matrices with small or smooth margins. This gives polynomial time approximate counting algorithms in both cases, resolving an important special case of a classical #P-complete problem. Our results are new even in the well studied case of magic squares. We view it as a major step towards completely resolving the approximate counting problem for contingency tables with *all* margins.

We also apply our results to obtain polynomial time mixing time for the well studied Diaconis–Gangolli Markov chain on contingency tables with small or smooth margins. This partially resolves a problem open for over 20 years. In particular, we give a polynomial time upper bound for constant margins. This is the first subexponential bound towards the conjecture by Diaconis and Saloff-Coste.

CHAPTER 6

Phase transition in dense contingency tables

6.1 Introduction

We present and analyze a new probabilistic model for random contingency tables with linear margins of two types. We establish a sharp phase transition for the number of such contingency tables, and for their structure.

6.2 Models

Definition 6.2.1. A *model* $\mathcal{M}(\overline{a}, \overline{b})$ for $\mathcal{T}(\overline{a}, \overline{b})$ is an $m \times n$ matrix of (not necessarily independent) random variables X_{ij} whose expectations have the same marginal sums a_i and b_j. Of course, if we understood the distribution $\mathcal{T}(\overline{a}, \overline{b})$, we could take as our X_{ij}'s the true distribution of c_{ij} under the uniform distribution on $\mathcal{T}(\overline{a}, \overline{b})$.

The Diaconis-Gangolli Markov chain acts on a model $\mathcal{M}(\overline{a}, \overline{b})$ in the same way it acts on the set $\mathcal{T}(\overline{a}, \overline{b})$. Formally, we consider every choice of two rows i and r, and two columns j and s, together with the choice of either adding or subtracting the 2×2 table:

$$\begin{matrix} +\mathbf{1} & -\mathbf{1} \\ -\mathbf{1} & +\mathbf{1} \end{matrix}$$

to the four variables

$$X_{ij} \quad X_{is}$$

$$X_{rj} \quad X_{rs}$$

if possible. Averaging over these $2\binom{n}{2}^2$ choices gives a new distribution $\{X'\}_{ij}$. We say that $\mathcal{M}(\overline{a}, \overline{b})$ is *DG-invariant* if $X_{ij} = X'_{ij}$ for all i and j.

6.3 Main example: Barvinok tables

In [Bar12], Barvinok asks whether a fixed entry of a random contingency table under the uniform distribution is a geometric random variable. Barvinok considered, as a special case, tables with margins satisfying $a_1 = \cdots = a_{n-1} = b_1 = \cdots = b_{n-1} = Cn$ and $a_n = b_n = BCn$. Because of the symmetry condition, there are only three entries to consider: center entries X_{ij}, side entries X_{in} and X_{nj} and the corner entry X_{nn}. We write X for the random variable representing the center entries, Y for the side entries, and Z for the corner entry. When we need to consider multiple center entries at the same time, we write X', X'', etc.

Part of our argument relies on considering the conditional distribution $\mathbf{P}(X_{ij} = a | Z = z)$. We refer to this distribution as ${}_zX_{ij}$, and likewise write ${}_zX$ and ${}_zY$ for the distributions of X and Y conditioned on $Z = z$.

We need the following low-correlation assumptions on our model $\mathcal{M}(\overline{a}, \overline{b})$:

[Different row and column, unconditional]: For X_{ij} and X_{rs}, with $i \neq r$ and $j \neq s$ we have

$$\frac{\mathbf{P}(X_{ij} = a \ \cap \ X_{rs} = b)}{\mathbf{P}(X_{ij} = a)\mathbf{P}(X_{rs} = b)} = (1 + O(1/n^2)) \tag{6.3.1}$$

[Different row and column]: For ${}_zX_{ij}$ and ${}_zX_{rs}$, with $i \neq r$ and $j \neq s$ we have

$$\frac{\mathbf{P}({}_zX_{ij} = a \ \cap \ {}_zX_{rs} = b)}{\mathbf{P}({}_zX_{ij} = a)\mathbf{P}({}_zX_{rs} = b)} = (1 + O(1/n^2)) \tag{6.3.2}$$

[Same row or column]: For $X = X_{ij}$ and $X' = X_{rs}$, with $i = r$ or $j = s$ and $X_{ij} \neq X_{nn}$

and $X_{rs} \neq X_{nn}$ we have

$$\frac{\mathbf{P}({}_zX_{ij} = a \ \cap \ {}_zX_{rs} = b)}{\mathbf{P}({}_zX_{ij} = a)\mathbf{P}({}_zX_{rs} = b)} = (1 + O(1/n)). \tag{6.3.3}$$

We also require that $\mathrm{Var}(X)$, $\mathrm{Var}({}_zX)$, $\mathrm{Var}(Y)$ and $\mathrm{Var}({}_zY)$ all be $O(1)$. We then prove the following:

Lemma 6.3.1. *Under the low-correlation assumptions given above, we have*

$$\frac{\mathbf{P}({}_zX = a+1)}{\mathbf{P}({}_zX = a)} = (1 - \mathbf{P}({}_zX = 0)) + O\left(\frac{1}{n}\right), \tag{6.3.4}$$

$$\frac{\mathbf{P}({}_zY = a+1)}{\mathbf{P}({}_zY = a)} = (1 - \mathbf{P}({}_zY = 0)) + O\left(\frac{1}{n}\right), \tag{6.3.5}$$

and

$$\frac{\mathbf{P}(Z = a+1)}{\mathbf{P}(Z = a)} = \frac{(1 - \mathbf{P}({}_aY = 0))^2}{1 - \mathbf{P}(X = 0)} + O\left(\frac{1}{n^2}\right). \tag{6.3.6}$$

Proof. All three results follow from the same style of calculation. We begin with (6.3.6). Any Diaconis-Gangolli move that alters the distribution Z will act on some table:

$$X \quad Y$$

$$Y' \quad Z$$

There are four cases that can occur to produce $Z = a$, for $a > 0$. We can successfully perform a Diaconis-Gangolli move subtracting on the main diagonal when $Z = a + 1$ and $X > 0$, which occurs with probability $\mathbf{P}(Z = a+1)\mathbf{P}({}_aX > 0)$. We can instead add on the main diagonal when $\mathbb{Z} = a - 1$ and both Y and Y' are not equal to zero, which occurs with probability:

$$\mathbf{P}(Z = a-1)\left(1 - (1 - \mathbf{P}({}_{a-1}Y = 0))(1 - \mathbf{P}({}_{a-1}Y' = 0 \big|_{a-1}Y = 0))\right).$$

W can also produce $Z = a$ by failing to perform a Diaconis-Gangolli move, because $Z = a$ and either $X = 0$ or one of Y and Y' is equal to 0. Applying (6.3.1) and (6.3.2) to the pairs (X, Z) and $({}_aY, {}_aY')$, respectively, allows us to re-express all conditional probabilities up to a $O(1/n^2)$ term.

$$\mathbf{P}(Z = a) = \tfrac{1}{2}\Big[\mathbf{P}(Z = a+1)\mathbf{P}(X > 0) + \mathbf{P}(Z = a)\mathbf{P}(X = 0)$$

$$+\mathbf{P}(Z=a-1)\left(1-(1-\mathbf{P}({}_{a-1}Y=0))^2\right)$$
$$+\mathbf{P}(Z=a)(1-\mathbf{P}({}_aY=0))^2\Big]$$
$$+O\left(\frac{\mathbf{P}(Z=a)+\mathbf{P}(Z=a-1)+\mathbf{P}(Z=a+1)}{n^2}\right).$$

Taking

$$f(a)=\mathbf{P}(Z=a+1)(1-\mathbf{P}(X=0))-\mathbf{P}(Z=a)\left(1-\mathbf{P}({}_aY=0)\right)^2$$

reduces this expression to the much simpler

$$f(a)=f(a-1)+O\left(\frac{\mathbf{P}(Z=a)+\mathbf{P}(Z=a-1)+\mathbf{P}(Z=a+1)}{n^2}\right).$$

The same calculation in the case $Z=0$ gives

$$f(1)=O\left(\frac{\mathbf{P}(Z=1)+\mathbf{P}(Z=0)}{n^2}\right),$$

and by induction it follows that

$$f(a)=O\left(\frac{\mathbf{P}(Z=a)}{n^2}\right).$$

The desired result follows immediately. Similarly, with probability $(1-O(1/n))$, a Diaconis-Gangolli move that changes some fixed center entry ${}_zX$ will act on a 2×2 table of the form:

$$\begin{matrix} {}_zX & {}_zX' \\ {}_zX'' & {}_zX''' \end{matrix}.$$

A similar calculation gives:

$$\frac{\mathbf{P}({}_zX=a+1)}{\mathbf{P}({}_zX=a)}=\frac{\left(1-\mathbf{P}({}_zX'=0|{}_zX=a)\right)^2}{1-\mathbf{P}({}_zX=0)}+O\left(\frac{1}{n}\right). \tag{6.3.7}$$

By equation (6.3.3), we have

$$1-\mathbf{P}({}_zX'=0|{}_zX=a)=1-\mathbf{P}({}_zX=0)-O\left(\frac{\mathbf{P}({}_zX=0)}{n}\right).$$

Substituting this into (6.3.7) gives (6.3.4). Likewise, with probability $(1-O(1/n))$, a Diaconis-Gangolli move that changes some fixed side entry Y will act on a 2×2 table

of the form:

$$\begin{matrix} {}_zX & {}_zY \\ & \\ {}_zX' & {}_zY' \end{matrix}.$$

Again, a calculation gives:

$$\frac{\mathbf{P}({}_zY = a+1)}{\mathbf{P}({}_zY = a)} = (1 - \mathbf{P}({}_zY' = 0 | {}_zY = a)) + O\left(\frac{1}{n}\right),$$

and applying (6.3.3) gives (6.3.5). □

6.4 The transition point

The expressions (6.3.4) and (6.3.5) tell us that ${}_zX$ and ${}_zY$ behave approximately like geometric random variables. We make this intuition precise by calculating the expectation:

Lemma 6.4.1. *Under the low-correlation assumptions given above,*

$$\mathbb{E}[{}_zX] = \frac{1 - \mathbf{P}({}_zX = 0)}{\mathbf{P}({}_zX = 0)} + O\left(\frac{1}{n\mathbf{P}({}_zX = 0)}\right), \tag{6.4.1}$$

and

$$\mathbb{E}[{}_zY] = \frac{1 - \mathbf{P}({}_zY = 0)}{\mathbf{P}({}_zY = 0)} + O\left(\frac{1}{n\mathbf{P}({}_zY = 0)}\right). \tag{6.4.2}$$

Proof. By direct computation. We write down the expression for $\mathbb{E}[{}_zX]$ and apply (6.3.4).

$$\begin{aligned} \mathbb{E}[{}_zX] &= \sum_{i=0}^{\infty} (i+1)\mathbf{P}({}_zX = i+1) \\ &= \sum_{i=0}^{\infty} \left[(i+1)\mathbf{P}({}_zX = i)(1 - \mathbf{P}({}_zX = 0)) + O\left(\frac{\mathbf{P}({}_zX = i)}{n}\right)\right] \\ &= \big(\mathbb{E}[{}_zX] + 1\big)\big(1 - \mathbf{P}({}_zX = 0)\big) + O\left(\frac{1}{n}\right), \end{aligned}$$

so that

$$\mathbb{E}[{}_zX] = \frac{1 - \mathbf{P}({}_zX = 0)}{\mathbf{P}({}_zX = 0)} + O\left(\frac{1}{n\mathbf{P}({}_zX = 0)}\right), \tag{6.4.3}$$

as desired. The same calculation applied to (6.3.5) gives (6.4.2). □

Theorem 6.4.2. *Under the low-correlation assumptions given above, Barvinok tables exhibit a transition point in the distribution and mean of the corner entry at* $B_C = 1 + \sqrt{1 + 1/C}$. *For* $B < B_C$, *the corner entry converges to a geometric random variable with mean* $O(1)$ *as* $n \to \infty$. *For* $B > B_C$, *the corner entry converges to a Gaussian with mean and variance* $\Theta(n)$ *as* $n \to \infty$.

Proof. The row and column constraints give us:

$$(n-1)\mathbb{E}[{}_zX] + \mathbb{E}[{}_zY] = Cn$$

and

$$(n-1)\mathbb{E}[{}_zY] = BCn - z.$$

Writing $z = DCn + \alpha n^\beta$ gives

$$\mathbb{E}[{}_zY] = (B-D)C - O\left(n^{\beta-1}\right)$$

and

$$\mathbb{E}[{}_zX] = C + O(\frac{1}{n}).$$

From (6.4.1) and (6.4.2) we obtain:

$$\mathbf{P}\big({}_zX = 0\big) = \frac{1}{1+C} + O\left(\frac{1}{n}\right),$$

and

$$\mathbf{P}\big({}_zY = 0\big) = \frac{1}{1+(B-D)C} + O\left(n^{\beta-1}\right).$$

Returning to (6.3.6), taking $a = z$ gives:

$$\frac{\mathbf{P}(Z = a+1)}{\mathbf{P}(Z = a)} = \frac{(B-D)^2C(1+C)}{(1+(B-D)C)^2} - O\left(n^{\beta-1}\right).$$

Setting the leading term equal to 1 gives a quadratic equation in $(B-D)$:

$$C(B-D)^2 - 2C(B-D) - 1 = 0,$$

with solution

$$B - D = 1 + \sqrt{1 + 1/C}.$$

We write $B_C = 1 + \sqrt{1 + 1/C}$, where we claim the transition point occurs. For $B < B_C$, and for all a, we have

$$\begin{aligned}\frac{\mathbf{P}(Z = a+1)}{\mathbf{P}(Z = a)} &= \frac{(B-D)^2 C(1+C)}{(1+(B-D)C)^2} - O\left(n^{\beta-1}\right) \\ &< \frac{B^2 C(1+C)}{(1+BC)^2} < 1.\end{aligned}$$

We conclude that Z has bounded mean and behaves like a geometric random variable.

For $B > B_C$, taking $a = (B - B_C)n + \alpha n^\beta$ gives

$$\frac{\mathbf{P}(Z = a+1)}{\mathbf{P}(Z = a)} = \frac{B_C^2 C(1+C)}{(1+B_C C)^2} - O\left(n^{\beta-1}\right) = 1 - O\left(n^{\beta-1}\right).$$

Taking the product of all such terms between a and $(B - B_C)n$ gives:

$$\frac{\mathbf{P}(Z = a)}{\mathbf{P}(Z = (B-B_C)n)} \geq \left(1 - O\left(n^{\beta-1}\right)\right)^{O(n^\beta)} = e^{-O(n^{2\beta-1})}.$$

Likewise, considering only the terms between a and $(B - B_C)n + \frac{\alpha}{2}n^\beta$ gives:

$$\frac{\mathbf{P}(Z = a)}{\mathbf{P}(Z = (B-B_C)n)} \leq e^{-O(n^{2\beta-1})}.$$

This is precisely the ratio of probabilities arising from a normal distribution with mean $(B - B_C)n$ and variance $\Theta(n)$. Thus as n grows, the discrete distribution of Z converges to the probability density function of a normal random variable, as desired. □

6.5 Discussion

We have investigated the low-correlation DG-invariant model for Barvinok tables and found that it exhibits a phase transition at the transition point $B_C = 1 + \sqrt{1 + 1/C}$. To the left of the transition point, all entries follow a geometric distribution and are $O(1)$. To the right of the transition point, the center and side entries still follow a geometric distribution. The corner entry now follows a normal distribution, with mean and variance $\Theta(n)$.

We hope that future research gives additional insights into the behavior of low-correlation models in families of tables that exhibit less symmetry than the Barvinok tables. We also hope the low-correlation assumptions can be proven mathematically to hold for the Barvinok tables, which would establish the existence of a transition point unconditionally.

CHAPTER 7

Experiments and extensions

7.1 Introduction

In this chapter, we describe two extensions of the SHM algorithm and present computational experiments. In Section 7.2, we explain formally how to use the SHM algorithm to count the number of tables with given constraints. In Section 7.3, we mention a modification of the SHM algorithm that samples from a nearly uniform distribution on higher dimensional tables with constraints on the 1-margins.

Next, we present the results of our computational experiments. In Section 7.4, we give experimental evidence for fast mixing of the SHM algorithm on five different datasets. In Section 7.5, we do an "apples-to-apples" type comparison of the SHM and Diaconis-Gangolli chains on two examples. In Section 7.6 we perform experiments with the counting algorithm described in Section 7.2. In Section 7.7, we give experiments that agree with the theoretical results in Chapter 6. These experiments are evidence in support of the low-correlation assumptions we made in Section 6.3.

7.2 Counting

7.2.1 Overview

The SHM algorithm can be adapted to compute the total number of contingency tables with given constraints. We give a detailed description of the algorithm in the next subsection. Here we present an overview of the algorithm together with a proof of correctness:

Theorem 7.2.1. *Let $\varepsilon > 0$ be chosen and let X be a table satisfying the conditions of Theorem 5.2.6 (or more generally a table for which Algorithm 1 mixes rapidly). Then there is a* FPRAS *for computing the number of contingency tables with row and column sums equal to those of X. Formally, there is an algorithm whose runtime is polynomial in $\frac{1}{\varepsilon}$ and n such that with probability greater than $\frac{1}{2}$ the output is within a factor of ε of the correct count.*

Proof. We prove this by strong induction on N. Order the rows and columns so that the row and column sums are in non-increasing order and consider the value of x_{11} in a table sampled uniformly at random. Then

$$\#(\overline{a}, \overline{b}) \cdot \mathbf{P}(x_{11} \geq k) = \#\left((a_1 - k, a_2, \ldots, a_m), (b_1 - k, b_2, \ldots, b_n)\right).$$

We can compute $\#\left((a_1 - k, a_2, \ldots, a_m), (b_1 - k, b_2, \ldots, b_n)\right)$ within a factor of $\varepsilon(N-1)/N$ by the induction hypothesis. We must then choose some k for which we can compute

$$\mathbf{P}(x_{11} \geq k)^{-1}$$

within a factor of $\varepsilon/2N$. In a collection of independent random samples from the uniform distribution on tables, the number of tables with $x_{11} \geq k$ is a binomial random variable. We use the SHM algorithm to choose some $k \geq 1$ so that, with high probability, $\mathbf{P}(x_{11} \geq k)$ is bounded away from 0 and 1.

Because the row and column sums are in non-increasing order, we have

$$\mathbf{P}\left(x_{11} \geq 1\right) \geq \frac{1}{n}.$$

We can then compute $\mathbf{P}(x_{11} \geq k)$ within a factor of $\varepsilon/2N$ with repeated sampling. □

7.2.2 Counting algorithm

We give here the details of our algorithm to approximately count the total number of tables using our Markov chain.

Counting

Input: Marginal sums $\overline{a}$, $\overline{b}$, and some constants ε and γ with $\varepsilon > 0$ and $\gamma \in (0, 1)$.

Output: An estimate of the number of tables with those marginal sums.

begin

$T \leftarrow \lceil (\frac{N}{\varepsilon})^2 \rceil$

if $\{a_i, b_j\} \in \{0, 1\}$

$n \leftarrow \sum a_i$

return $n!$

else

sort $\overline{a}$, $\overline{b}$ into non-decreasing order

for $i = 1$ **t**o T **d**o

sample uniformly from tables with marginal sums $\overline{a}$, $\overline{b}$

record entry in upper right corner

end

$c \leftarrow (100\gamma)$-th percentile in the distribution of upper-left corner entries

$c \leftarrow \min\{c, a_1, b_1, \max\{a_1 - 1, b_1 - 1\}\}$

$x \leftarrow$ fraction of tables with upper right entry $\leq c$.

return $\frac{1}{x} \cdot$ Counting$((a_1 - c, a_2, a_3, \dots), (b_1 - c, b_2, b_3, \dots), \varepsilon, \gamma)$.

end

end

The total run-time is bounded above by $NT \cdot t = N^3 t/\varepsilon^2$, where t is the effective mixing time defined in §7.4.6. By adjusting the constant ε, the count can be estimated with arbitrary desired precision. The value of γ affects both run-time and accuracy. Experimentally we found $\gamma = 0.7$ to be a good choice. The speed of the algorithm can be improved by stopping instead when we have an $n \times n$ table with row and column sums all equal to 1 and returning $n!$.

It can be further improved by stopping when row and column sums are $\leq k$ for some fixed k. A recurrence relation (whose complexity grows rapidly with k) can be used to compute the number of such tables. In our implementation, we used $k = 2$.

7.3 Multi-way tables

7.3.1 Constraints

For 2-dimensional contingency tables, we have focused entirely on the sample space $\mathcal{T}(\overline{a}, \overline{b})$, that is, on the set of tables with certain fixed row and column sums. The other natural possibilities would be to fix only the row sums, or to fix only the total sum N. Both possibilities do hold interest for statisticians [Kat14, §2.2.1], but in both cases sampling from the uniform distribution on these sets can already be done with elementary techniques (see e.g. [DE85, FLL17]). When sampling from multi-way tables, there are many more possible choices of constraints.

In a multi-way table, there are two families of associated tables. The *marginal* tables are produced by summing all possible values of some k variables to produce a $(d-k)$-dimensional table. The *partial* tables are produced by fixing the value of some k variables to produce a $(d-k)$-dimensional table (see e.g. [Kat14, §3.3.1]).

Our algorithm can be extended to multi-way tables of dimension d in the case where we constrain *only* on the 1-margins, i.e. by fixing the sum of each $d-1$ dimensional partial table. This is different from, and less general than, the SIS algorithm in [CDS06] where they can constrain on arbitrary sets of margins, dependent on the feasibility of certain algebraic computations (Markov bases and Gröbner bases). However, the algorithm in [CDS06] (see also [Sul18+]), seems best fitted for cases constraining on $(d-1)$-margins, or perhaps for $(d-k)$-margins for k small.[1]

[1] This is based on the comment in [CDS06] about the point when the algebraic computations became infeasible.

7.3.2 Algorithm

The general structure of a single Markov chain step for multi-way tables is the same as in the two-dimensional case. In fact, the algorithm is identical after we give a generalization of the HYPER subroutine to arbitrary dimensions. We give the details below. The only change in the Split and Merge subroutines is an adjustment in the indices so that the subroutines act on every entry of the table.

Recall that in the two-dimensional version of HYPER, we give a map from S_N to $\mathcal{T}(\overline{a}, \overline{b})$. In the d-dimensional case, we give a similar map from the $d-1$ term product $S_N \times \cdots \times S_N$ to $\mathcal{T}_1(\overline{a}_1, \ldots, \overline{a}_d)$. The block matrix construction in the two-dimensional case extends naturally to give a map to $\mathcal{T}_1(\overline{a}_1, \ldots, \overline{a}_d)$ from d-dimensional $0-1$ tables with size $N \times \ldots \times N$ and exactly one 1 in each $(d-1)$-dimensional partial table. It remains to give a bijection from such tables to the $(d-1)$-term product $S_N \times \cdots \times S_N$.

Let T be a d-dimensional $0-1$ table with exactly one 1 in each $(d-1)$-dimensional partial table, and fix two variables i and j. We observe that the marginal table produced by summing over the remaining $d-2$ variables gives a permutation matrix. Indeed, if we fix the value of i, there is a unique value of j for which the $d-2$ dimensional partial table contains a 1. Taking $i = 1$ and $j = 2, 3, \ldots, d$ gives $d-1$ elements of S_N. To reverse the map, send the element $(\sigma_2, \sigma_3, \ldots, \sigma_d) \in S_N \times \cdots \times S_N$ to the $0-1$ table with 1's at $(k, \sigma_2(k), \ldots, \sigma_d(k))$, for $k = 1, 2, \ldots, N$.

7.4 Examples

7.4.1 Victorian birthday/deathday table

The χ^2 value of the table is

Our first example is a 12×12 table with the month of birth and death of 82 of the descendants of Queen Victoria. This data appears first in [AH85], and is studied in [DS98].

Diaconis and Sturmfels use a Markov chain consisting of

$$\begin{matrix} + & - \\ - & + \end{matrix}$$

moves, which samples from the hypergeometric distribution on tables. Their calculations lead them to estimate that their Markov chain converges to its stationary distribution in 10^5 steps. Our Markov chain samples from the uniform distribution on the same tables in around 10 steps.

Month of birth	**Month of death**												
	Jan	**Feb**	**March**	**April**	**May**	**June**	**July**	**Aug**	**Sept**	**Oct**	**Nov**	**Dec**	**Total**
Jan	1	0	0	0	1	2	0	0	1	0	1	0	6
Feb	1	0	0	1	0	0	0	0	0	1	0	2	5
March	1	0	0	0	2	1	0	0	0	0	0	1	5
April	3	0	2	0	0	0	1	0	1	3	1	1	12
May	2	1	1	1	1	1	1	1	1	1	1	0	12
June	2	0	0	0	1	0	0	0	0	0	0	0	3
July	2	0	2	1	0	0	0	0	1	1	1	2	10
Aug	0	0	0	3	0	0	1	0	0	1	0	2	7
Sept	0	0	0	1	1	0	0	0	0	0	1	0	3
Oct	1	1	0	2	0	0	1	0	0	1	1	0	7
Nov	0	1	1	1	2	0	0	2	0	1	1	0	9
Dec	0	1	1	0	0	0	1	0	0	0	0	0	3
Total	13	4	7	10	8	4	5	3	4	9	7	8	82

Figure 7.1: Month of birth and death for descendants of Queen Victoria [DS98, Table 1].

We demonstrate the rapid mixing of our Markov chain with a series of computations. We choose a random table from the hypergeometric distribution, using a HYPER step, and then run our Markov chain for a certain number of steps and compute the χ^2 value of the resulting table. We perform $5 \cdot 10^4$ trials and compute the sample mean and sample deviation of the χ^2 values of our sampled tables. The last run, where we ran the Markov chain for 200 steps in each trial, took 204 seconds. [2]

[2]Computations were made with an Intel® Core™ i7-3610QM CPU with 2.30GHz, 4 cores and 8Gb of RAM.

Steps	Sample Mean	Sample Standard Deviation
0 (HYPER dist.)	122.49	13.69
2	160.96	20.15
5	169.02	22.03
100	169.74	21.99
200	169.94	22.12

Figure 7.2: Birthday/deathday, χ^2 after $5 \cdot 10^4$ trials.

We give a summary of results in Figure 7.2, and a plot of the sample means and sample standard deviations of the χ^2 values in Figure 7.8. We give plots comparing the sampled χ^2 values after 200 steps to the sampled χ^2 values after 2 and 5 steps in Figure 7.9.

7.4.2 Hair and eye color

Our next example is a 4×4 table with the hair and eye color of 592 people. This data comes originally from [Snee74], and is also analyzed in [DS98].

	Hair color				
Eye color	**Black**	**Brunette**	**Red**	**Blonde**	**Total**
Brown	68	119	26	7	220
Blue	20	84	17	94	215
Hazel	15	54	14	10	93
Green	5	29	14	16	64
Total	108	286	71	127	592

Figure 7.3: Hair and eye color [DS98, Table 2].

The distribution appears to be close to uniform after about 40 steps. We took $5 \cdot 10^4$ samples of the χ^2 vaues after running our Markov chain for various numbers of steps, as

in §7.4.1. We give a summary of results in Figure 7.4, and a plot of the sample means and sample standard deviations of the χ^2 values in Figure 7.10. We give plots comparing the sampled χ^2 values after 200 steps to the sampled χ^2 values after 10 and 40 steps in Figure 7.11. Further evidence for the mixing time being close to 40 steps is given in §7.6.2.

Steps	Sample Mean	Sample Standard Deviation
10	203.33	86.56
40	233.04	96.55
50	233.75	96.42
100	234.44	97.05
200	234.06	96.90

Figure 7.4: Hair and eye color, χ^2 values after $5 \cdot 10^4$ trials.

Because the SPLIT stage of our algorithm is based on the cycle decomposition of a random permutation, we would suspect that, for tables where the average entry was larger, the mixing time would be larger. Comparing this example to the previous example gives some evidence in support of that prediction. The apparent mixing time remains small enough, however, for computations to remain quite fast. It took 488 seconds (≈ 8.1 minutes) to perform $5 \cdot 10^4$ trials, consisting of 200 steps of the Markov chain in each trial.

7.4.3 Titanic survival rates

The RMS Titanic famously crashed on its maiden voyage in 1912 [Cam97]. Records from that time are well-preserved, and information about the passengers can be readily found in publicly available datasets [Kag]. We create a five dimensional multi-way table with variables survival, gender, class (first, second, or third), city of embarkation (Southampton,

Cherbourg, Queenstown, or not given) and age (0-11, 12-18, 19-48, 48+, or age not given). We use the data provided in a Kaggle model training data set with $N = 891$. We present the marginal sums in the table below. Note that in some dimensions the ratio between marginal sums is quite large.

Marginal Sums	1st	2nd	3rd	4th	5th
Survival (Y/N)	342	549			
Gender (M/F)	577	314			
Class (1/2/3)	216	184	591		
Embarkation (S/C/Q/NA)	645	168	77	2	
Age (0/12/19/48/NA)	68	71	495	80	177

Figure 7.5: Titanic marginal sums.

We perform 10^4 trials for various numbers of steps, as above, and evaluated the resulting distributions. The distribution of χ^2 after 50 steps is visually indistinguishable from the distribution after 200 steps, as shown in Figure 7.13. We estimate that the distribution is somewhat close to uniform after 50 steps, and very close to uniform after 200 steps.

We refine our analysis by looking at the sample mean and sample standard deviation, plotted in Figure 7.12. We estimate that the distributions after 100 and 200 steps are almost identical. Using the estimate of the error in sample mean derived from the sample standard deviation, after 100 steps, our sample mean is 1559.4 ± 2.7. After 200 steps, our sample mean is 1558.2 ± 2.8.

Remark 7.4.1. We derive our estimate of the error here and in §7.4.4 by using the estimate for the standard error of the mean

$$\sigma_{\overline{x}} \approx \frac{s}{\sqrt{t}},$$

				B	no		yes	
F	**E**	**D**	**C**	**A**	**no**	**yes**	**no**	**yes**
Negative	<3	<140	no		44	40	112	67
			yes		129	145	12	23
		≥140	no		35	12	80	33
			yes		109	67	7	9
	≥3	<140	no		23	32	70	66
			yes		50	80	(0) 7	13
		≥140	no		24	25	73	57
			yes		51	63	7	16
Positive	<3	<140	no		5	7	21	9
			yes		(0) 9	17	(0) 1	(0) 4
		≥140	no		(0) 4	3	11	8
			yes		14	17	5	(0) 2
	≥3	<140	no		7	(0) 3	14	14
			yes		9	16	(0) 2	(0) 3
		≥140	no		(0) 4	(0) 0	13	11
			yes		(0) 5	14	(0) 4	4

Figure 7.6: 6-way Czech autoworker data from [CDS06, Table 3].

where $\sigma_{\overline{x}}$ is the standard error, s is our sample standard deviation and t is the number of trials.

7.4.4 Czech autoworker dataset

Here we consider the 6-way $2 \times \cdots \times 2$ Czech autoworker data which appears originally in [EH85], and is further analyzed in [CDS06]. The data, given in Figure 7.6, consist of various risk factors for 1841 Czech autoworkers for coronary thrombosis. We sampled the value of χ^2 after $0, 5, 10, \ldots, 100$ steps, and performed 10^4 trials for each case. The sample mean and sample standard deviation of the χ^2 values are given in Figure 7.14. We conducted one additional run with 100 steps of the Markov chain and 10^5 trials, and computed the sample mean of χ^2 to be 1443.15 ± 1.1. We plot the distribution of χ^2 after 15 and 50 steps against the distribution after 100 steps in Figure 7.15. The distribution already appears appears to be somewhat close to uniform after 15 steps. Considering the plots of sample mean and sample standard deviation suggest that the distribution is very close to uniform after 50 steps.

Compared to the example of the Titanic, the dimension has increased by 1, the number

of entries has decreased by a factor of 4, the average entry value has increased by a factor of 10, and the range between the marginal sums has decreased dramatically. This suggests that a greater number of entries or a greater range between marginal sums is one of the stronger contributors to a greater mixing time.

7.4.5 A 16-way NLTCS table

We next consider the 16-way $2 \times 2 \times \cdots \times 2$ table with $N = 21,574$, from the National Long Term Care Survey (NLTCS), see [Ero02], which was further analyzed in [DF03]. This is near the edge of what is computationally feasible, at least on a personal computer. Running our Markov chain for 150 steps, repeating for 2000 trials, took $12,271$ seconds of computing time (≈ 3.4 hours). Still, even for this large example, the plots of χ^2 in Figure 7.16 suggest that the distribution is close to uniform in between 50 and 75 steps.

7.4.6 Summary of examples

We give a summary of the results from our examples in Figure 7.7. For a $m_1 \times m_2 \times \cdots \times m_d$ table, we let:

- Dim $:= d$
- Deg $:= m_1 m_2 \cdots m_d - (m_1 + m_2 + \ldots + m_d) + (d - 1)$
- $N :=$ # of samples
- $M :=$ our estimate for the mixing time
- $t := M \times$ (CPU per MC step) = CPU time to generate a table (nearly) uniformly

Example	Description	Dim	Deg	N	M	t (sec.)
Birthday/Deathday	§7.4.1	2	121	82	5	0.00010
Hair and Eye Color	§7.4.2	2	9	592	40	0.0019
Titanic	§7.4.3	5	228	891	100	0.020
Czech Autoworkers	§7.4.4	6	57	1841	50	0.015
NLTCS	§7.4.5	16	65519	21574	100	6.2

Figure 7.7: Summary of examples in this section.

Birthday/deathday example in §7.4.1

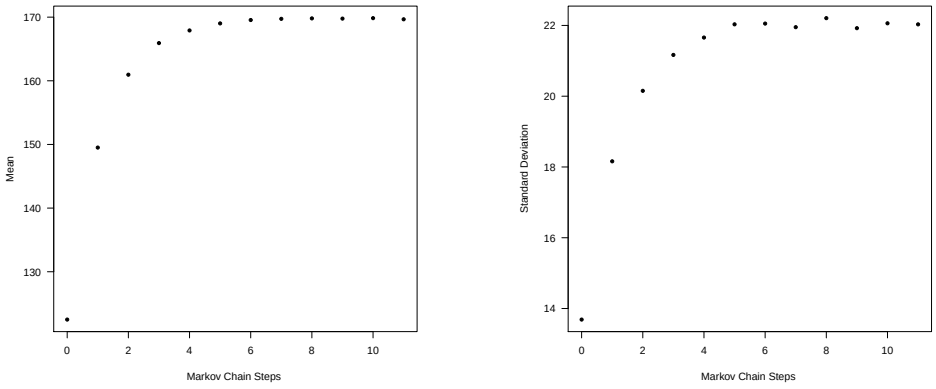


Figure 7.8: Birthday/deathday. Sample mean and sample standard deviation of χ^2.

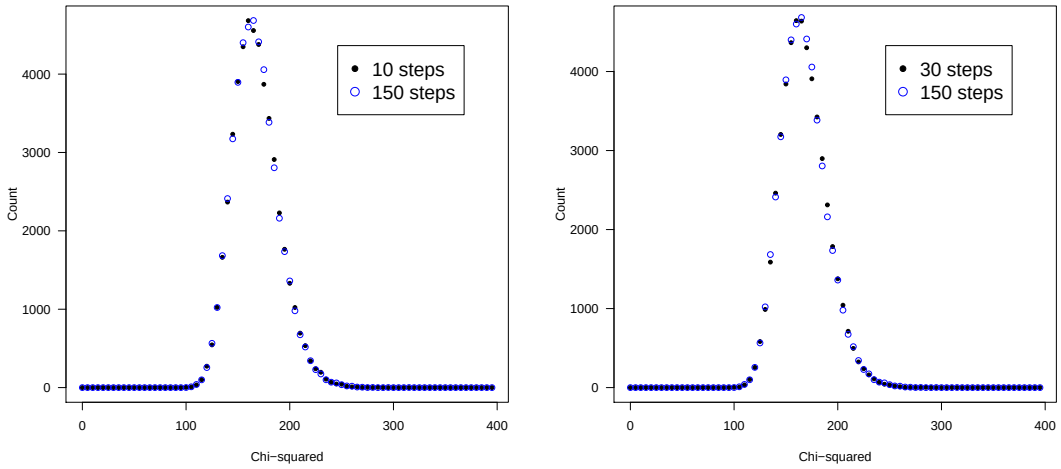


Figure 7.9: Birthday/deathday, χ^2 values after $5 \cdot 10^4$ trials with 10, 30 and 150 steps of the MC.

Hair and eye color example in §7.4.2

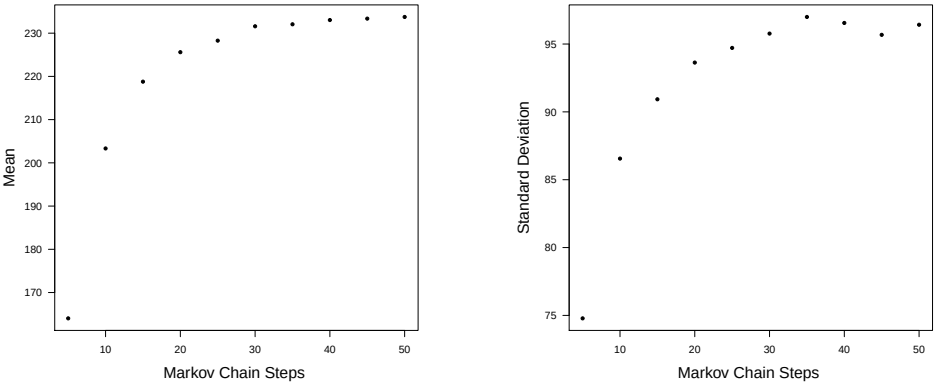


Figure 7.10: Hair and eye color. Sample mean and sample standard deviation of χ^2.

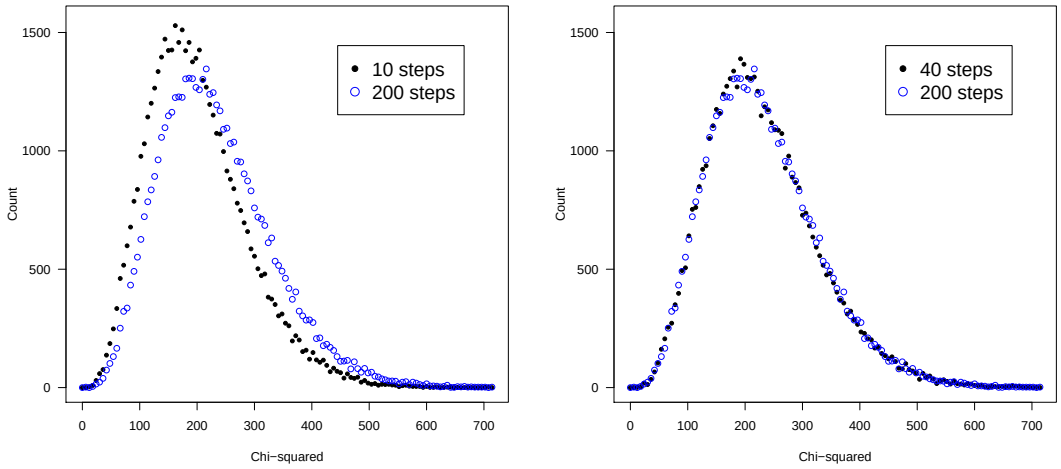


Figure 7.11: Hair and eye color, χ^2 values after $5 \cdot 10^4$ trials.

Titanic survival example in §7.4.3

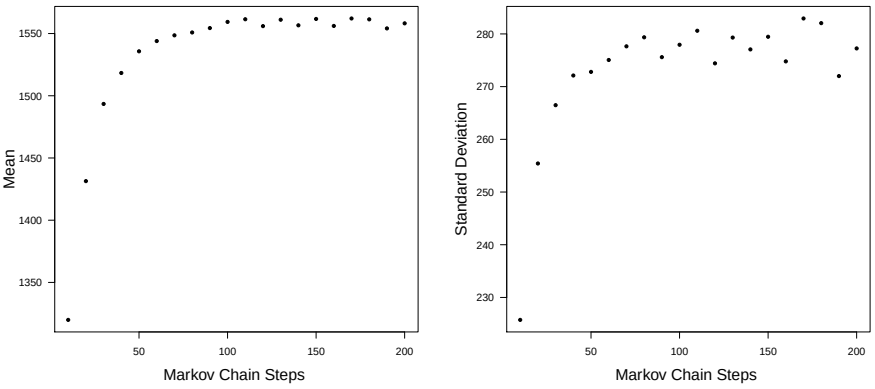


Figure 7.12: The Titanic dataset sample mean and sample standard deviations of χ^2 values after 10^4 trials with different number of steps of the MC.

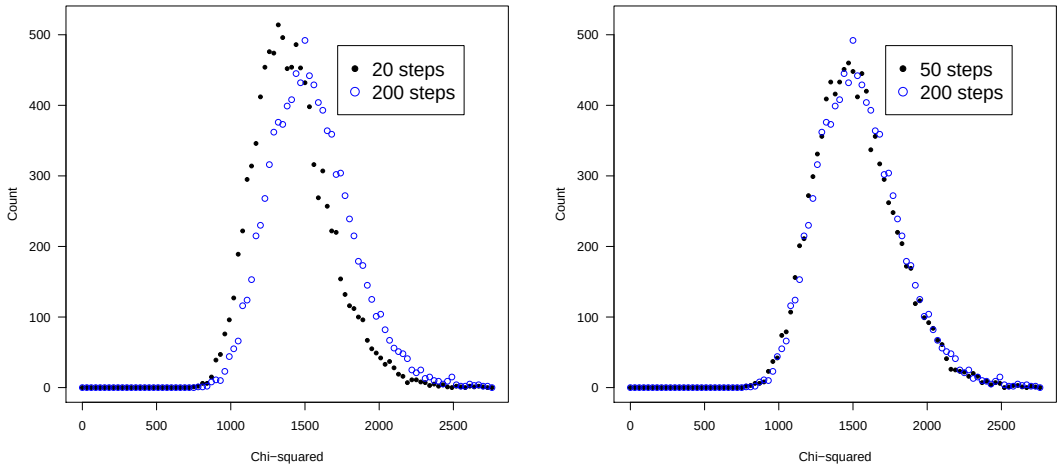


Figure 7.13: The Titanic dataset, χ^2 values after 10^4 trials with 20, 50 and 200 steps of the MC.

Czech autoworker example in §7.4.4

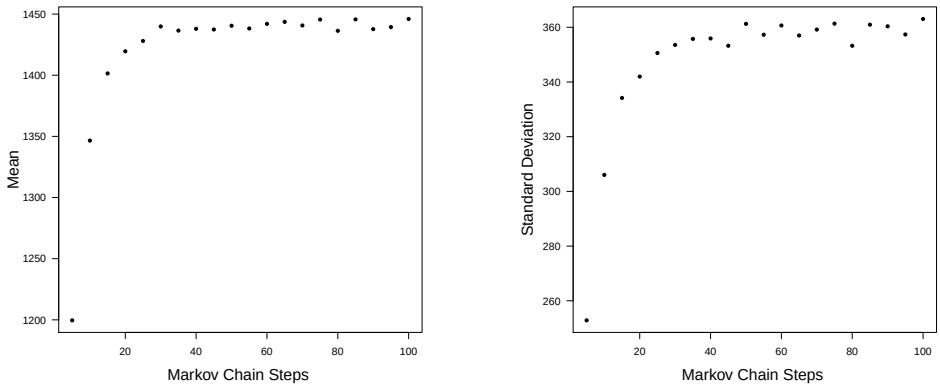


Figure 7.14: Czech auto worker sample mean and sample standard deviation of χ^2.

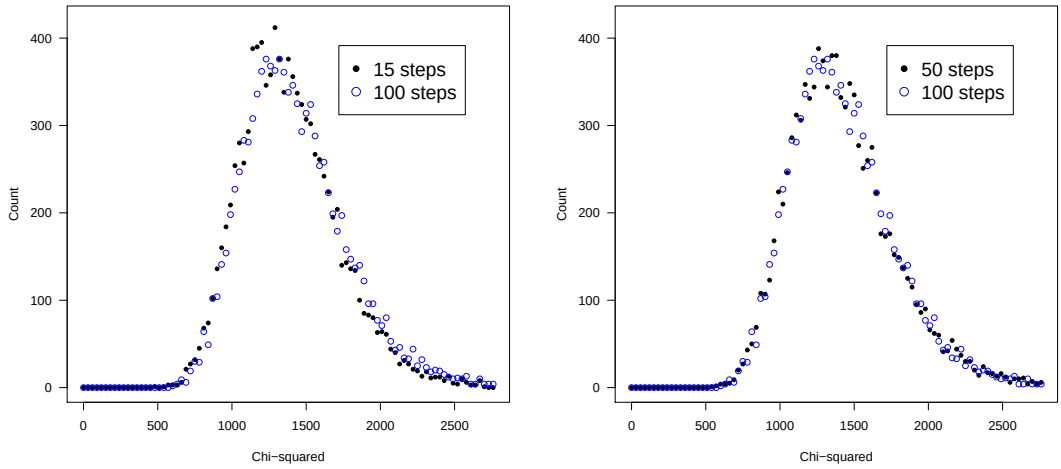


Figure 7.15: Czech autoworkers, χ^2 after 10^4 trials with 15, 50 and 100 steps.

A 16-way NLTCS example in §7.4.5

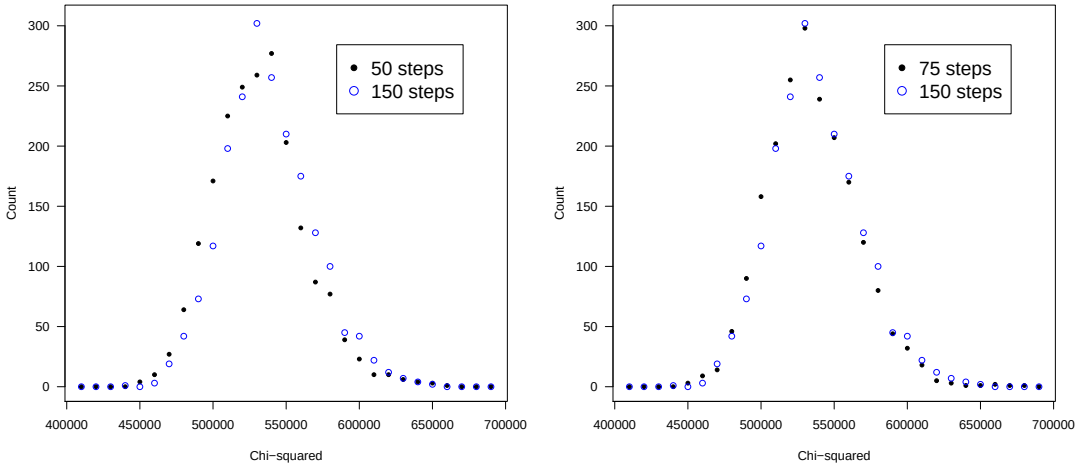


Figure 7.16: NLTCS χ^2 after 2000 trials with 50, 75 and 150 steps of the MC.

7.5 Experimental comparison of the SHM and DG chains

To compare the performance of the SHM and Diaconis-Gangolli chains, we perform a "speed test" on the birthday/deathday example of § 7.4.1 and on a 200×200 table with uniform margins of 1600 (so that the average entry is 8). A single step of the SHM chain is substantially more complex than a step of the DG chain, so comparing the mixing times of the two chains in terms of steps required is not a particularly useful metric. Instead, we compare the computer time required for each algorithm to produce a nearly uniform sample.[3]

More precisely, we measure the time required to produce, given one nearly uniform sample, a second nearly uniform sample that is nearly independent of the first. We estimate this by measuring the time required for the ℓ^2 distance between the first sample and the second sample to converge to its limit.

We first perform a calibration step to estimate the average computer time to perform one step of the SHM chain and one step of the DG chain. For the birthday/deathday example, we performed 100 trials of $500,000$ DG steps and $1,000$ SHM steps each. For the 200×200 table we performed 10 trials of $5,000$ DG steps and 10 SHM steps each.

Using the data from the calibration step, we estimate the number of MC steps that can be performed in various time intervals. For the birthday/deathday table, we use time intervals of length 0.00005, 0.0001, ... , .0.01 (i.e. $x/20,000$ for x from 1 to 200). For the 200×200 table, we use time intervals of length 0.1, 0.2, ... , 2.0 (i.e. $x/10$ for x from 1 to 20). We then perform 1,000 runs of each chain for each time interval, and compute the (entry-wise) ℓ^2 distance from the final table to the starting table. We initialize to our fixed starting table in the DG chain, and by performing a HYPER step in the SHM chain, as described in § 5.3.2. Since the performance of the DG chain depends on the initial table, while the SHM chain by design generates its own initial table, this is the best "apples to apples" comparison of the relative time required for the two algorithms to generate a family of nearly independent,

[3]Both algorithms were written in `C++` and run on the same machine, an Intel® Core™ i7-3610QM CPU with 2.30GHz, 4 cores and 8Gb of RAM. With both algorithms, we made reasonable efforts to write the code efficiently but did not exhaustively optimize by, for example, exploiting the sparseness of the underlying table.

nearly uniform samples. We plot the average ℓ^2 distance in Figure 7.17.

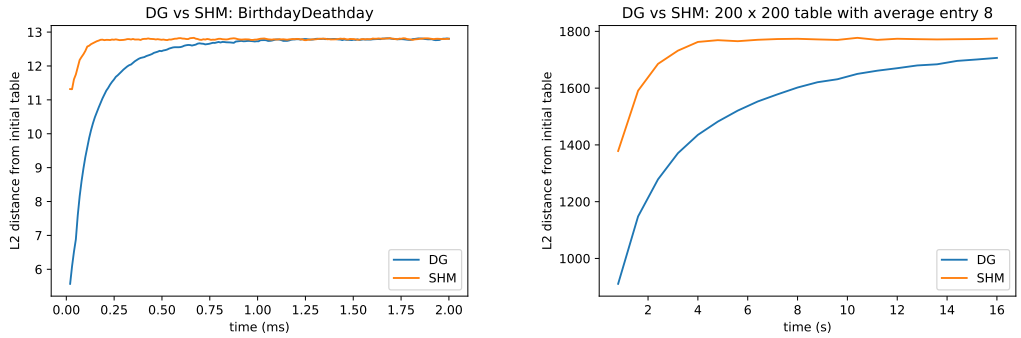


Figure 7.17: Birthday/Deathday and 200×200 speed tests

7.6 Counting experiments

In this section we give the results of tests of our counting algorithm on the birthday/deathday example from § 7.4.1 and the hair and eye color example from § 7.4.2. Recall from the discussion in § 7.2.2 that the run time is approximately proportional to t/ε^2, where t is our estimate of the mixing time.

The SHM algorithm is better suited for large, sparse tables than small, dense tables, and so the birthday/deathday example is a much more natural candidate than the hair and eye color example. However, as the results below indicate, getting an accurate estimate requires considerable computational resources in both cases. In contradistinction to the speed tests of the previous section, the use of the SHM algorithm as a FPRAS for $\mathrm{T}(\mathbf{a}, \mathbf{b})$ may be more valuable for its theoretical significance than its practical applications. If the SHM counting algorithm does offer practical advantages, those advantages are only manifest on much larger tables or require a different implementation.

7.6.1 Birthday/deathday table

Based on the experimental results for the mixing time of the SHM algorithm for the birthday/deathday table in §7.4.1, we assume that running our Markov chain for 20 steps produces

a truly uniform sample from the space of tables with those marginal sums. We perform 5 trials with $\varepsilon = 1.0$ and $\gamma = 0.7$, and 5 trials with $\varepsilon = 0.1$ and $\gamma = 0.7$.

For $\varepsilon = 1.0$, our computations lead us to estimate that the total number of tables is $6.6 \pm 0.5 \cdot 10^{39}$. Our trials gave estimates ranging from $6.10 \cdot 10^{39}$ to $7.62 \cdot 10^{39}$, with the average $6.58 \cdot 10^{39}$. The trials took on average 373.6 seconds (≈ 6.2 minutes) to run, or in total 1868 seconds (≈ 31.1 minutes).

For $\varepsilon = 0.1$, our computations lead us to estimate that the total number of tables is $6.2 \pm 0.1 \cdot 10^{39}$. Our trials gave estimates ranging from $6.16 \cdot 10^{39}$ to $6.29 \cdot 10^{39}$, with the average $6.21 \cdot 10^{39}$. The trials took on average $41,199.8$ seconds (≈ 11.4 hours) to run, or in total $205,999$ seconds (≈ 57.2 hours).

7.6.2 Hair and eye color

We return to the example of hair and eye color considered in [DS98] and above in §7.4.2. We use our counting algorithm to estimate the number of 4×4 tables with the same marginal sums. We performed 5 trials estimating the number of tables with $\varepsilon = 1.0$ and $\gamma = 0.7$, using 20 steps as an estimate of the mixing time. The trials took on average $5,764.8$ seconds (≈ 1.6 hours) to run and gave estimates of the total number of tables between $1.45 \cdot 10^{15}$ and $1.52 \cdot 10^{15}$. Des Jardins computed the actual number of tables to be exactly $1,225,914,276,768,514 \approx 1.23 \cdot 10^{15}$ [DG95]. This provides strong evidence that after 20 steps, the distribution is biased away from the uniform distribution in some way.

We performed 5 more trials with $\varepsilon = 4.0$ and $\gamma = 0.7$, using 40 steps for our mixing time. These trials took on average 834.4 seconds (≈ 13.9 minutes) to run and gave estimates of the total number of tables between $1.22 \cdot 10^{15}$ and $1.31 \cdot 10^{15}$, with the average count $1.26 \cdot 10^{15}$. This still shows a tendency to overcount, but the bias is much smaller. Our estimate of the count is $1.26 \pm 0.5 \cdot 10^{15}$, and the true value is within the margin of error.

We then performed 5 more trials with $\varepsilon = 2.0$ and $\gamma = 0.7$, using 50 steps for our mixing time. These trials took on average $3,918.8$ seconds (≈ 1.1 hours) to run and gave estimates of the total number of tables between $1.18 \cdot 10^{15}$ and $1.24 \cdot 10^{15}$, with the average

count $1.22 \cdot 10^{15}$. Our estimate of the count is $1.22 \pm 0.3 \cdot 10^{15}$, and the true value is near the center of our confidence interval.

7.7 Barvinok Experiments

Using the SHM algorithm [DP19+c], we experimentally investigated the distributions (X, Y, Z) described in Chapter 6 for various choices of n, B and C. Selected plots are given below.

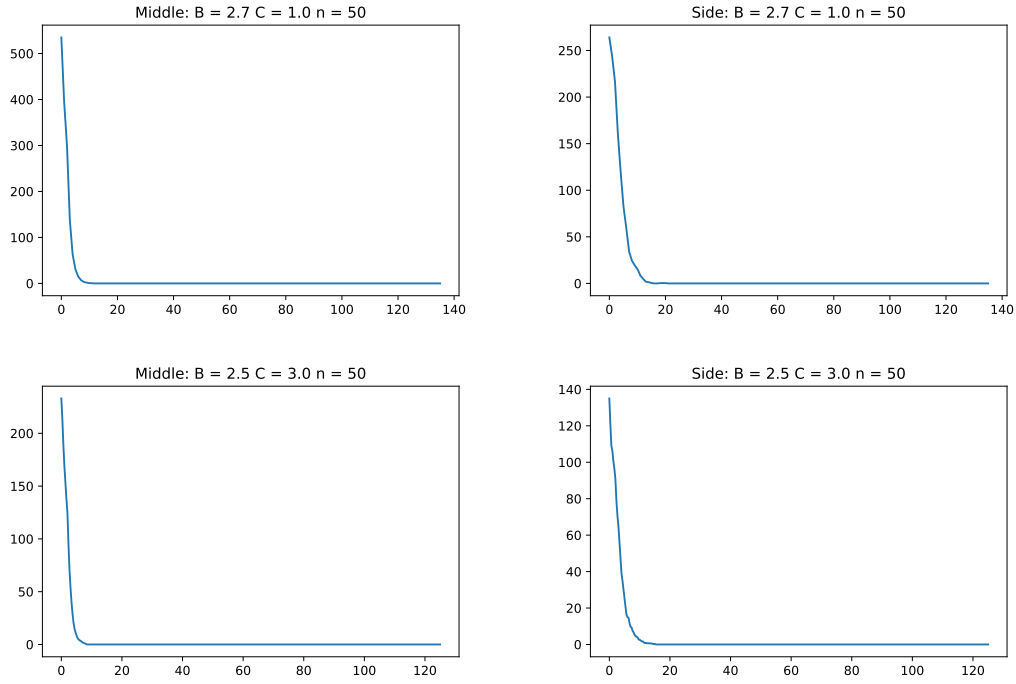
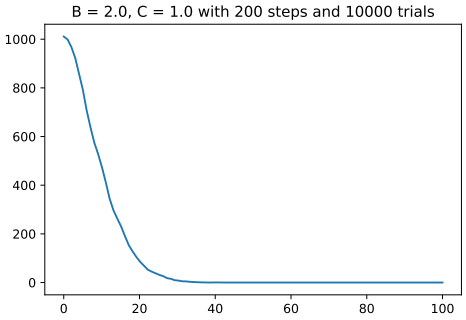


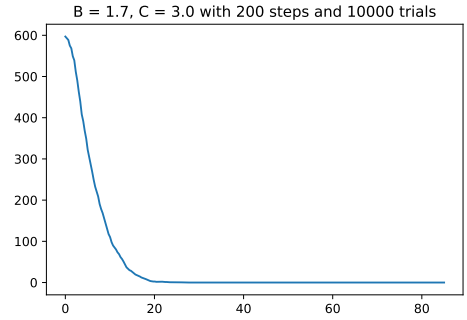
Figure 7.18: The center entry X and the side entry Y. 1,000 trials

As predicted X and Y follow a geometric distribution, regardless of our choice of B and C. In the following plots, we see the transition point for Z at $C = 1$ and at $C = 3$. For $C = 1$ we have $B_C = 2.42$, and for $C = 3$ we have $B_C = 2.15$.

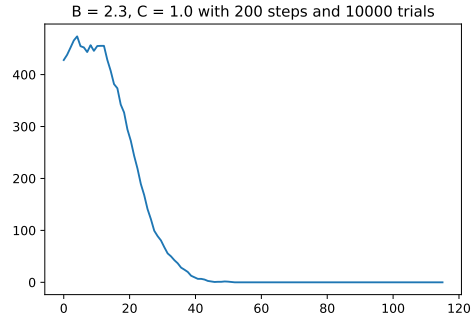
We next plot the mean and standard deviation of the corner entry Z as B varies. We show the plot for $n = 200$, $C = 1$, and for $n = 100$, $C = 3$. We again see the transition points at B around 2.42 and 2.15, repectively. To the left of the transition point, Z is



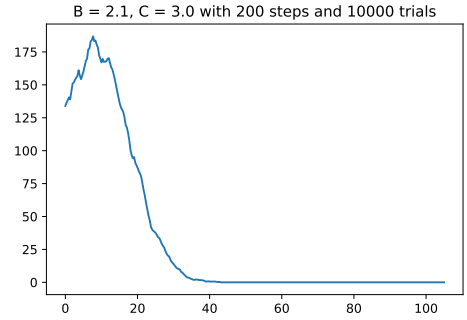
(a) For $B < B_C$, corner is geometric.



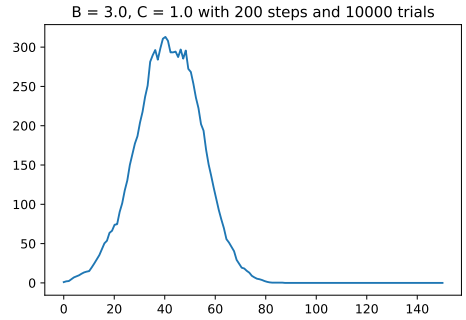
(b) For $B < B_C$, corner is geometric.



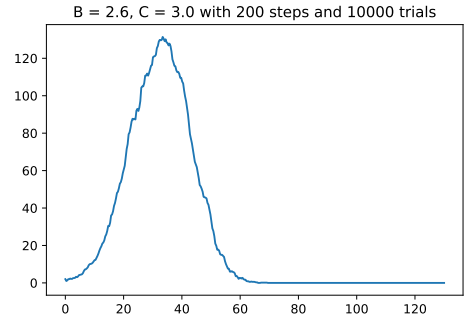
(c) For B near B_C, the behavior begins to change.



(d) For B near B_C, the behavior begins to change.



(e) For $B > B_C$, corner is Gaussian.



(f) For $B > B_C$, corner is Gaussian.

Figure 7.19: Plots of the corner entry with 10,000 trials

approximately geometric, with mean at least equal to $B > 1$, so that the mean and standard deviation are close to each other. To the right of the transition point, the standard deviation grows roughly as fast as the square root of the mean, as predicted.

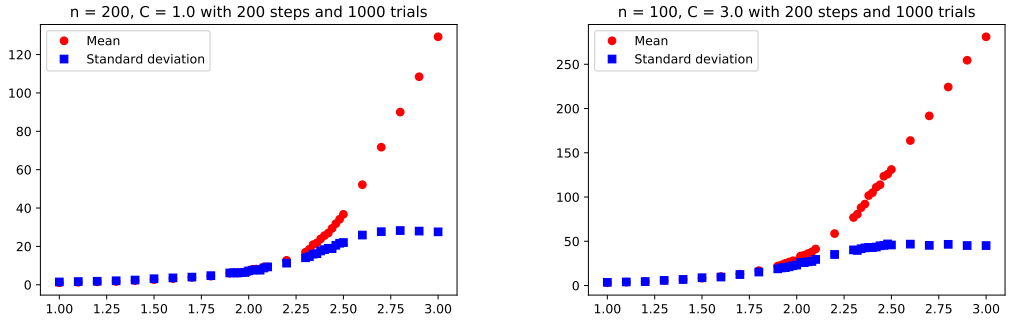


Figure 7.20: Corner entry, 1,000 trials

[AH85] D. Andrews and A. Herzberg, *Data*, Springer, New York, NY, 1985.

[Atk89] M. D. Atkinson, The complexity of orders, in *Algorithms and order* (I. Rival, Ed.), Kluwer, Dordrecht, 1989, 195–230.

[BDV04] W. Baldoni-Silva, J. A. De Loera and M. Vergne, Counting integer flows in networks, *Found. Comput. Math.* **4** (2004), 277–314.

[Ba99] A.-L. Barabási and R. Albert, Emergence of scaling in random networks, *Science* **286** (1999), 509–512.

[Bar93] A. Barvinok, A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed, in *Proc. 34th FOCS*, IEEE, Los Alamitos, CA, 1993, 566–572.

[Bar07] A. Barvinok, Brunn–Minkowski inequalities for contingency tables and integer flows, *Adv. Math.* **211** (2007), 105–122.

[Bar09] A. Barvinok, Asymptotic estimates for the number of contingency tables, integer flows, and volumes of transportation polytopes, *Internat. Math. Res. Notices* **2009** (2009), 348–385.

[Bar10a] A. Barvinok, On the number of matrices and a random matrix with prescribed row and column sums and 0-1 entries, *Adv. Math.* **224** (2010), 316–339.

[Bar10b] A. Barvinok, What does a random contingency table look like?, *Comb. Probab. Comp.* **19** (2010), 517–539.

[Bar12] A. Barvinok, Matrices with prescribed row and column sums, *Lin. Alg. Appl.* **436** (2012), 820–844.

[BLSY10] A. Barvinok, Z. Luria, A. Samorodnitsky and A. Yong, An approximation algorithm for counting contingency tables, *Rand. Struct. Algor.* **37** (2010), 25–66.

[BH12] A. Barvinok and J. A. Hartigan, An asymptotic formula for the number of non-negative integer matrices with prescribed row and column sums, *Trans. AMS* **364** (2012), 4323–4368.

[BP03] M. Beck and D. Pixton, The Ehrhart polynomial of the Birkhoff polytope, *Discrete Comput. Geom.* **30** (2003), 623–637.

[BBK72] A. Békéssy, P. Békéssy and J. Komlós, Asymptotic enumeration of regular matrices, *Studia Sci. Math. Hungar.* **7** (1972), 343–353.

[BC78] E. A. Bender and E. R. Canfield, The asymptotic number of labeled graphs with given degree sequences, J. Combin. Theory, Ser. A **24** (1978), 296–307.

[BBR11] I. Bezáková, N. Bhatnagar and D. Randall, On the Diaconis–Gangolli Markov chain for sampling contingency tables with cell-bounded entries, *J. Comb. Optim.* **22** (2011), 457–468.

[BSSV12] I. Bezáková, A. Sinclair, D. Štefankovič and E. Vigoda, Negative examples for sequential importance sampling of binary contingency tables, *Algorithmica* **64** (2012), 606–620.

[BiB01] G. Bianconi and A.-L. Barabási, Bose-Einstein condensation in complex networks, *Phys. Rev. Lett.* **86** (2001), 5632–5635.

[BjB05] A. Björner and F. Brenti, *Combinatorics of Coxeter groups*, Springer, New York, 2005.

[BjW91] A. Björner and M. Wachs, Permutation statistics and linear extensions of posets, *J. Combin. Theory* **A 58** (1991), 85–114.

[Bo01] B. Bollobás, *Random graphs* (Second ed.), Cambridge Univ. Press, Cambridge, 2001.

[BW91] G. Brightwell and P. Winkler, Counting linear extensions, *Order* **8** (1991), 225–247; extended abstract in *Proc. 23rd STOC* (1991), 175–181.

[Cam97] J. Cameron (Dir.), *Titanic* (1997), 194 min.

[CM09] E. R. Canfield and B. D. McKay, The asymptotic volume of the Birkhoff polytope, *Online J. Anal. Comb.* No. 4 (2009), 4 pp.

[CM10] E. R. Canfield and B. D. McKay, Asymptotic enumeration of integer matrices with large equal row and column sums, *Combinatorica* **30** (2010), 655–680.

[CDHL05] Y. Chen, P. Diaconis, S. Holmes and J. Liu, Sequential Monte Carlo methods for statistical analysis of tables, *Jour. Amer. Stat. Assoc.* **100** (2005), 109–120.

[CDS06] Y. Chen, I. Dinwoodie and S. Sullivant, Sequential importance sampling for multiway tables, *Ann. Stat.* **34** (2006), 523–545.

[CGY96] F. R. K. Chung, R. L. Graham and S.-T. Yau, On sampling with Markov chains, *Random Structures Algorithms* **9** (1996), 55–77.

[CV16] B. Cousins and S. Vempala, A practical volume algorithm, *Math. Program. Comput.* **8** (2016), 133–160.

[CD03] M. Cryan and M. Dyer, A polynomial-time algorithm to approximately count contingency tables when the number of rows is constant, *Jour. Comp. System Sci.* **67** (2003), 291–310.

[C+06] M. Cryan, M. Dyer, L. A. Goldberg, M. Jerrum and R. Martin, Rapidly mixing Markov chains for sampling contingency tables with a constant number of rows, *SIAM J. Comput.* **36** (2006), 247–278.

[CDR10] M. Cryan, M. Dyer and D. Randall, Approximately counting integral flows and cell-bounded contingency tables, *SIAM J. Comput.* **39** (2010), 2683–2703.

[DK14] J. De Loera and E. D. Kim, Combinatorics and geometry of transportation polytopes: an update, in *Discrete geometry and algebraic combinatorics*, AMS, Providence, RI, 2014, 37–76.

[DO04] J. De Loera and S. Onn, The complexity of three-way statistical tables, *SIAM J. Comput.* **33** (2004), 819–836.

[DZ15+] S. DeSalvo and J. Y. Zhao, Random Sampling of Contingency Tables via Probabilistic Divide-and-Conquer, 32 pp.; arXiv:1507.00070.

[DL97] M. Deza and M. Laurent, *Geometry of cuts and metrics*, Springer, Berlin, 1997.

[DE85] P. Diaconis and B. Efron, Testing for independence in a two-way table: new interpretations of the chi-square statistic, *Ann. Stat.* **13** (1985), 845–913.

[DG04] P. Diaconis and A. Gamburd, Random matrices, magic squares and matching polynomials, *El. J. Combin.* **11** (2004/06), no. 2, RP 2, 26 pp.

[DG95] P. Diaconis and A. Gangolli, Rectangular arrays with fixed margins, *Disc. Prob. Alg.* **72** (1995), 15–41.

[DS95] P. Diaconis and L. Saloff-Coste, Random walk on contingency tables with fixed row and column sums, Technical Report, Department of Mathematics, Harvard University, 1995.

[DS98] P. Diaconis and B. Sturmfels, Algebraic algorithms for sampling from conditional distributions, *Ann. Stat.* **26** (1998), 363–397.

[DP19+a] S. Dittmer and I. Pak, Counting linear extensions of dimension two posets, submitted (2019).

[DP19+b] S. Dittmer and I. Pak, Counting linear extensions of restricted posets, submitted (2019).

[DP19+c] S. Dittmer and I. Pak, Fast sampling of contingency tables, in preparation.

[DP19+d] S. Dittmer and I. Pak, Random sampling and approximate counting of sparse contingency tables, submitted (2019).

[DLP19+a] S. Dittmer, H. Lyu and I. Pak, Phase transition in random contingency tables with non-uniform margins, arXiv:1903.08743.

[DLP19+b] S. Dittmer, H. Lyu and I. Pak, Phase transition in dense contingency tables, in preparation.

[DF03] A. Dobra and S. Fienberg, Bounding entries in multi-way contingency tables given a set of marginal totals, in *Foundations of statistical inference (Shoresh, 2000)*, Physica, Heidelberg, 2003, 3–16.

[DKM97] M. Dyer, R. Kannan and J. Mount, Sampling contingency tables, *Random Structures Algorithms* **10** (1997), 487–506.

[EH85] D. Edwards and T. Havránek, A fast procedure for model search in multidimensional contingency tables, *Biometrika* **72** (1985), 339–351.

[EGKO16] E. Eiben, R. Ganian, K. Kangas and S. Ordyniak, Counting linear extensions: parametrizations by treewidth, in *Proc. 24th ESA* (2016), Art. 39, 18 pp.

[Ero02] E. Erosheva, Grade of membership and latent structure models with application to longitudinal disability survey data, Ph.D. thesis, 2002, Department of Statistics, Carnegie Mellon University, 245 pp.; available electronically at https://tinyurl.com/erosheva.

[Eve92] B. S. Everitt, *The analysis of contingency tables* (Second ed.), Chapman & Hall, London, 1992.

[FLL17] M. W. Fagerland, S. Lydersen and P. Laake, *Statistical analysis of contingency tables*, CRC Press, Boca Raton, FL, 2017.

[1] M. Faloutsos, P. Faloutsos and C. Faloutsos, On Power-Law Relationships of the Internet Topology, in *Proc. SIGCOMM*, Boston, 1999, 251–262.

[FM14] S. Felsner and T. Manneville, Linear extensions of N-free orders, *Order* **32** (2014), 147–155.

[FW97] S. Felsner and L. Wernisch, Markov chains for linear extensions, the two-dimensional case, in *Proc. 8th SODA* (1997), 239–247.

[2] B. K. Fosdick, D. B. Larremore, J. Nishimura and J. Ugander, Configuring random graph models with fixed degree sequences, *SIAM Review* **60** (2018), 315–355.

[3] A. Frieze and M. Karoński, *Introduction to random graphs*, Cambridge Univ. Press, Cambridge, 2016.

[GM77] M. Gail and N. Mantel, Counting the Number of $r \times c$ Contingency Tables with Fixed Margins, *Jour. Amer. Stat. Assoc.* **72** (1977), no. 360, part 1, 859–862.

[Gey92] C. J. Geyer, Practical Markov Chain Monte Carlo, *Stat. Sci.* **7**, No. 4 (1992), 473–483.

[GJ02] L. Goldberg and M. Jerrum, The "Burnside process" converges slowly, *Comb. Probab. Comp.* **11** (2002), 21–34.

[GM13] C. Greenhill and B. McKay, Asymptotic enumeration of sparse multigraphs with given degrees, *SIAM J. Discrete Math.* **27** (2013), 2064–2089.

[Hub14] M. Huber, Near-linear time simulation of linear extensions of a height-2 poset with bounded interaction, *Chicago J. Theoret. Comput. Sci.* (2014), Art. 3, 16 pp.

[IJ94] R. W. Irving and M. R. Jerrum, Three-dimensional statistical data security problems, *SIAM J. Comput.* **23** (1994), 170–184.

[4] S. Janson, T. Łuczak and A. Rucinski, *Random graphs*, Wiley, New York, 2000.

[Jer93] M. Jerrum, Uniform sampling modulo a group of symmetries using Markov chain simulation, in *Expanding graphs*, DIMACS Ser., AMS, Providence, RI, 1993, 37–47.

[Jer03] M. Jerrum, *Counting, sampling and integrating: algorithms and complexity*, Birkhäuser, Basel, 2003.

[Kag] Kaggle, Online Data Science Community, *Titanic: Machine Learning from Disaster*, available electronically at https://www.kaggle.com/c/titanic.

[KHNK16] K. Kangas, T. Hankala, T. Niinimäki and M. Koivisto, Counting linear extensions of sparse posets, in *Proc. 25th IJCAI* (2016), 603–609.

[KK91] A. Karzanov and L. Khachiyan, On the conductance of order Markov chains, *Order* **8** (1991), 7–15.

[Kat14] M. Kateri, *Contingency table analysis: methods and implementation using R*, Birkhäuser, New York, NY, 2014.

[Knu98] D. E. Knuth, *The art of computer programming. Vol. 3. Sorting and searching* (Second ed.), Addison–Wesley, Reading, MA, 1998.

[LS17] J. Lee and D. Skipper, Volume computations for sparse boolean quadric relaxations; arXiv: 1703.02444.

[LSS18] J. Lee, D. Skipper and E. Speakman, Algorithmic and modeling insights via volumetric comparison of polyhedral relaxations, *Math. Program.* **170** (2018), no. 1, Ser. B, 121–140.

[LPW09] D. A. Levin, Y. Peres and E. L. Wilmer, *Markov chains and mixing times*, AMS, Providence, RI, 2009.

[Lin86] N. Linial, Hard enumeration problems in geometry and combinatorics, *SIAM J. Alg. Disc. Math.* **7** (1986), 331–335.

[Lov99] L. Lovász, Hit-and-run mixes fast, *Math. Program.* **86** (1999), no. 3, Ser. A, 443–461.

[LV06] L. Lovász and S. Vempala, Hit-and-run from a corner, *SIAM J. Comput.* **35** (2006), 985–1005.

[LW98] L. Lovász and P. Winkler, Mixing times, in *Microsurveys in discrete probability*, DIMACS Ser., AMS, Providence, RI, 1998, 85–133.

[Lus03] G. Lusztig, *Hecke algebras with unequal parameters*, AMS, Providence, RI, 2003.

[Mac91] I. G. Macdonald, *Notes on Schubert polynomials*, Publ. LaCIM, UQAM, 1991.

[Man01] L. Manivel, *Symmetric functions, Schubert polynomials and degeneracy loci*, SMF/AMS, 2001.

[Mat91] P. Matthews, Generating a random linear extension of a partial order, *Ann. Probab.* **19** (1991), 1367–1392.

[MH13] J. W. Miller and M. T. Harrison, Exact sampling and counting for fixed-margin matrices, *Ann. Statist.* **41** (2013), 1569–1592.

[MS05] E. Miller and B. Sturmfels, *Combinatorial commutative algebra*, Springer, New York, 2005.

[Möh89] R. H. Möhring, Computationally tractable classes of ordered sets, in *Algorithms and order* (I. Rival, Ed.), Kluwer, Dordrecht, 1989, 105–193.

[MR16] G. Montúfar and J. Rauh, Mode poset probability polytopes, *J. Algebr. Stat.* **7** (2016), 1–13.

[MM11] C. Moore and S. Mertens, *The nature of computation*, Oxford Univ. Press, Oxford, 2011.

[MPP17] A. Morales, I. Pak and G. Panova, Hook formulas for skew shapes III. Multivariate and product formulas, arXiv:1707.00931.

[MPP18] A. Morales, I. Pak and G. Panova, Asymptotics of the number of standard Young tableaux of skew shape, *European J. Combin.* **70** (2018), 26–49.

[Mor02] B. J. Morris, Improved bounds for sampling contingency tables, *Random Structures Algorithms* **21** (2002), 135–146.

[Mou95] J. Mount, *Application of Convex Sampling Problem to Optimization and Contingency Table Gene-ration/Counting*, Ph.D. thesis, CMU, 1995, 111 pp.

[5] M. E. J. Newman, S. H. Strogatz and D. J. Watts, Random graphs with arbitrary degree distributions and their applications, *Phys. Rev. E* **64** (2001), 026118, 17 pp.

[Pad89] M. Padberg, The boolean quadric polytope: Some characteristics, facets and relatives, *Math. Program.* **45** (1989), 139–172.

[Pak00] I. Pak, Four questions on Birkhoff polytope, *Ann. Comb.* **4** (2000), 83–90.

[Pak18] I. Pak, Complexity problems in enumerative combinatorics, in *Proc. ICM Rio de Janeiro*, Vol. 3, 3139–3166, 2018; expanded version at arXiv:1803.06636, 31 pp.

[PP86] I. Pak and G. Panova, On the complexity of computing Kronecker coefficients, *Computational Complexity* **26** (2017), 1–36.

[Pap94] C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, MA, 1994.

[PA86] B. D. Patterson and W. Atmar, Nested subsets and the structure of insular mammalian faunas and archipelagos, *Bio. Jour. Linnean Soc.* **28** (1986), 65–82.

[Pit91] I. Pitowsky, Correlation polytopes: their geometry and complexity, *Math. Program.* **50** (1991), no. 3, Ser. A, 395–414.

[PB83] J. S. Provan and M. O. Ball, The complexity of counting cuts and of computing the probability that a graph is connected, *SIAM J. Comput.* **12** (1983), 777–788.

[Reu96] K. Reuter, Linear extensions of posets as abstract convex sets, *Hamburger Beiträge zur Mathematik* **56** (1996), 9 pp.; available electronically at https://tinyurl.com/ycnvbcak

[Snee74] R. D. Snee, Graphical display of two-way contingency tables, *Amer. Stat.* **38** (1974), 9–12.

[Sta86] R. P. Stanley, Two poset polytopes, *Discrete Comput. Geom.* **1** (1986), 9–23.

[Sta97] R. P. Stanley, *Enumerative Combinatorics*, Vol. 1 and 2, Cambridge Univ. Press, Cambridge, MA, 1997 and 1999.

[Tod91] S. Toda, PP is as hard as the polynomial-time hierarchy, *Siam J. Comput.* **20** (1991), 865–877.

[Tro92] W. T. Trotter, *Combinatorics and Partially Ordered Sets: Dimension Theory*, Johns Hopkins Univ. Press, Baltimore, MD, 1992.

[Tro95] W. T. Trotter, Partially ordered sets, in *Handbook of combinatorics*, Vol. 1, Elsevier, Amsterdam, 1995, 433–480.

[TW14] W. T. Trotter and R. Wang, Incidence posets and cover graphs, *Order* **31** (2014), 279–287.

[Sul18+] S. Sullivant, *Algebraic Statistics*, monograph draft, 499 pp.

[Val79] L. G. Valiant, The complexity of enumeration and reliability problems, *SIAM J. Comput.* **8** (1979), 410–421.

[Vol99] H. Vollmer, *Introduction to Circuit Complexity: A Uniform Approach*, Springer Verlag, 1999.

[Wor18] N. Wormald, Asymptotic enumeration of graphs with given degree sequence, *Proc. ICM Rio de Janeiro*, Vol. 3, 2018, 3229–3248.