A Combinatorial Tale of Two Scattering Amplitudes: See Two Bijections

by

Simeng Simone Hu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2021

Author's declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

ABSTRACT

In this thesis, we take a journey through two different but not dissimilar stories with an underlying theme of combinatorics emerging from scattering amplitudes in quantum field theories.

The first part tells the tale of the c_2-invariant, an arithmetic invariant related to the Feynman integral in ϕ^4-theory, which studies the zeros of the Kirchoff polynomial and related graph polynomials. Through reformulating the c_2-invariant as a purely combinatorial problem, we show how enumerating certain edge bipartitions through fixed-point free involutions can complete a special case of the long sought after c_2 completion conjecture.

The second part tells the tale of the positive Grassmannian and a combinatorial T-duality map on its cells, as related to scattering amplitudes in planar $\mathscr{N} = 4$ SYM theory. In particular, T-duality is a bridge between triangulations of the hypersimplex and triangulations of the amplituhedron, two objects that appear as images of the positive Grassmannian. We give an algorithm for viewing T-duality as a map on Le diagrams and characterize a nice structure to the Le diagrams (which can then be used in lieu of the algorithm). Through this Le diagram perspective on T-duality, we show how the dimensional relationship between the positroid cells on either side of the map can be directly explained.

TABLE OF CONTENTS

1 | INTRODUCTION

1.1. COMBINATORICS IN DISGUISE IN THE WORLD OF PHYSICS: THE STORY OF TWO TYPES OF SCATTERING AMPLITUDES

One of the fundamental problems in high energy physics is to understand how particles interact, how they scatter when they collide. For example, as illustrated in Figure 1.1, we can think of experiments done in particle accelerators like the Large Hadron Collider (LHC) in which some number of particles are brought together, and the particles resulting from that interaction are detected. **Quantum field theory** (QFT) is one such framework to understand these processes, and one which unifies quantum mechanics and special relativity. A natural question that arises is: what is the probability of a given configuration of particle scatterings? In quantum field theory, this probability is encoded in the **scattering amplitude** [IZ80].

However, we don't know what actually happens when the particles interact and what paths the particles are taking, the "history" of the collision. Perturbative quantum field theory thus expresses each of these scatterings as a formal power series, a perturbative expansion, in the **coupling constant** which is measuring the strength of the interactions. In this expansion, each term corresponds to a particular "history" that the particles trace out when interacting. In other words, in the spirit of quantum physics, we are taking a weighted sum of all possible interactions that get us to some particular scattering configuration. The weights in this sum are called **Feynman integrals** and the scattering amplitude of some configuration is then the sum of all the Feynman integrals in the expansion. However, in general these integrals are notoriously hard to calculate and the number of terms in each sum rapidly expands as the number of particles grows.

The fascinating part about these particular problems is the emergence of rather beautiful connections to combinatorics, and the study of discrete structures. Here begins our tale of two different connections to combinatorics as related to scattering amplitudes in quantum field theory.
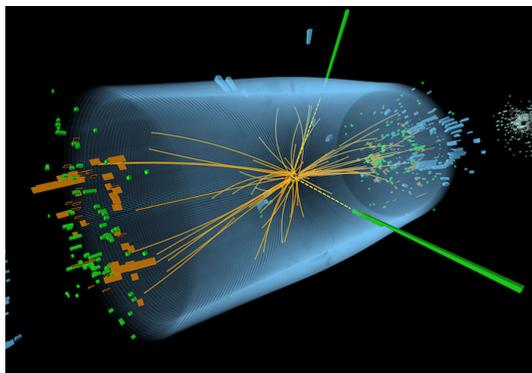

FIGURE 1.1: A candidate Higgs boson event from collisions between protons in the LHC where the particle is decaying into two protons indicated by the yellow dashed-lines with green towers (Image: CMS/CERN)

(**1.1.1**) **Feynman diagrams.**

One way to view these particle scatterings is through the underlying graphs of the interactions, called Feynman diagrams or **Feynman graphs**, formed by the paths that the particles take. The edges of the graphs correspond to the propagations of particles, while the vertices correspond to the collisions or interactions. Incoming and outgoing particles are then represented by external half-edges (legs). As such, the complexity of a problem involving Feynman graphs can be measured in two ways, via the number of cycles, which physicists call **loops**, or via the structure of the external legs. For a particular field theory, we can further encode the different fundamental particles through having different edge types, and the permissible interactions between them can be encoded through allowable vertex types (as defined by the number of each edge type adjacent to a vertex).
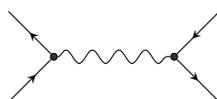
FIGURE 1.2: A Feynman diagram in quantum electrodynamics (QED), a quantum field theory of the electromagnetic force. The squiggly edge represents the propagation of a proton and the directed half-edges represent incoming and outgoing fermions.

Each graph can then be associated to a Feynman integral though the **Feynman rules**. That is the rules dictate how to build the integral based on the structure of the graph. In the standard approach to quantum field theory, the Feynman rules can be seen as arising from the Lagrangian in the path integral formulation for the scattering amplitude. However, by taking a graphs-first approach to particle interactions, this opens the door to using tools from the well established and classical field of graph theory to tackle problems in perturbative quantum field theory [Yea17].

In practice, as many of the Feynman integrals are often divergent, physicists introduced a technique called **renormalization** in order to extract meaningful quantities from these integrals. As an example, one such scheme for renormalizing Feynman integrals is called **BPHZ renormalization**, which we can think of as recursively subtracting off divergent parts of the integral corresponding to divergent subgraphs, "subdivergences". In particular, renormalization itself has an interesting connection to combinatorics via the combinatorial Hopf algebras that appear as the underlying algebraic structure to some of these schemes. For BPHZ renormalization, this is the Connes-Kreimer Hopf algebra of rooted trees. Renormalization, Hopf algebras, and recursive structures on Feynman diagrams then also open the door to Dyson-Schwinger equations, Green's functions, and their related combinatorics such as chord diagrams [Yea17]. With renormalization, we can much more easily study graphs with high loop orders and a low number of external legs.

Going back to Feynman integrals, other than their relation to the scattering amplitude, one of the other reasons to study them lies in the many interesting numbers that they can evaluate to. As an example, there are Euler sums, some of which evaluate to zeta values, and polylogarithms. However as we mentioned before, these integrals are quite complicated due to the need for renormalization and very hard to compute. Thus in order to study the numbers that appear as Feynman integrals, we instead restrict to a particular residue of the integral called the **Feynman period** [Bro10] which is renormalization scheme independent and captures much of the number-theoretic content for massless scalar field theories in four-dimensional space-time. We can think of this period as taking only the
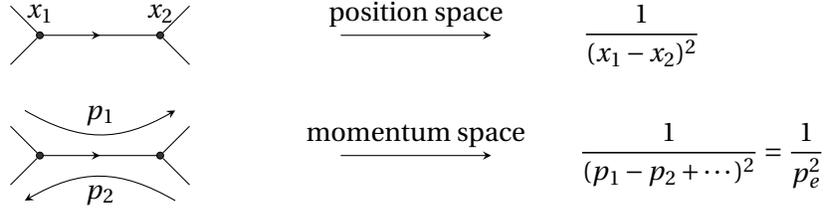
F I G U R E 1 . 3 : Feynman rules for massless scalar field theories. For momentum space, p_e refers to the signed sum of the cycles running through edge e.

"primitive divergent" part of the leading contribution to the integral, that is the part corresponding to subdivergence free Feynman graphs.

We will focus on periods in massless Euclidean ϕ^4-**theory**, which is a scalar field theory with one particle type and whose interactions are all quartic, that is, there is only one type of edge and all vertices are 4-valent. In graph theory language, the Feynman graphs are just 4-regular graphs with some number of external half-edges. Historically, ϕ^4-theory has been an important toy model for more complicated field theories. As an motivating example, some of the numerical structure seen in the anomalous magnetic moment of the electron in perturbative QED is also manifest in the beta-function of ϕ^4-theory via its coefficients, which are periods.

Now, to build the period of a given primitive divergent graph G, we can use the Feynman rules which give equivalent representations of the period in the different spaces: position, momentum, parametric, and dual parametric. The first two of these corresponds to assigning 4-dimensional vectors to vertices and cycles (in a cycle basis) respectively, with each edge of the graph contributing a factor to the integrand following the rules in Figure 1.3. Note that we are arbitrarily orienting the edges and cycles of the graph and choosing a cycle basis in momentum space. For a graph with loop order ℓ, i.e. the number of cycles or first Betti number, the period of G in momentum space is

$$P_G = \pi^{-2(\ell-1)} \int \frac{\mathrm{d}p_2 \cdots \mathrm{d}p_\ell}{\prod_e p_e^2}\bigg|_{p_1=\mathbb{1}},$$

where we are integrating over $\mathbb{R}^{4(\ell-1)}$ and we take p_e^2 to mean taking the norm squared. Note that we can set one momentum vector, say p_1, to a fixed choice of unit $\mathbb{1}$ as we can always normalize the momentum variables. In momentum space, we can then think of the period as the Feynman integral where we are setting all external properties, masses and momenta, to zero (and setting a variable to 1).

We could similarly represent the period in position space using the rules or through a Fourier transform of the momentum space period. Using the Schwinger trick, we can transform the period to parametric space, which will be our main focus,

$$P_G = \int_0^\infty \frac{\mathrm{d}\alpha_1 \cdots \mathrm{d}\alpha_{|E(G)|-1}}{\Psi_G^2\Big|_{\alpha_{|E(G)|}=1}}.$$

Here Ψ_G is the **Kirchhoff polynomial**. Continuing with the graphs-first approach to study the period, here starts one story of combinatorics emerging in scattering amplitudes. We will continue with this story in Part I.

3

(**1.1.2**) **On-shell diagrams.**

One of the limitations of the classical formulation of perturbative quantum field theory is the explosion in the number of Feynman diagrams that can appear for any particular scattering configuration, even in the simple case of two particles interacting. This is a consequence of taking an all possible "histories" expansion, which introduces a large amount of redundancy adding to the complexity of computing scattering amplitudes. These redundancies can be seen as coming from the fact that we don't actually know the paths that the particles are taking and the internal edges in the Feynman diagrams are actually representing **virtual**, or **off-shell particles**. They are called virtual particles as they do not satisfy the energy-momentum relation and thus are non-physical particles whose momenta cannot exist.

As an answer to this problem, a small group of researchers [AH+16] (as part of an ongoing program) developed a new way of thinking about scattering amplitudes in quantum field theory, one using only physical, or **on-shell particles**. This reformulation illuminated the many beautiful mathematical structures underlying on-shell processes and the remarkable simplicity of their amplitudes.

At the heart of it all, this theory was built upon the extensive exploration of scattering amplitudes in planar $\mathcal{N} = 4$ **supersymmetric Yang-Mills (SYM) theory**. To try to simply explain what this theory is, we take a brief detour using the language of differential geometry. Instead of working directly on space-time like scalar field theories, *gauge theories* are models that are defined on fibre bundles over space-time in which the Lagrangian is invariant under certain symmetries, *gauge transformations*. To get back to space-time, one would then need to pick a local section, a *gauge*. In particular, *Yang-Mills theory* is one such important gauge theory when trying to understand the Standard model. To study this rather complicated theory, researchers turn to $\mathcal{N} = 4$ SYM, which is a maximally supersymmetric theory often considered as a toy model for four-dimensional Yang-Mills theory, and thus for quantum chromodynamics (QCD), which is a quantum field theory (and a particular Yang-Mills theory) describing the strong interactions between quarks and gluons. The $\mathcal{N} = 4$ here refers to the number of supersymmetries. While $\mathcal{N} = 4$ SYM incorporates many complicated particles into its theory, its highly symmetric nature lends itself to having many special properties and dualities, and is what makes this theory attractive.

Unlike the Feynman integral story where the main study is on diagrams with high loop orders but low numbers of external legs, much of the work in these gauge theory amplitudes started with the opposite complexity; looking at low loop orders but high numbers of external legs. In particular, at tree-level (no loops) one of the breakthroughs has been in the discovery of the **BCFW recurrence relations** [Bri+05] for amplitudes in Yang-Mills theory, which could decompose the amplitude in various ways depending on the choices at each recursive step. Furthermore, each step only ever involved on-shell particles.

Building upon these relations and related work at the time, a surprising connection between scattering amplitudes in $\mathcal{N} = 4$ SYM theory and the **positive Grassmannian**, which is a certain "non-negative" subset of the Grassmannian, started to emerge. In the flurry of work which culminated in [AH+16], **on-shell diagrams** were introduced to describe on-shell processes like those that decompose the scattering amplitude and which arise from glueing together three-particle amplitudes.

In graph theory language, these diagrams are bi-coloured trivalent graphs which are planar when in the context of planar $\mathcal{N} = 4$ SYM. In this story about scattering amplitudes, the on-shell diagram plays the role of the Feynman diagram. These on-shell diagrams, which satisfied certain equivalences,
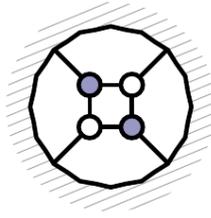
FIGURE 1.4: An on-shell diagram which represents a four-particle tree-level amplitude. Note that "tree" here refers to the amplitude in the context of the Feynman diagram expansion. (Image: [AH+16])

turned out to be **plabic graphs**, which Postnikov [Pos06] introduced as one of the many combinatorial objects that index cells of the positive Grassmannian!

Inspired by these connections and the question of where the on-shell diagram decompositions of the amplitude comes from, Arkani-Hamed and Trnka [AHT14] introduced a new mathematical object called the **amplituhedron**. They proposed that the $m = 4$ amplituhedron was a geometric representation of scattering amplitudes in planar $\mathcal{N} = 4$ SYM, which would be encoded through its "volume", and one in which the BCFW recursion relations could be manifested through its "triangulations". Through the positive Grassmannian, the amplituhedron and triangulations, herein lies our second combinatorial story, this time in a different type of scattering amplitude. We will pursue this story in Part II.

1.2. OVERVIEW AND ORGANIZATION

This thesis is composed of two self-contained parts, consisting of two stories with the underlying theme of combinatorics related to scattering amplitudes in quantum field theories.

In the first part, Part I, we delve deeper into the combinatorics of an arithmetic invariant, called the c_2-**invariant**, related to a particular residue of the Feynman integral called the Feynman period. This invariant studies the zeros of the denominators of these periods that arise after several integrations.

We start off in Chapter 2 with an overview of the main object of interest in this story, the c_2-invariant. Section 2.1 provides some background on ϕ^4-theory and Feynman periods, setting up the motivation for defining the c_2-invariant which we present in Section 2.2. In Section 2.3 we develop the framework of graph polynomials, specifically Dodgsons and spanning forest polynomials, which then allows us to start interpreting the c_2 combinatorially via spanning trees and spanning forests. Going back to the period and using this framework, in Section 2.4 we present the main algorithms used to compute the c_2 which are based on the denominators appearing in periods after successive integrations. In particular in Section 2.4.2 we see how we can reformulate the c_2 into a purely combinatorial problem, one about taking particular coefficients of graph polynomials! We end this introductory chapter with Section 2.5, giving some of the symmetries of and computational results on the c_2-invariant.

In Chapter 3 we answer a special case of an over 10-year old conjecture on the c_2-invariant using combinatorial methods involving enumerating certain edge bipartitions, completing the argument as first started by Yeats [Yea18b]. We first set-up the problem in Section 3.1 and then give a complete,

self-contained proof of each of the three cases in the following sections, Sections 3.2, 3.3 and 3.4. In particular in Sections 3.3.3 and 3.4.3 we prove two new involutions that together with results from [Yea18b] completes the proof of the conjecture in their respective cases. We assemble everything in Section 3.5 and give the main result of Part I in Theorem 3.5.1.

The second part, Part II, then moves to exploring the combinatorics of a geometric object called the **positive Grassmannian** and a particular map on cells of the positive Grassmannian called **T-duality**. These cells on one side of the map are then related to the on-shell diagrams from $\mathcal{N} = 4$ SYM theory.

We open this part with an introductory chapter, Chapter 4, on the many objects behind T-duality. In Section 4.1 we motivate the study of the positive Grassmannian through one particular decomposition of the Grassmannian called its matroid stratification. Section 4.2 then introduces the main combinatorial objects of interest that are in bijection with cells of the positive Grassmannian, which we call positroids cells as they are indexed by positroids. We end this chapter with Section 4.3, where we look at the geometric objects arising from images of the positive Grassmannian under two different maps and how they are related. In particular in Section 4.3.3, we define T-duality as a combinatorial map which arose from looking at triangulations of these geometric objects, and set-up the motivation for the central question of Chapter 5.

In Chapter 5 we investigate what T-duality looks as a map on Le diagrams, one of the combinatorial objects in bijection with positroids, doing so in two different ways. Starting with Section 5.1, we define an explicit algorithm in Section 5.1.3 for determining the Le diagram resulting from the T-duality map. At the end of the section, in Section 5.1.5, we work through an example of this intricate algorithm. In Section 5.2 we notice that there is actually a nice structure to the Le diagrams created from the above algorithm, giving a more visual perspective of how T-duality affects Le diagrams, the main result being Theorem 5.2.5. Finally in Section 5.3, we prove that the algorithm does indeed give the correct Le diagram, resulting in Theorem 5.3.3. In particular, here we see in Theorem 5.3.1 how viewing T-duality on Le diagrams directly explains the dimensional relationship between the positroid cells on either side of the map.

We conclude this thesis in Part III with Chapter 6, giving some further research directions for both stories in Section 6.1 and tying everything together with some final thoughts in Section 6.2.

THE c_2-INVARIANT

In the first part of this thesis, we present one story on the combinatorics arising in perturbative quantum field theory. Among the many combinatorial connections, this is specifically a story about an arithmetic invariant called the c_2-invariant that is related to scattering amplitudes in scalar ϕ^4 theory and the Feynman integrals in their expansion.

We begin with an introductory chapter on the c_2-invariant, providing the background needed for the specific problem that we will focus on in Chapter 3, which aims to answer one of the conjectured symmetries of the c_2-invariant via combinatorial techniques. As we will see in Section 2.4.2, a combinatorial picture emerges where the c_2-invariant can be thought of as an enumeration problem on graphs, counting certain edge partitions.

2.1. MOTIVATION FROM ϕ^4-THEORY

(**2.1.1**) **4-point graphs.**

The Feynman graphs we are interested in are those arising from 4-point Feynman integrals in four-dimensional ($D = 4$) massless Euclidean ϕ^4-theory. The "4-point" here corresponds to the number of particles in the scattering configuration that we are interested in and the "massless" corresponds to setting masses to 0. Viewing this scalar field theory as a **combinatorial physical theory** (see §5.2 of [Yea17]), the Feynman graphs in question correspond to graphs with 4 external half-edges, or "legs", where the (internal) edges are thought of as consisting of two half-edges, and where every vertex is of degree 4 (with external edges contributing to the degree). As we will be considering the Feynman period in the massless case, these external legs can usually be disregarded outside of playing a role in vertex degree counting. An example of a 4-point graph in ϕ^4-theory is given in Figure 2.1a.

Let G be a 4-point graph in ϕ^4-theory, that is a 4-regular graph, not necessarily simple, with 4 external legs. We will also assume throughout that G is connected. Here $|E(G)|$ is the number of internal edges of G and $\ell(G)$ is the loop order of G.

As we want to talk about Feynman integrals in this combinatorial context, recalling that renormalization was needed to circumvent the divergent nature of these integrals, we need to translate these notions into our setting. To measure how badly a Feynman integral diverges as the energies get large, we can use power counting (counting the exponents) on the integration variables in comparison with the number of such variables, which can then be directly distilled via the Feynman rules into a property of the underlying graph. More precisely, the **superficial degree of divergence** (sdd) of a graph is defined as

$$\mathrm{sdd}(G) = D\ell(G) - \sum_e w(e) - \sum_v w(v),$$

where D is the dimension of space-time, w is the power counting weight function associated to the specific combinatorial physical theory in question and the sums run over all internal edges e and all

vertices v, respectively. In scalar ϕ^4-theory, which has only one type of edge and vertex, edges have weight 2 ($w(e) = 2$) and vertices have weight 0 ($w(v) = 0$). We can see this directly through the Feynman rules in momentum space (see Figure 1.3), where each edge contributes a power of 2 of its associated variable in the denominator of the integrand. Generally, there may be different weights for different edge types and for different vertex types.

We call a graph **divergent** when its sdd is non-negative and **logarithimcally divergent** when it is exactly 0. We then say that a combinatorial physical theory is **renormalizable** if the sdd of a graph in the theory only depends on the structure of the external legs.

In our case, ϕ^4-theory is renormalizable in $D = 4$ and thus the superficial degree of divergence of G is

$$\mathrm{sdd}(G) = 4\ell(G) - 2|E(G)| = 4 - q,$$

where q is the number of external legs. Here we use that there are $2|E(G)| + q = 4|V(G)|$ half-edges and Euler's formula for connected graphs,

$$|V(G)| - |E(G)| + \ell(G) = 1,$$

which gives

$$4\ell(G) - 2|E(G)| = 4(1 + |E(G)| - |V(G)|) - 2|E(G)| = 4 + 2|E(G)| - 4|V(G)| = 4 - q.$$

As $q = 4$, every 4-point graph is logarithmically divergent (because $\mathrm{sdd}(G) = 0$), giving the equality $|E(G)| = 2\ell(G)$. Using Euler's formula again gives us that $|V(G)| = \ell(G) + 1$.

(**2.1.2**) **Feynman periods.**

Instead of looking at the full Feynman integral, we will look at a particular residue of it called the **Feynman period**. In parametric space, the Feynman period (also referred to just as the period) of a graph G is defined as

$$P_G := \int_0^\infty \frac{\mathrm{d}\alpha_1 \cdots \mathrm{d}\alpha_{|E(G)|-1}}{\Psi_G^2 \Big|_{\alpha_{|E(G)|}=1}}, \tag{2.1.1}$$

where we associate a Schwinger parameter α_e to each edge e in G and

$$\Psi_G = \sum_{\substack{T \\ \text{spanning tree}}} \prod_{e \notin T} \alpha_e$$

is the **graph polynomial** or **Kirchhoff polynomial** of G. Note that this integral is independent of the choice of edge $e = |E(G)|$ to set $\alpha_e = 1$. There are many equivalent formulations of this integral, in the different spaces (position, momentum etc.) as well projective versions of each. We refer the reader to [Sch10] and §3.1, 3.2 of [Bro10] for further details.

Note that the term "period" comes from algebraic geometry: looking at P_G in its parametric form Ψ_G (if it exists) is simply a polynomial in the variables α_e with integer coefficients. Thus Ψ_G^{-2} is a rational function with P_G the evaluation of its integral over $\alpha_e \geq 0$. That is, P_G is a period as defined by Kontsevich and Zagier [KZ01] and in the same sense as how multiple zeta values are periods.

It turns out that P_G is well-defined when the graph G is primitive and logarithmically divergent. In fact, these are necessary and sufficient conditions for the convergence of P_G, see Proposition 5.2 of [BEK06].

In this case P_G defines a positive real number which we call the **period of G**. In physics language, G is a 4-point graph with no 1PI divergent subgraphs and algebraically primitive here means primitive for the co-product of the renormalization Hopf algebra on Feynman graphs. In terms of graphs we can define primitivity with logarithmic divergence included as follows.

Definition 2.1.1 — *A graph G is **primitive** (or **primitive divergent**) if*

- *$|E(G)| = 2\ell$; where $|E(G)|$ is the number of edges in G and $\ell = \ell(G)$ is the loop number of G, and*
- *every non-empty proper subgraph $\gamma \subset G$ has $|E(\gamma)| > 2\ell(\gamma)$.*

We note that the second condition tells us our primitive graphs have no divergent subgraphs (subdivergences), as

$$\mathrm{sdd}(\gamma) = 4\ell(\gamma) - 2|E(\gamma)| \geq 0 \quad \Longleftrightarrow \quad 2\ell(\gamma) \geq |E(\gamma)|.$$

We also have that any primitive graph with at least three vertices must be simple, namely without multiple edges and self-loops. The only non-simple primitive graph is given in Figure 2.1a, which is also the only graph of loop order $\ell = 1$. This graph also (most likely) corresponds to the only rational ϕ^4 period and has by far the easiest period to calculate.
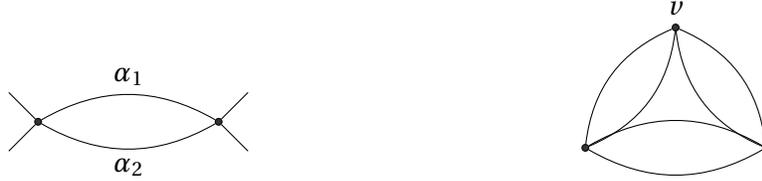
(a) A primitive one-loop Feynman graph.   (b) The unique completion of (a).

FIGURE 2.1: The primitive graph of one loop and its completion.

Example 2.1.2 — (A period calculation) *Consider the (unique) primitive graph with one loop G in Figure 2.1a. Using Equation* (2.1.1) *we get*

$$P_G = \int_0^\infty \frac{\mathrm{d}\alpha_1}{\Psi_G^2}\bigg|_{\alpha_2=1} = \int_0^\infty \frac{\mathrm{d}\alpha_1}{(\alpha_1 + \alpha_2)^2}\bigg|_{\alpha_2=1} = \int_0^\infty \frac{\mathrm{d}\alpha_1}{(\alpha_1 + 1)^2} = -\frac{1}{1 + \alpha_1}\bigg|_0^\infty = 1.$$

As the loop order ℓ increases, so does the number of non-isomorphic Feynman graphs and the difficulty of the integrations. Starting from $\ell = 3$, we already see the appearance of special numbers, namely multiple zeta values (see Section 2.1.4), as periods. For example, the complete graph on four vertices K_4 can be viewed as a primitive ϕ^4 graph and has a period of $6\zeta(3)$.

(**2.1.3**) **Period symmetries.**

One way to tackle the explosion of graphs is to find symmetries of the period, allowing us to create equivalence classes of graphs with the same period. One particularly important symmetry is called

completion, which reduces the problem of calculating periods for primitive divergent graphs to one on 4-regular graphs. Note that 4-regular graphs can also be thought of as ϕ^4 graphs with no external legs and thus no external momenta, sometimes called **vacuum graphs**.

As we are looking at graphs with 4 external legs where each vertex is of degree 4, notice that we can uniquely "complete" such graphs by adding a new vertex connected to all the external half-edges creating a 4-regular graph which we say is the **completion** of the primitive graph. Conversely, given a 4-regular graph and then "decompleting" it by removing a vertex, we are not guaranteed to always get a primitive divergent graph, nor will we always get the same resulting graph when picking a different vertex to remove. Thus first we need a notion of primitivity for these 4-regular graphs such that removing any vertex indeed gives a primitive divergent graph which we then call a **decompletion** of the 4-regular graph.

Definition 2.1.3 — (Definition 2.4 of [Sch10]) *A 4-regular graph G with ≥ 3 vertices is called **completed primitive** if the only way to split G into multiple connected components with 4 edge cuts is to separate off a vertex, in other words, there are only trivial 4-edge cuts. In other words, G is **internally 6-edge connected**. In this case, we say that G has **loop order** ℓ if $G - v$ has loop order ℓ for any vertex v.*

As an example, Figure 2.1 gives a primitive graph and its unique completion, which is completed primitive. The complete graph on 5 vertices K_5 is another example of a completed primitive graph. Notice that any decompletion of K_5 gives the primitive graph K_4. In Section 2.5.1, we briefly discuss why this completed primitive condition is necessary by relating it back to having no subdivergences in primitive graphs.

Proposition 2.1.4 — (Proposition 2.6 of [Sch10]) *Let G be a 4-regular graph and v any vertex in G. Then G is completed primitive if and only if $G - v$ is primitive.*

With this notion of primitivity for completed graphs, Schnetz [Sch10] then proved that the period is completion invariant: any two decompletions of the same 4-regular completed primitive graph have the same period.

Theorem 2.1.5 — (Theorem 2.7 of [Sch10]) *Let G be a 4-regular completed primitive graph of loop order ℓ. The period of $G - v$ for any vertex v is the same for all choices of v.*

This completion symmetry tells us that rather than looking at periods of 4-point graphs, we can instead focus on 4-regular graphs. Note that completion considerably reduces the number of relevant graphs at each loop order ≥ 5.

Other than completion, there are currently four other known period symmetries which we briefly describe below. For each of the symmetries, we are taking our graphs G to be completed primitive, that is 4-regular and internally 6-edge connected. The last three symmetries tells us that under their respective transforms, the period stays invariant.

- **Product identity** for 3-vertex joins: When there is a 3-vertex cut, in which case we call G **reducible**, the period decomposes as a product of two periods corresponding to the two sides of the cut where to each side a triangle is added to the three vertices in the cut. In other words,

this occurs when G is the 3-vertex join of two connected completed primitive graphs. For primitive divergent graphs, this product identity holds for 2-vertex joins.

- **Planar duality**, also called the **Fourier identity**: One natural period identity arises from reinterpreting the Fourier transform as a graph transform. Graphically, this is taking the planar dual of a decompletion $G - v$, and then completing it (if possible).

- **Twist identity**: This identity due to Schnetz arises from the twist transform which uses a 4-vertex cut that separates G into two connected subgraphs and reattaches one side of this separation in a particular manner.

- **Fourier split identity** – This is a relatively new period symmetry which combines the ideas of the Fourier and twist transforms. Instead of just reattaching the subgraph in the twist transform, the Fourier split transform first takes a dual of this subgraph and then reattaches it in a particular manner.

For more details on the first three symmetries, see §2.5-2.7 of [Sch10] and for the Fourier split see §2.3 of [Hu+18]. We note that in the context of Feynman integrals and primitive divergent graphs, the product identity for 2-vertex joins is a well-known result and the Fourier identity has been used as early as [BK95].

While perhaps not as hard as the Feynman integral to compute, the period still poses quite the computational challenge even after utilizing these symmetries. As such, related invariants were introduced to study properties of the period such as the c_2-invariant, which is our main focus here, the extended graph permanent [CDY16] and the Hepp bound [Pan19]. It is either conjectured or proven that these invariants have all the above symmetries of the period.

(**2.1.4**) **Multiple zeta values.**

The number-theoretic content of the period is itself an interesting problem, in particular due to its relationship to the underlying geometries of the period, see for instance [BEK06; Bro10; BS12]. By "geometries", we refer to types of varieties that appear as defined by the vanishing of the graph polynomial. In some sense these varieties control the period, giving an algebro-geometric feel to the period. In particular, many of the early computable periods [BK95; Sch10] were all found to be rational linear combinations of multiple zeta values.

For $s_1, \ldots, s_k$ positive integers and $s_1 > 1$, the **multiple zeta value** (MZV) is defined to be

$$\zeta(s_1, \ldots, s_k) = \sum_{n_1 > \cdots > n_k > 0} \frac{1}{n_1^{s_1} \cdots n_k^{s_k}},$$

and we call the sum of the exponents $n = s_1 + \cdots + s_k$ the **weight** of the MZV.

Viewing an MZV through its iterated integral representation, we can then also think of its weight in the sense of Kontsevich-Zagier periods, as the minimum number of nested integrals needed to write it as such a period. As such, we refer to the **weight** of a Feynman period in this same way, as the mininum number of integrals needed to write it as an integral of an algebraic function over an algebraic domain. From the definition of the Feynman period, we can already say that an upper bound for its weight is $2|E(G)| - 1 = 2\ell(G) - 1$. We will discuss more about weight in Section 2.4.

We end this section on ϕ^4-theory and periods with a beautiful result [BS15] that gives a formula for the period of an infinite family of graphs called *zig-zag graphs*, denoted by Z_ℓ where ℓ is the loop order of the graph. The formula was first conjectured by Broadhurst and Kreimer in [BK95]. This special family includes the complete graph $Z_3 = K_4$ and their completions form the circulant graphs $C_{\ell+2}(1,2)$.

Theorem 2.1.6 — (Theorem 1.1 of [BS15]) *The period of the zig-zag graph Z_ℓ is given by*

$$P_{Z_\ell} = \frac{4(2\ell - 2)!}{\ell!(\ell - 1)!}\left(1 - \frac{1 - (-1)^\ell}{2^{2\ell - 3}}\right)\zeta(2\ell - 3).$$

2.2. DEFINING THE c_2-INVARIANT

Looking at the period in its parametric form (2.1.1), notice we are integrating over the denominator Ψ_G^2, which is just a polynomial in $|E(G)|$ variables. In particular, in order to understand and characterize properties of the period, we need to understand the structure of Ψ_G. This motivates the study of the zeros of Ψ_G and the polynomials (denominators) that arise after several steps of integration.

In 1997, inspired by the work of Broadhurst and Kreimer [BK95; BK97], Kontsevich informally conjectured that for all graphs, the function taking prime powers q to the number of zeros (also known as point counts) of Ψ_G over the finite finite $\mathbb{F}_q$ is a polynomial in q. This stemmed from the appearance of multiple zeta values in certain ϕ^4 periods, which also led to the conjecture that all ϕ^4 periods were linear combinations of products of multiple zeta values. As such, it was thought that there must have been some reason these special numbers were showing up in the period, namely due to specific structures [BEK06] of the algebraic varieties defined by Ψ_G, which in turn would imply Kontsevich's conjecture. However, the conjecture turned out to be false in general. Belkale and Brosnan [BB03] first proved that Kontsevich's conjecture was generically false in 2000 and later Doryn [Dor11] and Schnetz [Sch11] independently found counterexamples specifically in the class of ϕ^4 graphs. Brown and Schnetz [BS12] then further showed that even under certain "physical" constraints on the graphs, Kontsevich's conjecture still could not hold.

Nevertheless, this point count function, which we denote by $[\Psi_G]_q$, still has an interesting connection to the period. When Kontsevich's conjecture is indeed true for some graph G, then the coefficient of q^2 in $[\Psi_G]_q$, which is now a constant with respect to q, was found to give information on the weight of the multiple zeta values that appear in the period of this graph. In [Sch11], Schnetz introduced the following arithmetic invariant.

Definition 2.2.1 — (Theorem 2.9 of [Sch11]) *Let $q = p^n$ be a prime power and $\mathbb{F}_q$ the finite field with q elements. Let G be a connected graph with at least 3 vertices. Then the $\mathbf{c_2}$-invariant of G at q is*

$$c_2^{(q)}(G) \equiv \frac{[\Psi_G]_q}{q^2} \mod q, \tag{2.2.1}$$

where $[\Psi_G]_q$ is the number of zeros of Ψ_G in $\mathbb{F}_q^{|E(G)|}$.
Denote by $\mathbf{c_2(G)}$ the sequence of $c_2^{(q)}(G)$ for all prime powers q.

To show that this invariant is well-defined, Schnetz proved that $[\Psi_G]_q$ was indeed divisible by q^2. In particular, the $c_2^{(q)}(G)$ would then exactly be the quadratic coefficient of $[\Psi_G]_q$ if it was indeed a polynomial in q, hence the name "c_2". When specifically referring to the c_2 at primes $q = p$, we will often use p instead of q.

In a sense, the c_2-invariant is measuring how badly Kontsevich's conjecture fails for a graph. More importantly, other than its combinatorial flavour which we will see shortly, the interest in the c_2-invariant lies in how it seems to detect the types of numbers that appear in the period and thus telling us something about the geometries underlying them [BS12; BS13; Sch19]. In Section 2.5, we give a brief overview of some of the results in this direction.

This connection between the c_2-invariant and the period is further strengthened by the following conjecture, which says that the c_2 is a period invariant.

Conjecture 2.2.2 — (Remark 2.11 (2) of [Sch11], Conjecture 2 of [BS13])
Let G_1 and G_2 be primitive divergent graphs. If their periods coincide, that is, $P_{G_1} = P_{G_2}$, then

$$c_2(G_1) = c_2(G_2).$$

Currently, this conjecture holds for all known examples. However, whether all the symmetries of the period also hold for the c_2-invariant is still unknown, which we would need for the above conjecture to be true. There has been some progress along this front: first shown in 2013, Doryn [Dor17] proved that the c_2-invariant is indeed an invariant across the four representations of the period under a condition for the graph called *duality admissibility*. This condition encompasses all planar graphs, and thus the c_2 remains invariant under planar duality, as does the period. For the completion symmetry, which for the c_2 we refer to as the **completion conjecture**, in 2018 Yeats [Yea18b] first made headway on this in some special cases, and in Chapter 3 we complete the argument for when $q = 2$.

Finally, there wouldn't be much interest in the c_2-invariant if computationally it was as hard as the period. In the following two sections, we will see how to transform the c_2 into a more tractable form, along the way illuminating its combinatorial nature. In this transformed form, the c_2-invariant becomes relatively easy to calculate compared to the period, at least for low loop orders or small primes $q = p$, while still encapsulating many of the properties of the period and its underlying graph.

2.3. GRAPH POLYNOMIALS

Before we can work on massaging Equation (2.2.1), we need to familiarize ourselves with the function in the numerator, the graph polynomial. Let G be a connected graph. Recall that the graph polynomial of G is defined as

$$\Psi_G = \sum_{\substack{T \\ \text{spanning tree}}} \prod_{e \notin T} \alpha_e,$$

where the α_e are parameters associated to each edge e and the sum runs over all spanning trees of G. Notice that Ψ_G is a homogeneous polynomial of degree $|E(G)| - |V(G)| + 1$ in $|E(G)|$ variables. There is also a nice deletion-contraction relation, where for any edge e in G we have

$$\Psi_G = \alpha_e \Psi_{G \setminus e} + \Psi_{G/e}, \tag{2.3.1}$$

since we can partition the sum based on whether a spanning tree of G contains e or not. To see this, notice that the set of spanning trees of G not containing e is equal to the set of spanning trees on $G \setminus e$ and thus α_e can be factored out of the monomials. The set of spanning trees of G containing e is then equal to the set of spanning trees of G/e.

As the graphs get larger, these polynomials explode as well, and thus the first step in transforming the c_2 is to find smaller polynomials in which we can take point counts of. To do this, we need the theory of some related graph polynomials, which following Brown we call **Dodgson polynomials**, and which stem from the deletion-contraction relation of the graph polynomial and Kirchhoff's matrix-tree theorem.

As first introduced in [Bro10] in the context of calculating periods and the theory of **denominator reduction**, which we will cover in the next section, Dodgson polynomials were originally defined up to overall sign. Up until very recently, in the subsequent work on the c_2-invariant this overall sign did not matter. In 2019, Schnetz [Sch19] re-derived much of the work on Dodgson polynomials to account for the correct signs, which we present here. Note that in the $q = 2$ case, which is our main focus in Chapter 3, these signs won't be needed.

(**2.3.1**) **Determinantal framework.**

The underlying framework of this theory on graph polynomials relies on the fact that the graph polynomial can be seen as the determinant of a particular matrix. To represent the graph polynomial as such, we first define the following.

Definition 2.3.1 — *Given a connected graph G, choose an arbitrary orientation on the edges and an order ι on the edges and vertices of G, where the edges are ordered before the vertices,*

$$\iota : E(G) \cup V(G) \quad \longrightarrow \quad \{1, \ldots, |E(G)| + |V(G)|\}.$$

We denote by ι_e or ι_v, to mean the order of edge e or vertex v under ι, respectively.

Let $\mathscr{E}_G$ be the $|V(G)| \times |E(G)|$ signed incidence matrix, with any one row (corresponding to a vertex) removed. That is, for all vertices v except one, and all edges e

$$[\mathscr{E}_G]_{v,e} = \begin{cases} 1 & \text{if } e = (v, u), u \in V(G), \\ -1 & \text{if } e = (u, v), u \in V(G), \\ 0 & \text{otherwise.} \end{cases}$$

Let Λ be the diagonal matrix of indeterminates α_e for e in $E(G)$ in the chosen order, that is

$$[\Lambda]_{e_i, e_j} = \begin{cases} \alpha_{e_i} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

*Then we define the **expanded Laplacian** of G to be*

$$L_G = \left[\begin{array}{c|c} \Lambda & \mathscr{E}_G{}^T \\ \hline \mathscr{E}_G & 0 \end{array} \right],$$

which is an $(|E(G)| + |V(G)| - 1) \times (|E(G)| + |V(G)| - 1)$ matrix, with rows and columns ordered by ι.

While this matrix is not well-defined, as it depends on the choice of row removed in $\mathscr{E}_G$ as well as the choice of orderings and orientation, we have, for any such choice

$$\Psi_G = (-1)^{|V(G)|-1} \det(L_G). \tag{2.3.2}$$

To see why this holds, following the proof in §2.2 of [Bro10], we need a lemma due to Kirchhoff which we can think of as essentially the matrix-tree theorem.

Lemma 2.3.2 — ([Kir47]) *Let U be a subset of edges of G such that*

$$|E(G \setminus U)| = \ell(G) = |E(G)| - |V(G)| + 1.$$

Let $\mathscr{E}_G(G \setminus U)$ denote the square $(|V(G)| - 1) \times (|V(G)| - 1)$ matrix obtained from $\mathscr{E}_G$ by removing the columns indexed by the edges of $G \setminus U$ (recall that $\mathscr{E}_G$ already has one row removed). Then

$$\det \mathscr{E}_G(G \setminus U) = \begin{cases} \pm 1 & \text{if } U \text{ is a spanning tree of } G, \\ 0 & \text{otherwise.} \end{cases}$$

We also need the Leibniz formula for determinants of square $n \times n$ matrices A with entries $a_{i,j}$

$$\det(A) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)},$$

where S_n is the symmetric group on n elements and $\operatorname{sgn}(\sigma)$ is the sign of the permutation σ

$$\operatorname{sgn}(\sigma) = (-1)^{N(\sigma)}.$$

Here $N(\sigma)$ is the number of inversions in σ, i.e. the number of pairs (i, j) such that $i < j$, but $\sigma(i) > \sigma(j)$.

Then, expanding the determinant of L_G via the Leibniz formula, we have that

$$\det(L_G) = \sum_{U \subset E(G)} \det \left[\begin{array}{c|c} 0 & \mathscr{E}_G(G \setminus U)^T \\ \hline \mathscr{E}_G(G \setminus U) & 0 \end{array} \right] \prod_{e \notin U} \alpha_e = \sum_{\substack{U \subset E(G) \\ |E(G \setminus U)| = \ell(G)}} (-1)^{|V(G)|-1} \det(\mathscr{E}_G(G \setminus U))^2 \prod_{e \notin U} \alpha_e.$$

The first equality comes from choosing $E(G \setminus U)$ to be the fixed points of permutation σ and passing to the minor with the corresponding rows and columns removed, setting $\alpha_e = 0$ for $e \in U$. Now, the terms corresponding to any permutation with fixed points in the last $|V(G)| - 1$ columns vanish, and thus the only fixed points for non-zero terms in the sum are those in the first $|E(G)|$ columns. Additionally, the sign of a permutation σ is the same as the sign of the permutation π created by removing the fixed points of σ. The second equality comes from a determinantal identity on block matrices and the restriction in the sum occurs as all the terms with $|E(G \setminus U)| \neq \ell(G)$ vanish since the rank of $\mathscr{E}_G$ is $|V(G)| - 1$ (recall we are removing the columns corresponding to $E(G \setminus U)$). Then, applying Lemma 2.3.2 and rearranging gives Equation (2.3.2) since $\det(\mathscr{E}_G(G \setminus U))^2 = 1$ if U is a spanning tree of G.

(**2.3.2**) **Dodgson polynomials.**

Under this determinantal framework and looking at the deletion-contraction relation (2.3.1) for Ψ_G, we see that removing an edge e to get $\Psi_{G\setminus e}$ corresponds to the minor where the row and column indexed by e is removed. This comes from noticing that $\Psi_{G\setminus e}$ is the coefficient of α_e in $\det(L_G)$. Contracting an edge e to get $\Psi_{G/e}$ then corresponds to setting α_e to zero. In either case, some consideration is needed for the signs. Motivated by this observation, we can extend the definition to minors of L_G which we call **Dodgson polynomials**, or just Dodgsons.

Before defining Dodgsons, we need to take care of the signs. As originally defined in §2.3 of [Bro10], the edges corresponding to the rows and columns being removed from L_G were taken as sets and thus the original Dodgsons were only defined up to overall sign, depending on the minor taken. We refer to these as **unsigned Dodgsons**.

The main idea here is that to get the correct signs, we need to keep track of the order in which the rows and columns are being removed, as this affects the sign in the coefficient of the corresponding minor in the cofactor expansion of L_G following this order. In particular, this sign is determined by the indices of the rows and columns being removed and the orders of these removals.

To get the correct signs, we lift the sets of edges being removed to words on $E(G)$, where the letters are the edges ordered with respect to the chosen ordering ι from Definition 2.3.1. For convenience, we interchangeably use the terms edges and letters, as well as refer to edges e by their numerical position ι_e as given by ι. We also use set notation with these words when we are referring to the set of edges in the words. For such a word $w = w_1 \cdots w_n$ in the letters $E(G)$, let ι_w be the sum of the orders ι_{w_j} of the letters w_j of w under ι. We also define

$$\text{sgn}(w) = \begin{cases} 0 & \text{if } w \text{ has repeated letters,} \\ \text{sgn}(\pi) & \text{otherwise, where } \pi(i) = j \text{ if } w_j \text{ is the } i\text{th smallest letter in } w \text{ with respect to } \iota. \end{cases}$$

As an example, the word $w = 31$, where the edges are ordered naturally, has $\text{sgn}(31) = -1$ as the permutation corresponding to w is $\pi = 21$. To add a letter x to w, we have

$$\text{sgn}(wx) = (-1)^{N(\sigma)} = (-1)^{N(\pi)+|\{j\,:\,i<j\,,\,\sigma(i)=n+1>\sigma(j)\}|} = (-1)^{|\{w_j \in w\,:\,x<w_j\}|}\,\text{sgn}(w),$$

where σ is the permutation corresponding to the word wx. The equalities hold as the inversions in σ coming from x are exactly those involving $\sigma^{-1}(n+1)$, which then correspond to the letters of w larger than x (with respect to ι). In other words, the sign of a word is the parity of how many transpositions it takes to get the edges back in their order under ι. Now we are ready to define our Dodgson polynomials.

Definition 2.3.3 — (Definition 7 of [Sch19]) *Let I and J be words in the edges such that $|I| = |J|$ and let K be a subset of edges of G. Denote $L_G(I, J)_K$ the matrix obtained from L_G by removing rows indexed by I and columns indexed by J, and setting $\alpha_e = 0$ for $e \in K$. Then the **Dodgson polynomial** is defined to be*

$$\Psi_{G,K}^{I,J} = (-1)^{|V(G)|+\iota_I+\iota_J-1}\,\text{sgn}(I)\,\text{sgn}(J)\,\det L_G(I, J)_K,$$

We define $\Psi_{G,K}^{I,J} = 0$ if $|I| \neq |J|$ and define the empty determinant to be 1.

While there is a choice of ordering ι and vertex removed in $\mathcal{E}_G$ for the matrix L_G, Dodgsons do not depend on this choice (see Lemma 9 of [Sch19]). When the graph G is clear from the context, we will drop the subscript of G. We also omit empty indices. When I, J, K are all empty, we recover Ψ_G.

Most, if not all, of the properties of unsigned Dodgsons (see §2.2, 2.3 of [Bro10]) also hold for Dodgsons, and in particular Dodgsons also satisfy a deletion-contraction relation. The following proposition captures some of these properties, see §2 of [Sch19].

Proposition 2.3.4 — *Let I and J be words in the edges such that $|I| = |J|$, and K a subset of edges of G.*

- $\Psi^{I,J}_{G,K} = \Psi^{J,I}_{G,K}$.

- *For any edge $e \in E(G) \setminus (I \cup J \cup K)$, there is a deletion-contraction relation*

$$\Psi^{I,J}_{G,K} = \alpha_e \Psi^{Ie,Je}_{G,K} + \Psi^{I,J}_{G,K\cup e} = \alpha_e \Psi^{I,J}_{G\setminus e,K} + \Psi^{I,J}_{G/e,K}.$$

- *It follows from the relation that we can always pass to a minor of G*

$$\Psi^{I,J}_{G,K} = \Psi^{I',J'}_{G'},$$

  where $G' = G \setminus (I \cap J)/(K \setminus (I \cup J))$, $I' = I \setminus (I \cap J)$, and $J' = J \setminus (I \cap J)$.
  By doing so, we can assume that $I \cap J = K = \emptyset$.

- *If $\Psi^{I,J}_{G,K} \neq 0$, the degree of $\Psi^{I,J}_{G,K}$ is $\ell(G) - |I|$.*

Lastly, like how the monomials of Ψ_G correspond to spanning trees of G, there is a nice combinatorial interpretation of the monomials appearing in Dodgsons. Once again, like for Equation (2.3.2), expanding the determinant of a minor of L_G and using Lemma 2.3.2 gives the following theorem.

Theorem 2.3.5 — (Proposition 23 of [Bro10]) *Suppose $I \cap J = K = \emptyset$. Then we have*

$$\Psi^{I,J}_{G} = (-1)^{\iota_I + \iota_J} \operatorname{sgn}(I) \operatorname{sgn}(J) \sum_{U \subset G\setminus(I\cup J)} \det(\mathscr{E}_G(G \setminus (U \cup I))) \det(\mathscr{E}_G(G \setminus (U \cup J))) \prod_{e \notin U} \alpha_e,$$

where the sum runs over all subgraphs U such that $U \cup I$ and $U \cup J$ are both spanning trees in G.

Ignoring the signs, what this theorem is saying is that we can view Dodgson polynomials through the possible shapes of the underlying graph after some deletions and contractions. For $U \cup I$ to be a spanning tree in G, we must have that U is a spanning tree in $G \setminus J/I$ since U are edges not in I and J. Similarly for $U \cup J$ to be a spanning tree, U must be a spanning tree in $G \setminus I/J$. The polynomial can then be thought of as an "intersection" of graphs $(G \setminus J/I) \cap (G \setminus I/J)$ where the graphs represent the spanning trees and $\cap$ is taken to mean the resulting polynomial of common terms which are spanning trees in each minor. When $K \neq \emptyset$, this corresponds to contracting the edges in K (that are not in I, J) in both graphs. When $I \cap J \neq \emptyset$, this corresponds to removing the edges in $I \cap J$ in both graphs.

(**2.3.3**) **Dodgson identities.**

In addition to having these nice properties for general Dodgsons, there is also a multitude of other relations falling into two types:

- graph-specific identities,
- determinantal identities.

The first type stems from the combinatorial nature of Dodgsons, which are enumerating particular spanning trees of minors. Thus we can get identities, sometimes also using determinantal identities, based on specific structures within a graph. Some structures include cuts, cycles, vertex-joins, 3-valent vertices, and triangles, see [Bro10; BS12; BSY14; Sch19].

As a simple example, consider a graph with a 3-valent vertex v. We immediately see that if we label the three adjacent edges of v as $1, 2, 3$, then we have $\Psi^{123,123} = 0$, as removing all three edges from the graph disconnects v and thus there are no possible spanning trees. Another identity occurs by noticing that removing any two of the edges and contracting the third edge adjacent to v results in the same underlying graph. Thus we have the identity $\Psi_3^{12,12} = \Psi_2^{13,13} = \Psi_1^{23,23}$.

The second type stems from the determinantal framework of Dodgsons which allows us to use known determinantal identities and formulas in the context of Dodgsons. Two determinantal identities that we will highlight here are what are called Plücker identities and Dodgson identities.

Based on the Plücker relations, we have the following formula (§2.4 of [Bro10]) which we call the **Plücker identities**

$$\sum_{k=n}^{2n} (-1)^k \Psi_G^{(i_1 \cdots i_{n-1} i_k),(i_n \cdots \hat{i}_k \cdots i_{2n})} = 0,$$

where $i_1 < \cdots < i_{2n}$ is an increasing sequence of edges and $\hat{i}_k$ means removing letter i_k. For example, in the case when $n = 2$, we have

$$\Psi_G^{12,34} - \Psi_G^{13,24} + \Psi_G^{14,23} = 0.$$

An important family of identities are what we call **Dodgson identities**, which are based on Jacobi's determinant formula and Dodgson condensation (§2.5 of [Bro10], §2 of [Sch19]). One special case is due to the classical Dodgson identity

$$\det(M)\det(M^{ij,ij}) = \det(M^{i,i})\det(M^{j,j}) - \det(M^{i,j})\det(M^{j,i}),$$

where M is a square matrix, $i \neq j$ are indices and $M^{I,J}$ is the matrix obtained from M by removing the rows indexed by I and the columns indexed by J. Translating this to our Dodgson polynomials gives the quadratic identity

$$\Psi_K^{I,J}\Psi_K^{Ief,Jef} = \Psi_K^{Ie,Je}\Psi_K^{If,Jf} - \Psi_K^{Ie,Jf}\Psi_K^{If,Je},$$

where I, J, are words of equal length in the edges, $K = K' \cup \{e, f\}$ for some subset of edges K' and any two edges $e, f \notin I \cup J \cup K'$. In particular, when $I = J = K' = \emptyset$

$$\Psi_{12}\Psi^{12,12} = \Psi_2^{1,1}\Psi_1^{2,2} - \left(\Psi^{1,2}\right)^2, \tag{2.3.3}$$

where we are using the property of Dodgsons that we can always pass to $K \setminus (I \cup J)$ in the subscript since the entries α_e for $e \in I \cup J$ are removed from the matrix via the removal of rows I and columns J. We can also think of Equation (2.3.3) as taking a particular coefficient of the more general quadratic identity above.

(**2.3.4**) **Spanning forest polynomials.**

Following the discussion in Section 2.3.2, we saw that we could interpret Dodgson polynomials via their underlying graphs. However, we are still dealing with the common spanning trees between two

possibly different graphs. In an effort to better understand these polynomials combinatorially, Brown and Yeats introduced spanning forest polynomials [BY11].

Definition 2.3.6 — (Definition 9 of [BY11]) *Let $P = P_1 \cup \cdots \cup P_k$ be a set partition of a subset of the vertices of G. Let F be a spanning forest that partitions the vertices of P exactly into $P_1 \cup \cdots \cup P_k$. More precisely, if $F = T_1 \cup \cdots \cup T_k$ then each tree T_i of F contains all the vertices in P_i and no other vertices of P, and possibly other vertices in $V(G) \setminus P$. Then we say that the spanning forest F is **compatible** with the vertex partition P. Note that the vertices not in P can belong to any tree of F.*

*We define a **spanning forest polynomial** of G to be*

$$\Phi_G^P = \sum_F \prod_{e \notin F} \alpha_e,$$

where the sum runs over all spanning forests F that are compatible with P.

To represent spanning forest polynomials graphically, we can associate a shape or colour to each part of P and draw the vertices in P accordingly on the graph G.

Like Dodgsons, spanning forest polynomials also satisfy deletion-contraction relations.

Proposition 2.3.7 — (Proposition 10 of [BY11]) *Let e be an edge in G and let P be a partition of a subset of vertices of G. Then we have*

$$\Phi_G^P = \begin{cases} \alpha_e \Phi_{G \setminus e}^P & \text{if the ends of } e \text{ are in different parts of } P, \\ \alpha_e \Phi_{G \setminus e}^P + \Phi_{G/e}^{P/e} & \text{otherwise,} \end{cases}$$

where P/e is the partition created from P by identifying the ends of e if they appear in P.

To relate spanning forest polynomials back to Dodgsons, instead of viewing Dodgsons through the matrix-tree theorem like in Theorem 2.3.5, we can interpret them via the all-minors matrix-tree theorem [Cha82]. As a consequence, Dodgsons can be viewed as sums of spanning forest polynomials.

Theorem 2.3.8 — (Proposition 12 of [BY11]) *Let I, J and K be sets of edges of G where $|I| = |J|$. Then we can write*

$$\Psi_{G,K}^{I,J} = \sum f_P \Phi_{G \setminus (I \cup J \cup K)}^P,$$

where $f_P \in \{-1, 1\}$, and the sum runs over all partitions P of the vertices in $(I \cup J \cup K) \setminus (I \cap J)$ such that all forests compatible with P are spanning trees in both $G \setminus I / (J \cup K)$ and $G \setminus J / (I \cup K)$.

Following the proof in [BY11], comparing with Theorem 2.3.5 we can determine the sign f_P via the appropriate determinants. Passing to the case when $I \cap J = K = \emptyset$, notice from that theorem the subgraphs U are exactly the forests in the spanning forest polynomials above since $U \cup I$ and $U \cup J$ must be spanning trees in G. Equivalently, for a fixed forest U, this condition means that I and J must be spanning trees in $G \setminus \bar{U} / U$ where $\bar{U}$ are the edges in $G \setminus (I \cup J \cup U)$.

Now in $G \setminus \bar{U} / U$, each tree of U has been contracted into a vertex and since the only edges left are those in $I \cup J$, each contracted vertex also tells us which vertices of $I \cup J$ are in the same tree of U. That is, the vertices of $G \setminus \bar{U} / U$ partition the vertices in $I \cup J$ exactly into P, which is the partition that U is compatible with. In particular, any forest that is compatible with P gives a graph isomorphic to

$G \setminus \bar{U}/U$. Thus, in $\Psi_G^{I,J}$ the monomials of Φ_G^P all have the same coefficient f_P which is the following product of determinants for any forest U compatible with P

$$\det\left(\mathscr{E}_G(G \setminus (U \cup I))\right) \det\left(\mathscr{E}_G(G \setminus (U \cup J))\right),$$

up to an overall sign.

Similarly to Dodgsons, there are many identities and relations that spanning forest polynomials satisfy, see §2, 3 of [BY11]. In particular, based on the Dodgson identity (2.3.3) we have the following identity which we represent graphically:
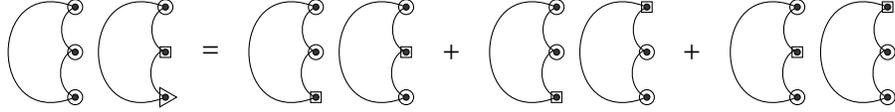
Here each blob represents a spanning forest polynomial, all with the same underlying graph, and the shapes (circle, square, and triangle) indicate the partitions of the three distinguished vertices for each polynomial. Each term in the identity is a product of two spanning forest polynomials.

2.4. DENOMINATOR REDUCTION

With the theory of graph polynomials in place, we return to our main goal of transforming the definition of the c_2-invariant (2.2.1) into a more tractable form. The main idea behind this is an algorithm called **denominator reduction** due to Brown [Bro10; Bro09], which goes back to the period of a primitive divergent ϕ^4 graph G and looks at its denominators after several integrations.

Order the edges according to ι and let $N = |E(G)|$. Recall that the period is defined as

$$P_G := \int_0^\infty \frac{d\alpha_1 \cdots d\alpha_{N-1}}{(\Psi_G)^2}\bigg|_{\alpha_N=1}.$$

Using the deletion-contraction relation (2.3.1), Dodgson identities, and the theory of hyperlogarithms (see §10.2 of [Bro10], §5 of [Bro09] and lecture notes on iterated integrals in this context [Bro13]), we can explicitly compute the first few integrations, where we are integrating with respect to the first five edges. The first two integrations are relatively simple, starting with integrating with respect to α_1

$$I_1 = \int_{\alpha_N=1} \frac{d\alpha_1 \cdots d\alpha_{N-1}}{(\alpha_1 \Psi^{1,1} + \Psi_1)^2} = \int_{\alpha_N=1} -\frac{d\alpha_2 \cdots d\alpha_{N-1}}{(\alpha_1 \Psi^{1,1} + \Psi_1)\Psi^{1,1}}\bigg|_{\alpha_1=0}^\infty = \int_{\alpha_N=1} \frac{d\alpha_2 \cdots d\alpha_{N-1}}{\Psi^{1,1}\Psi_1}.$$

For the second integration with respect to α_2, we use the Dodgson identity (2.3.3) in the denominator

$$I_2 = \int_{\alpha_N=1} \frac{d\alpha_2 \cdots d\alpha_{N-1}}{(\alpha_2 \Psi^{12,12} + \Psi_2^{1,1})(\alpha_2 \Psi_1^{2,2} + \Psi_{12})} = \int_{\alpha_N=1} \frac{\log\left(\frac{\alpha_2 \Psi^{12,12} + \Psi_2^{1,1}}{\alpha_2 \Psi_1^{2,2} + \Psi_{12}}\right)\bigg|_{\alpha_2=0}^\infty}{\Psi^{12,12}\Psi_{12} - \Psi_2^{1,1}\Psi_1^{2,2}} d\alpha_3 \cdots d\alpha_{N-1}$$

$$= \int_{\alpha_N=1} \frac{\log(\Psi_2^{1,1}) + \log(\Psi_1^{2,2}) - \log(\Psi^{12,12}) - \log(\Psi_{12})}{(\Psi^{1,2})^2} d\alpha_3 \cdots d\alpha_{N-1}.$$

Starting from here is where the integrations get more complicated, though reducing with respect to α_3 is still doable. By integration by parts, we first note that

$$\int_0^\infty \frac{\log(cx+d)}{(ax+b)^2}\,dx = \frac{\log(d)}{ab} + \frac{c\left(\log(ab)-\log(cd)\right)}{a(ad-bc)}.$$

Applying this to I_2 after expanding the Dodgsons with respect to edge 3 gives the next integral

$$I_3 = \int_{\alpha_N=1} \frac{\log(\Psi_{23}^{1,1}) + \log(\Psi_{13}^{2,2}) - \log(\Psi_3^{12,12}) - \log(\Psi_{123})}{\Psi^{13,23}\Psi_3^{1,2}}\,d\alpha_4 \cdots d\alpha_{N-1} + \cdots,$$

where we've combined the first terms of integration of each of the four logs in I_2 and omit the rest of the terms in the sums. Another form of I_3 as given in §10.2 of [Bro10] is

$$I_3 = \int_{\alpha_N=1} \frac{\Psi^{123,123}\log(\Psi^{123,123})}{\Psi^{12,13}\Psi^{12,23}\Psi^{13,23}} - \frac{\Psi_{123}\log(\Psi_{123})}{\Psi_1^{2,3}\Psi_2^{1,3}\Psi_3^{1,2}}$$
$$+ \sum_{\{i,j,k\}} \frac{\Psi_{jk}^{i,i}\log(\Psi_{jk}^{i,i})}{\Psi^{ij,ik}\Psi_j^{i,k}\Psi_k^{i,j}} - \frac{\Psi_k^{ij,ij}\log(\Psi_k^{ij,ij})}{\Psi^{ij,ik}\Psi^{ij,jk}\Psi_k^{i,j}}\,d\alpha_4 \cdots d\alpha_{N-1},$$

where the sum runs over the 6 permutations of $\{1,2,3\}$. For the last two integrations, we only present the results, also from the same section of [Bro10]

$$I_4 = \int_{\alpha_N=1} \frac{A}{\Psi^{12,34}\Psi^{13,24}} + \frac{B}{\Psi^{13,24}\Psi^{14,23}} + \frac{C}{\Psi^{12,34}\Psi^{14,23}}\,d\alpha_5 \cdots d\alpha_{N-1},$$

$$I_5 = \int_{\alpha_N=1} \frac{F}{{}^5\Psi(1,2,3,4,5)}\,d\alpha_6 \cdots d\alpha_{N-1},$$

where A,B,C are di-logarithms, and F is a hyperlogarithm of weight 3. The denominator in I_5 is an important combination of Dodgson polynomials called the 5-invariant, which is defined as follows.

Definition 2.4.1 — *Given edges* $1,\ldots,5$ *for a graph* G, *define the **5-invariant** of* G, ${}^5\Psi_G(1,\ldots,5)$, *as*

$${}^5\Psi_G(1,\ldots,5) = \pm\det\begin{pmatrix} \Psi_5^{12,34} & \Psi_5^{13,24} \\ \Psi^{125,345} & \Psi^{135,245} \end{pmatrix} = \pm\left(\Psi_5^{12,34}\Psi^{135,245} - \Psi_5^{13,24}\Psi^{125,345}\right).$$

The 5-invariant is defined up to overall sign and holds for any five distinct edges. Furthermore, permuting the order of the edges only changes the sign of ${}^5\Psi_G(1,\ldots,5)$, see Lemma 87 in [Bro10].

The things to notice in these integrations are that:

- The denominators are specific products of Dodgsons. Specifically the next denominator is the resultant of the two factors of the previous denominator (the determinant of their associated Sylvester matrix), assuming it is not a square.
- The weight of the numerators, in the sense of MZVs and hyperlogarithms, increases by exactly 1 when the previous denominator is not a square, i.e. $1 \to 1 \to \log \to \log \to$ di-log $\to$ weight 3.
- After the fifth integration, the denominator is in a special form.

These observations form the basis of denominator reduction, which in the context of the c_2-invariant allows us to reduce Ψ_G with respect to a sequence of edges.

(**2.4.1**) **Reducing variables.**

As we noticed above, after the fifth integration there was a special denominator which is the 5-invariant. Thus using the theory of hyperlogarithms, when possible, we can then completely determine the denominators of successive integrations, see Corollary 126 of [Bro10]. This gives the following algorithm called **denominator reduction**, see §10 of [Bro10], which gives higher invariants.

Definition 2.4.2 — (Proposition 130 of [Bro10]) *Given a graph G and a sequence of edges* $1,\ldots,|E(G)|$, *we define*

$$D_G^5 = {}^5\Psi_G(1,2,3,4,5).$$

*For $n > 5$, we recursively define the **n-invariant** $D_G^n(1,\ldots,n)$, also referred to as ${}^n\Psi_G(1,\ldots,n)$, as follows: Suppose D_G^n for $n \geq 5$ factors into the following product of linear factors in α_{n+1}*

$$D_G^n(1,\ldots,n) = (A\alpha_{n+1} + B)(C\alpha_{n+1} + D).$$

Then we define

$$D_G^{n+1}(1,\ldots,n+1) = \pm(AD - BC),$$

*which is the resultant of the two factors of D_n. Otherwise, if $D_G^{n+1} = 0$ or D_G^n cannot be factored in such a form (in which case the $n+1$-invariant and higher do not exist), we say that **denominator reduction** ends. If $D_G^{n+1} = 0$ (and thus also for higher invariants) for some sequence of edges, we say that G has **weight drop**. Note that in particular, when D_G^n is a perfect square, then G has weight drop.*

Like the 5-invariant, these higher invariants (if they exist) are defined up to overall sign and do not depend on the order of edges up to that point. However, a different sequence of edges may give a sequence of denominators of a different length. As these D_G^n are indeed the denominators in the period after integrating out the n edges variables, we can say we are "reducing" with respect to the n edges. Note that the factors in the denominators are usually Dodgsons or combinations of Dodgsons.

The notion of weight drop for a graph G directly relates back to the weight of its Feynman period P_G, which is the minimum number of nested integrals needed to write P_G as a period. As the number of edges in ϕ^4 graphs is $|E(G)| = 2\ell$ and we have set one edge variable to 1, we start with $2\ell - 1$ integrals. From the initial integrations above, we notice that the weight of the numerators does not increase in two of the integrations. Thus the maximum weight of a ϕ^4 period is $2\ell - 3$. Note that this maximum is frequently achieved like in the case of the zig-zag graphs, see Theorem 2.1.6. Now, when G has a weight drop, this means that the weight of its period P_G will be less than this maximum weight of $2\ell - 3$. Hence denominator reduction, and thus the c_2-invariant, can tell us about the weight of a graph's period. In Section 2.5.1, we discuss weight drop in the context of the c_2-invariant.

We can also extend the definition of the n-invariant to $n = 3, 4$ at the expense of uniqueness, where now the denominator depends on the order of the edges chosen. For $n = 3$ we have three choices for D_G^3 and taking the edges $1, 2, 3$ in that order we can define

$$D_G^3(1,2,3) = \pm\Psi_G^{13,23}\Psi_{G,3}^{1,2}.$$

Similarly for $n = 4$ we also have three choices for D_G^4, as seen in I_4, but notice we can apply denominator reduction to the D_G^3 above and use a Dodgson identity to get one such choice

$$D_G^4(1,2,3,4) = \pm\Psi_G^{13,24}\Psi_G^{14,23}.$$

As we can denominator reduce any choice of D_G^3 or D_G^4 to the 5-invariant, these choices are sufficient.

This brings us to the first equivalent definition of the c_2-invariant where simultaneously we can get rid of dividing by q^2 and reduce the number of variables in the polynomial we need take point counts of.

Theorem 2.4.3 — (Theorem 29 of [BS12] with Corollary 28 of [BS12] for $n < 5$) *Let G be a connected graph with $2\ell(G) \leq |E(G)|$ and $|E(G)| \geq 3$. Suppose that $D_G^n(e_1, \ldots, e_n)$ is the result of the denominator reduction after $3 \leq n < |E(G)|$ steps. Then*

$$c_2^{(q)}(G) \equiv (-1)^n \left[D_G^n(e_1, \ldots, e_n) \right]_q \mod q.$$

If $2\ell(G) < |E(G)| \geq 4$, then $c_2^{(q)} \equiv 0 \mod q$.

In particular, for any three distinct edges labelled $1, 2, 3$ of G we have

$$c_2^{(q)}(G) \equiv - \left[\Psi_G^{13,23} \Psi_{G,3}^{1,2} \right]_q \mod q.$$

This theorem becomes a powerful tool in computing the c_2-invariant, at least for small q, as we can usually reduce via denominator reduction for many steps before needing to take point counts. One strategy then becomes to find "good" sequences of edges in which we can denominator reduce for as long as possible. In [BS13], Brown and Schnetz was able to exhaustively compute the $c_2^{(p)}$ for all ϕ^4 graphs up to loop order 10 and for small primes p using these methods.

(2.4.2) Becoming a counting problem.

While Theorem 2.4.3 brings us to a much more tractable form of the c_2-invariant, we are still left with the problem of point counting, which itself is a hard problem as the denominators can still have many variables. To tackle this we turn to a consequence of a theorem from number theory, see §2 of [Ax64], which will allow us to determine coefficients instead of counting zeros.

Theorem 2.4.4 — (Corollary of Chevalley-Warning Theorem) *Let F be a polynomial of degree N in N variables, $x_1, \ldots, x_N$, with integer coefficients. Then we have*

$$coefficient\ of\ x_1^{q-1} \cdots x_N^{q-1}\ in\ F^{q-1} \equiv (-1)^{N-1}[F]_q \mod p,$$

where q is some power of the prime p.

Note that this theorem holds when reducing modulo prime p and not when reducing modulo q. An example of this is given in §5 of [Sch19]. Thus, to use this theorem in the context of the c_2-invariant, we will restrict to when $q = p$ is a prime (see also Conjecture 2.5.1).

To apply this theorem to D_G^n, we just need to make sure that the degree of the polynomial matches up with the number of variables. We verify this recursively, recalling that $|E(G)| = 2\ell(G)$ for our graphs of interest (primitive divergent graphs):

– Starting with Ψ_G^2, as the monomials in Ψ_G correspond to the complement edges of spanning trees, Ψ_G^2 is a homogeneous polynomial of degree $2(|E(G)| - |V(G)| + 1) = 2\ell(G) = |E(G)|$.

– As $\Psi_G = \alpha_1 \Psi^{1,1} + \Psi_1$, we have that $\Psi^{1,1}$ must be a homogeneous polynomial of degree $\ell(G) - 1$ and $\Psi^{1,1}$ is of degree $\ell(G)$. This means that $\Psi^{1,1}\Psi_1$ has degree $2\ell(G) - 1 = |E(G)| - 1$ which is exactly the number of variables left after reducing with respect to edge 1.

– As $\Psi^{1,1} = \alpha_2 \Psi^{12,12} + \Psi_2^{1,1}$ and $\Psi_1 = \alpha_2 \Psi_1^{2,2} + \Psi_{12}$ we have that $\Psi^{12,12}\Psi_{12}$ and $\Psi_2^{1,1}\Psi_1^{2,2}$ are both polynomials of degree $\ell(G) - 1 + \ell(G) - 1 = |E(G)| - 2$. Thus the second denominator is of degree $|E(G)| - 2$.

– Following a similar process for D_G^3, we get a degree of $|E(G)| - 3$.

– For $n \geq 3$, suppose D_G^n is a homogeneous polynomial of degree $|E(G)| - n$ in $|E(G)| - n$ variables. If D_G^n factors as $(A\alpha_{n+1} + B)(C\alpha_{n+1} + D)$ then we must have that AD and BC are both homogeneous polynomials of degree $|E(G)| - n - 1$. Thus $D_G^{n+1} = \pm(AD - BC)$ must be homogeneous of degree $|E(G)| - (n+1)$ and in $|E(G)| - (n+1)$ variables.

Thus, $D_G^n(e_1, \dots, e_n)$ satisfies the criterion for Theorem 2.4.4 for $n \geq 3$, where $N = |E(G)| - n$. Applying this to Theorem 2.4.3 and returning the notation $[\cdot]$ back to meaning taking coefficients, we arrive at the second transformation for the c_2-invariant.

Corollary 2.4.5 — *Let G be a connected graph with $2\ell(G) \leq |E(G)|$ and $|E(G)| \geq 3$. Suppose that $D_G^n(e_1, \dots, e_n)$ is the result of the denominator reduction after $3 \leq n < |E(G)|$ steps. Then*

$$c_2^{(p)}(G) \equiv (-1)^{|E(G)|-1}[\alpha_{n+1}^{p-1}\cdots\alpha_{|E(G)|}^{p-1}] \left(D_G^n(e_1, \dots, e_n)\right)^{p-1} \mod p.$$

In particular, for any three distinct edges labelled $1, 2, 3$ of G and when $|E(G)| = 2\ell(G)$ we have

$$c_2^{(p)}(G) \equiv -[\alpha_4^{p-1}\cdots\alpha_{|E(G)|}^{p-1}] \left(\Psi_G^{13,23}\Psi_{G,3}^{1,2}\right)^{p-1} \mod p. \tag{2.4.1}$$

An example of using this equation to calculate the c_2-invariant is given in Example 2.5.5.

What's nice about this version of the c_2-invariant is that we have reformulated what was a problem on counting zeros of polynomials into a problem about finding the coefficient of a particular monomial in some power of said polynomials. Furthermore, these polynomials inherently have a combinatorial nature as they arise from spanning trees or spanning forests of particular graph minors. Thus, we can interpret this coefficient as counting the number of ways to distribute edges into some of these spanning trees or spanning forests, which is a purely combinatorial problem! We will expand on this interpretation in Chapter 3.

Using this method with the theory of spanning forest polynomials and combinatorial techniques, Yeats independently and also with Chorney were able to compute the $c_2^{(p)}$ for entire families of graphs (those which are called *recursively constructible*) and fixed small primes p [Yea16; Yea18c; CY19]. Some examples of these families include particular circulants and toroidal grids.

Combining the two methods of finding "good" sequences of edges for denominator reduction and taking coefficients, Shaw et al. [Hu+18] were able to extend Brown and Schnetz's computations of $c_2^{(p)}$ to all ϕ^4 graphs up to loop order 11 and $p = 7$, with many graphs up to $p = 13$.

(**2.4.3**) **Quadratic denominator reduction.**

Up until now, most of the results on the c_2-invariant did not require us to know the specific signs of Dodgsons. It was not until [Hu+18] that the question of signs became needed, where point counts of sums of Dodgsons were being taken. Furthermore, up until now the main tool in computing the c_2-invariant has been denominator reduction. As we saw, denominator reduction only guarantees up to the first five integrations. After the fifth integration, the use of denominator reduction requires the previous denominator to be factored into two different linear factors. If this does not hold or we have a perfect square, we cannot reduce any further. Often for high loop ordered graphs, even after denominator reduction, point counting would be too computationally intensive. In [Sch19], Schnetz addresses both of these issues; fixing a sign convention for Dodgsons as we presented in Section 2.3.2 and introducing an improved reduction called **quadratic denominator reduction**.

The main idea of quadratic denominator reduction is that instead of using point counts directly, we can use sums of **Legendre symbols**, which enable us to move to quadratic factors instead of linear factors. We can then define a similar reduction algorithm for particular forms of these quadratic factors. This idea stemmed from the observation that if the denominator D_G^n does not factor into linear factors and is a general quadratic in variable α_{n+1}, integrating with respect to α_{n+1} produces a square root of its discriminant in the next denominator. Notice that in the denominator reduction case, this fact is still true, except now the discriminant is a perfect square (hence the ambiguity of the sign)! Thus quadratic denominator reduction is generalizing the standard denominator reduction.

The Legendre symbol for any $a \in \mathbb{F}_q$ where q is an odd prime power is defined as

$$\left(\frac{a}{q}\right) = |\{x \in \mathbb{F}_q : x^2 = a\}| - 1 \ \in \{-1, 0, 1\}.$$

For any polynomial $F \in \mathbb{Z}[\alpha_1, \dots, \alpha_N]$, we can define

$$(F)_q = \sum_{\alpha \in \mathbb{F}_q^N} \left(\frac{F(\alpha)}{q}\right) \in \mathbb{Z}.$$

Then the point count $[F]_q$ can be expressed as

$$[F]_q = q^N - (F^2)_q \equiv -(F^2)_q \mod q.$$

Note that we can also extend these definitions and results to when $q = 2$ (Remark 3.7 of [Sch19]). For more details see §5 of [Sch19].

Relating this back to the c_2, we are now reducing with respect to $(\Psi_G)^4$. Based on the results of an additional integration after denominator reduction ends, Schnetz found two cases of denominators which behave nicely, with respect to the geometry of the varieties, under such an integration, see §6 of [Sch19].

Definition 2.4.6 — (Definition 34 of [Sch19]) *Given a graph G with at least three edges and a sequence of edges $1, \dots, |E(G)|$, we define*

$$^3\Psi_G^2(1,2,3) = \left(\pm^3\Psi_G(1,2,3)\right)^2 = \left(\Psi_G^{13,23}\Psi_{G,3}^{1,2}\right)^2.$$

*For $n \geq 3$, we recursively define the **quadratic n-invariant** $^n\Psi_G^2(1, \dots, n)$, as follows.*
If $^n\Psi_G^2$ for $n \geq 3$ is of the form

$$^n\Psi_G^2(1, \dots, n) = (A\alpha_{n+1}^2 + B\alpha_{n+1} + C)^2,$$

we define

$$^{n+1}\Psi_G(1,\ldots,n+1) = B^2 - 4AC.$$

If $^n\Psi_G^2$ for $n \geq 3$ is of the form

$$^n\Psi_G^2(1,\ldots,n) = (D\alpha_{n+1}^2 + E\alpha_{n+1} + F)(H\alpha_{n+1} + J)^2,$$

we define

$$^{n+1}\Psi_G(1,\ldots,n+1) = DJ^2 - EHJ + FH^2.$$

*Otherwise, if $^{n+1}\Psi_G^2 = 0$ or $^n\Psi_G^2$ cannot be factored in such a form (in which case the quadratic $n+1$-invariant and higher do not exist), we say that **quadratic denominator reduction** ends at step n. If $^n\Psi_G^2 = 0$ (and thus also for higher invariants), we say that G has **weight drop**.*

Note that for quadratic denominator reduction, if $^n\Psi_G^2$ factors into squares of linear factors, the signs are completely determined, and both cases reduce to the standard denominator reduction. That is, when D_G^n exists, we have that $D_G^n = \pm(^n\Psi_G^2)^{\frac{1}{2}}$.

Using quadratic invariants, we can once again reformulate the c_2-invariant.

Theorem 2.4.7 — (Theorem 36 of [Sch19]) *Let q be an odd prime power. Suppose G is a connected graph with $2\ell(G) \leq |E(G)|$ and $|E(G)| \geq 3$. Then*

$$c_2^{(q)}(G) \equiv (-1)^{n-1} \left(^n\Psi_G^2(e_1,\ldots,e_n)\right)_q \mod q,$$

whenever $^n\Psi_G^2$ exists.

The power of quadratic denominator reduction allowed Schnetz to compute and extend previous results on initial prime sequences for the $c_2^{(p)}$ for all ϕ^4 graphs up to loop order 11 and partially to loop orders 12 and 13, see §10 of [Sch19]. This comes from the fact that we can always do a minimum of nine initial reductions (and in numerous graphs we can do many more) using quadratic denominator reduction as compared to the previous seven for standard denominator reduction, see §8 of [Sch19]. Furthermore, using this new technique, Schnetz and Yeats were able to determine a formula for a special family of infinite graphs [SY21], which we briefly describe in Section 2.5.4.

2.5. FAMILIES OF GRAPHS AND NUMERICAL RESULTS

As an avatar for the period, continued interest in the c_2-invariant mostly falls into two categories which are deeply intertwined. The first, based on its relative ease of computability, lies in the types of sequences that can arise as a c_2 and what that means for the geometries of the varieties underlying the corresponding periods. This direction then motivates us to come up with better algorithms for computing the c_2-invariant that can handle higher values of q and larger graphs. In Section 2.4, we introduced two algorithms, denominator reduction and quadratic denominator reduction, which currently form the basis for most of the computational results.

The second category is based on finding symmetries and properties of the c_2-invariant. For example, resolving the period symmetries from Section 2.1.3 in the context of the c_2 or finding graph transforms that preserve the c_2, both of which gives families of graphs with the same c_2. On the one hand, these aid computational results and the discovery of better algorithms as they allow us to either reduce the

number of graphs we need to calculate the c_2 for or extend computations by transforming the problem to a simpler form (with respect to computability). On the other hand, many symmetries and properties of the c_2-invariant have an inherently combinatorial flavour and thus are interesting combinatorial problems in their own right. These combinatorial problems then also use known algorithms to solve some aspect of the problem or even to initially get the problem into a combinatorial form. In Section 2.3, we introduced the notion of Dodgsons and spanning forest polynomials which allows us to interpret the c_2 through spanning trees and spanning forests. Chapter 3 uses this interpretation to solve the c_2 completion conjecture when $q = 2$.

In this final section, we present some of the results in both of these categories: the types of sequences that appear as a c_2, and graph transforms that give infinite families of graphs with the same c_2.

First, we note that we will restrict the c_2 sequence to all primes $q = p$ instead of prime powers $q = p^n$. This is based on the following conjecture by Schnetz which says that prime powers do not give any extra information.

Conjecture 2.5.1 — (Conjecture 2 of [Sch19]) *Let G_1 and G_2 be two graphs with equivalent $c_2^{(p)}$ for all primes p, that is $c_2^{(p)}(G_1) \equiv c_2^{(p)}(G_2) \mod p$. Then for all prime powers $q = p^n$*

$$c_2^{(q)}(G_1) \equiv c_2^{(q)}(G_2) \mod q.$$

Part of the motivation for this conjecture (other than for computational reasons) came from the use of only the primes to identify certain c_2 sequences, see Section 2.5.5. Some evidence for this conjecture was also given in [Sch11] for a few small prime powers. Very recently, Esipova and Yeats [EY21] were able to prove a version of this conjecture where instead of working modulo q they worked modulo p. Specifically, they proved that $c_2^{(q)}(G_1) \equiv c_2^{(q)}(G_2) \mod p$.

Secondly, we note that in the context of numerical results on ϕ^4 graphs, we normally assume the completion conjecture for the c_2-invariant which says that for any completed primitive graph G, decompleting at any vertex v gives the same c_2 (see Section 2.1.3). Finally, when the c_2 is a constant sequence consisting of all c's, we may say that $c_2 = c$.

For the rest of this section, G will be a connected graph with $2\ell(G) \leq |E(G)|$ and at least three edges.

($2.5.1$) **Weight drop, subdivergences and the trivial c_2.**

The first families of graphs that we will consider is when the c_2 is trivial. These are graphs G with $c_2^{(p)}(G) \equiv 0 \mod p$ for all primes p.

One type of family is based on the denominator reduction algorithm (Definition 2.4.2) which tells us that when G has weight drop (i.e. when the denominator is 0 after reducing with respect to some sequence of edges), by Theorem 2.4.3 we have that

$$c_2^{(p)}(G) \equiv 0 \mod p.$$

Thus, any graph G that has weight drop will have a trivial c_2. Intuitively, recall that for a graph the weight of the (possibly transcendental) number that is its period can be thought of as the minimum

number of nested integrals needed to write the period. From denominator reduction, we saw that the maximal weight of a period is $2\ell(G) - 3$, and having weight drop then means that the weight is actually less than this maximal value.

Two graph properties that give rise to weight drop are:

- doubled edges in graphs with at least five edges (Lemma 92 of [Bro10]), and
- 2-vertex reducible (Proposition 36 of [BY11]): These are graphs which have a 2-vertex cut, in other words, they have vertex connectivity 2. For completed primitive graphs, the corresponding property is what we call reducible, having 3-vertex cuts.

Note that being (2- or 3-vertex) reducible and having $c_2 = 0$ is the c_2-invariant equivalent of the product identity (see Section 2.1.3) for the period!

The second type of graph we will look at that has trivial c_2 are those from ϕ^4-theory which are not primitive but still logarithmic divergent. Recall from Definition 2.1.1 that these graphs G contain non-trivial divergent subgraphs γ which are subgraphs such that $|E(\gamma)| \le 2\ell(\gamma)$. To translate this condition into an equivalent graph property, we think of γ as being 4-regular with k external legs giving

$$|E(\gamma)| = \frac{4|V(\gamma)| - k}{2} \le 2\ell(\gamma) \quad \implies \quad |V(\gamma)| \le \ell(\gamma) + \frac{k}{4}.$$

Then, using Euler's formula with the divergent subgraph condition, we have

$$|E(\gamma)| = |V(\gamma)| + \ell(\gamma) - 1 \le 2\ell(\gamma) + \frac{k}{4} - 1,$$

which tells us that $k \le 4$. Now, thinking of these k external legs in the context of the full graph G, this means that G has a cut with k edges where one side of the cut is the subgraph γ. Thus, having subdivergences corresponds to having non-trivial cuts of size at most 4. In §5 of [BSY14], Brown, Schnetz and Yeats were able to prove that if a graph has a non-trivial cut of size at most 4, then its c_2 is trivial!

Note that after completing G (see Section 2.1.3), the same argument still holds for any subdivergence γ. This shows why we need the completed primitive condition (being internally 6-edge connected) for 4-regular graphs to ensure we get primitive divergent decompletions.

(**2.5.2**) **Double triangle reduction.**

Now that we know when graphs have $c_2 = 0$, we switch our focus to a graph transformation called the **double triangle reduction** (DTR). While double triangle reduction does not preserve the period, it does preserve the c_2-invariant. Thus, having this transformation allows us to reduce our problems to smaller graphs and so we only need to look at the c_2's of those without double triangles.

Suppose a graph G has an edge that is shared by exactly two triangles. Call this edge (A, B) with triangles (A, B, C) and (A, B, D). The double triangle reduced graph of G is G with one of the vertices of (A, B), say B, replaced with the edge (C, D). If B has a neighbour not in the triangles, then it is now adjacent to the remaining vertex A. This is illustrated in Figure 2.2. We note that in fact A, B, C and D do not need to be 4-valent.
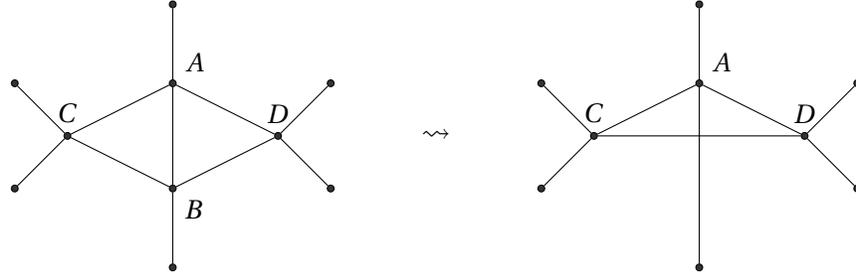
FIGURE 2.2: Double triangle reduction from left to right.
Note: A, B, C and D do not need to be 4-valent.

Theorem 2.5.2 — (Theorem 3.5 of [Hu+18] using Corollary 34 in [BS12], Theorem 35 in [BY11])
Let G be a connected 4-regular graph and G' be the double triangle reduction of G. Then, decompleting at any (same) vertex v, we get

$$c_2(G - v) = c_2(G' - v).$$

In particular, when G_1 is a primitive divergent graph in ϕ^4 and G'_1 is its double triangle reduction, then

$$c_2(G_1) = c_2(G'_1).$$

(**2.5.3**) K_5 **and its descendants.**

Now, assuming the completion conjecture for the c_2-invariant allows us to only look at completed primitive graphs. The results in Section 2.5.1 says that we can then disregard those graphs with vertex connectivity < 4 and with non-trivial 4-edge cuts since they yield trivial c_2's. Finally, double triangle reduction further reduces our graphs to those without double triangles. Combining all three reductions, we are led to the following type of graphs.

Definition 2.5.3 — (Definition 22 of [Sch19]) *A graph with ≥ 5 vertices is a **prime ancestor** if all of the following properties hold:*

   (i) *it is 4-regular,*
  (ii) *it is internally 6-edge connected,*
 (iii) *it has vertex connectivity 4, and*
 (iv) *it has no edge which is shared by exactly two triangles.*

Thus, we only need to investigate the c_2-invariant for decompletions of prime ancestors as each prime ancestor defines an equivalence class of graphs all with the same c_2. In each such class, the graphs are related by double triangle reductions and thus we call these graphs **descendants** of the ancestor. There is one prime ancestor of particular interest which is conjectured to exactly correspond to the class of graphs with $c_2 = -1$.

Conjecture 2.5.4 — (Conjecture 25 of [BS13]) *The only ϕ^4 prime ancestor with $c_2^{(q)} \equiv -1 \mod q$ for all q is the complete graph K_5.*

30

In [LMY19] Laradji, Mishna and Yeats studied the structure of K_5 descendants and gave some partial results on properties of these graphs. One interesting result is that the minimum number of triangles in a K_5 descendant is four.

We verify here that decompletions of K_5 indeed have a c_2 of -1, giving an example of a c_2 calculation.

Example 2.5.5 — (A c_2 calculation) *Consider the complete graph K_4 on four vertices whose completion is K_5, with the edges labelled as follows:*
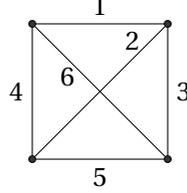
To compute it's c_2-invariant, we use Equation (2.4.1). *First we compute the two Dodgsons in the equation with respect to K_4 using Theorem* 2.3.5 *and the discussion thereafter. For* $\Psi^{12,13}$ *notice that $K_4 \setminus 12/3$ is isomorphic to $K_4 \setminus 13/2$ which is the triangle with edges $4,5,6$ and thus*

$$\Psi^{12,13} = \Psi\left(\begin{array}{c} \text{triangle } 4,5,6 \end{array} \right) = \alpha_4 + \alpha_5 + \alpha_6.$$

For $\Psi_1^{2,3}$ *we compute the common spanning trees of $K_4 \setminus 2/13$ and $K_4 \setminus 3/12$, which is just the edge 5,*

$$\Psi_1^{2,3} = \Psi\left(\begin{array}{c} \text{graph} \end{array} \cap \begin{array}{c} \text{graph} \end{array} \right) = \alpha_4 \alpha_6.$$

Putting everything together we get

$$c_2^{(q)}(K_4) \equiv -[\alpha_4^{q-1} \alpha_5^{q-1} \alpha_6^{q-1}]\, (\Psi^{12,13}\Psi_1^{2,3})^{q-1} \mod q$$

$$\equiv -[\alpha_4^{q-1} \alpha_5^{q-1} \alpha_6^{q-1}]\, (\alpha_4 \alpha_6 (\alpha_4 + \alpha_5 + \alpha_6))^{q-1} \mod q$$

$$\equiv -[\alpha_5^{q-1}]\, (\alpha_4 + \alpha_5 + \alpha_6)^{q-1} \mod q$$

$$\equiv -1 \mod q.$$

So the c_2 of K_4 is the constant sequence -1!

In fact all zig-zag graphs Z_ℓ with $\ell \geq 3$, which includes $K_4 = Z_3$, have $c_2(Z_\ell) = -1$ (see Corollary 54 of [BS12]). This makes zig-zag graphs particularly special as they are an infinite family of graphs for which both the period (Theorem 2.1.6) and the c_2 are completely known!

(2.5.4) Hourglass chains.

After double triangle reduction, up until recently, there hasn't been much progress in the direction of new graph transforms giving families of graphs with the same c_2. A breakthrough came after

Schnetz [Sch19] introduced quadratic denominator reduction (see Section 2.4.3) as a new technique to compute the c_2-invariant. The potential of this new technique was illustrated in [SY21] where Schnetz and Yeats were able to determine a formula (using both cases in Definition 2.4.6) for the c_2 of special infinite families of graphs.

The interesting part is that these graphs, which are formed from attaching **hourglass chains** to a **kernel** graph, were able to be reduced to their kernels (which then may be further reduced). That is, the c_2 of such graphs only depends on said kernel, thus creating infinite families of graphs (of all loop orders) with the same c_2 and for which the c_2 is completely determined. Furthermore, for suitable kernels these hourglass chain graphs actually form infinite families of prime ancestors!

This is exciting as we have previously only seen families of graphs for which the c_2 is trivial or $c_2 = -1$ (like zig-zag graphs), or we have double-triangle families, some of which the c_2 is unknown. With hourglass chains, we now have families with a variety of interesting c_2s, including sequences that have never been found before!

(**2.5.5**) **Legendre symbols and modular forms.**

Finally we very briefly mention some of the (other) sequences that appear as c_2-invariants of ϕ^4 graphs. Of the identified sequences, these c_2s currently fall into two types. The first type corresponds to constant and quasi-constant c_2s, more generally those that are Legendre symbols. The second type are what we call modular, where the c_2 arises as the Fourier coefficients of certain modular forms. We refer the reader to [BS13; Sch19] and the references therein for complete details.

3 | COMPLETING THE c_2 COMPLETION CONJECTURE FOR $p = 2$

One of the motivating reasons to study the c_2-invariant (Definition 2.2.1) is its close relation to the Feynman period of a graph, as given in Equation (2.1.1), where these are primitive divergent graphs in ϕ^4-theory. Recall from Section 2.1 that these particular graphs can be described as connected 4-regular graphs with one vertex removed and with the property that $|E| = 2\ell$, where ℓ is the loop order, and $|E(\lambda)| > 2\ell(\lambda)$ for any non-empty proper subgraph λ.

Not only is the c_2-invariant essentially capturing the behaviour of the zeros of the denominators of the period, but the connection is further strengthened by Conjecture 2.2.2, which says that the c_2 should be a period invariant. For this conjecture to hold, we would need the c_2-invariant to have all the symmetries of the period as described in Section 2.1.3. In particular, it should be invariant under the period symmetry called completion.

Schnetz proved in [Sch10] that the period could be defined for the graphs created by adding back the vertex that was "removed" from primitive divergent graphs, and thus "completing" them to become 4-regular. We call this 4-regular graph G the **completion** and call the original primitive divergent graph $G - v$ for some vertex v a **decompletion** of G. Note that decompletions of 4-regular graphs could be non-isomorphic. To utilize this, Schnetz further proved that if two primitive divergent graphs have the same completion, then their periods are also the same (see Theorem 2.1.5). Thus the power of completion is that it allows us to lift the problem of calculating periods on primitive divergent graphs to one on 4-regular graphs, in the process cutting down the number of graphs to compute periods for.

For the c_2-invariant, whether or not the completion symmetry holds is still an open conjecture.

The c_2 Completion Conjecture — (Conjecture 4.35 of [BS12])
Let G be a connected 4-regular graph, and let v and w be vertices of G. Then,

$$c_2(G - v) = c_2(G - w).$$

While the c_2-invariant is defined for all prime powers q and more general connected graphs, we will restrict to when $q = p$ is a prime (see also Conjecture 2.5.1) and to decompletions of connected 4-regular graphs G. Note that for any vertex v, by Euler's formula, the decompletion $G - v$ indeed satisfies $|E(G - v)| = 2\ell$ where ℓ is the loop order of $G - v$.

Yeats in [Yea18b] first made progress in the special case of $p = 2$ by reducing the conjecture to a combinatorial counting problem; one involving enumerating certain edge bipartitions. Using two different constructions, Yeats was able to partially prove the $p = 2$ case. However, the obstruction to a full proof lay in the second construction, which required restricting the argument to completed graphs with an odd number of vertices.

In this chapter, inspired by the first construction in [Yea18b], we present this argument and show how by using the same ideas we can get rid of the mysterious parity condition on the vertices, thereby completing the c_2 completion conjecture for this special case.

3.1. THE CONJECTURE AND THE $p = 2$ CASE

Let G be a connected 4-regular graph, and let v and w be vertices of G. Before even approaching the conjecture, we notice that it suffices to prove the equation holds when v and w are adjacent. This is because as G is connected, there is a path between any two vertices, and successively applying the equation to each pair of neighbours along the path gives the more general result.

As we will see shortly, the neighbours of v and w play an important role, and thus we split the conjecture into cases based on the number of neighbours that v and w have in common. Since G is a 4-regular graph and v and w are adjacent, there are four possibilities: they share all three neighbours, or exactly two or one neighbour, or they have no neighbours in common. In the first case, when v and w share all neighbours, $G - v$ and $G - w$ are isomorphic, and thus the conjecture holds trivially. For the others, we will deal with each case separately and label them with T, S, and R, respectively. The T-case is when v and w share exactly two neighbours, the S-case is when they share only one neighbour, and the R-case is when they do not share any neighbours. The three cases are depicted in Figure 3.1 where the grey blobs are the graphs $G - \{v, w\}$ in each case, and will be addressed in Sections 3.2, 3.3, and 3.4, respectively.
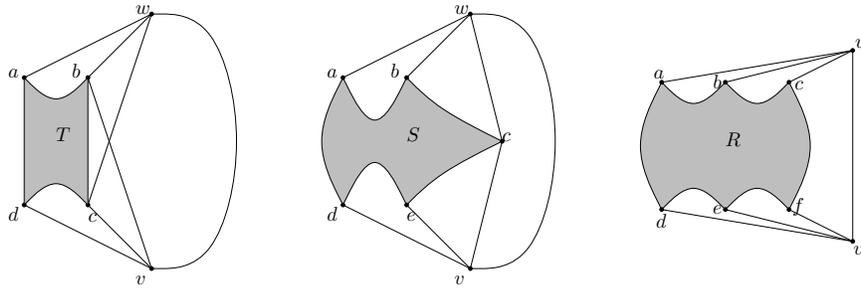
FIGURE 3.1: The graph G, and the three cases T, S, and R.

With the set-up complete, the first step in tackling the c_2 completion conjecture is the question of how to reduce it into an approachable combinatorial counting problem. Recall that in Section 2.4.2, we were able to transform the c_2-invariant for primes p into the following (2.4.1) for a graph $G - v$

$$c_2^{(p)}(G - v) \equiv -[\alpha_4^{p-1} \cdots \alpha_{|E|}^{p-1}] \left(\Psi^{13,23} \Psi_3^{1,2} \right)^{p-1} \mod p,$$

where $1, 2, 3$ are the labels of any three distinct edges of $G - v$, and the Dodgson polynomials in the equation are with respect to $G - v$.

Now suppose u is a 3-valent vertex in $G - v$. Let the edges $1, 2, 3$ be the three edges incident to u, and let u_1, u_2, u_3 be the respective vertices adjacent to these edges as below:
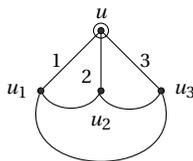
Looking at the term $\Psi_3^{1,2}$, as discussed in Section 2.3.2 we can view this Dodgson polynomial as coming from the following intersection of spanning trees
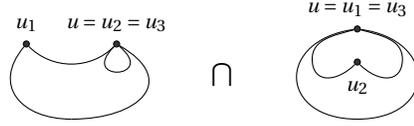
where the first blob depicts spanning trees on $(G-v) \backslash 1/\{2,3\}$, and the second blob is on $(G-v) \backslash 2/\{1,3\}$.

To translate this into spanning forests on $(G-v) \backslash \{1,2,3\}$ using Theorem 2.3.8, we just need to determine the partitions P of $\{u, u_1, u_2, u_3\}$ (as these are the vertices of edges $\{1,2,3\}$) such that all the spanning forests compatible with P are spanning trees in the above intersection. First, spanning trees on the first blob correspond to spanning forests such that u, u_2, and u_3 are in different trees, and u_1 can be in any of the three trees. Similarily, spanning trees on the second blob correspond to spanning forests such that u, u_1, and u_3 are in different trees, and there is no restriction for which tree u_2 is in. As these spanning forests have exactly three parts, the intersection of these blobs are spanning forests such that u is in one tree, u_3 is in another tree, and u_1 and u_2 are in the last tree. Furthermore, this is the only possible partition of $\{u, u_1, u_2, u_3\}$ that satisfies the required condition.

In particular, there is only one spanning forest polynomial in the sum, which is $\Phi_{(G-v) \backslash \{1,2,3\}}^{\{u\},\{u_3\},\{u_1,u_2\}}$. However, notice that u is an isolated vertex in $(G-v) \backslash \{1,2,3\}$ since edges $1,2,3$ are exactly the edges incident to u. This means we can remove the vertex u, giving the equation

$$\Psi_3^{1,2} = \Phi_{G-\{u,v\}}^{\{u_3\},\{u_1,u_2\}}.$$

Thus we are left with spanning forests with exactly two parts, which we call **spanning 2-forests**, such that they are compatible with the vertex partition $\{u_3\}, \{u_1, u_2\}$.

Following the same process for $\Psi^{13,23}$, in either case we want spanning trees on the graph $G - \{u, v\}$
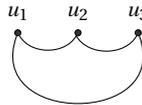
where for $(G-v) \backslash \{1,3\}/2$, we have $u = u_2$, and for $(G-v) \backslash \{2,3\}/1$, we have $u = u_1$. Once again, we can translate these to spanning forests on $(G-v) \backslash \{1,2,3\}$, this time with exactly two parts, giving the polynomial $\Phi_{(G-v) \backslash \{1,2,3\}}^{\{u\},\{u_1,u_2,u_3\}}$. Noticing that u is an isolated vertex in this graph, removing vertex u gives

$$\Psi^{13,23} = \Phi_{G-\{u,v\}}^{\{u_1,u_2,u_3\}} = \Psi_{G-\{u,v\}}.$$

Putting everything together, we have reformulated the c_2-invariant to

$$c_2^{(p)}(G - v) \equiv -[\alpha_4^{p-1} \cdots \alpha_{|E|}^{p-1}] \left(\Phi_{G-\{u,v\}}^{\{u_3\},\{u_1,u_2\}} \Psi_{G-\{u,v\}} \right)^{p-1} \mod p,$$

which we call **reducing with respect to vertex** u. Note that the ordering of the vertices u_1, u_2, and u_3 was completely arbitrary, and this formulation of the c_2 does not depend on this choice.

Finally, by working with the c_2 in this form, there's a particularly nice interpretation of what this equation is counting. From the coefficient extraction, the monomial we are looking for can be viewed as distributing $p-1$ copies of each edge in $G - \{u, v\}$ across $2p - 2$ polynomials, where now $G - \{u, v\}$

is the underlying graph for the polynomials. Of these $2p - 2$ polynomials, $p - 1$ of them arise from spanning 2-forests compatible with the vertex partition $\{u_3\}, \{u_1, u_2\}$ and the other $p - 1$ come from spanning trees. Then, since all the monomials in Ψ and Φ appear with a coefficient of 1, we are enumerating all the ways to do such a distribution.

Now, recalling that the terms in Ψ and Φ correspond to the edges *not* in the respective trees or forests, we need to shift our view slightly to make sense of what this coefficient is counting. Given $p - 1$ spanning trees of $G - \{u, v\}$, the corresponding monomial in $\left(\Psi_{G-\{u,v\}}\right)^{p-1}$ comes from combining all the edges of the graph not in each spanning tree. Thus the edges left over to be distributed to the $p - 1$ Φ polynomials are exactly the edges of the $p - 1$ spanning trees. Furthermore, these edges must be grouped in a such a way that removing them gives $p - 1$ spanning 2-forests.

Similarily, given $p - 1$ spanning 2-forests the corresponding monomial in $\left(\Phi_{G-\{u,v\}}^{\{u_3\},\{u_1,u_2\}}\right)^{p-1}$ combines all the edges of the graph not in each forest. Notice this monomial must then match the edges left over from the monomial given by the $p - 1$ spanning trees above, which are exactly the edges of the spanning trees. By the same logic, the monomial arising from the $p - 1$ spanning trees must then match the edges of the $p - 1$ spanning 2-forests. Thus, we can swap the roles of the spanning trees and forests, and equivalently express the enumeration as counting the number of ways to partition $p - 1$ copies of each edge into $p - 1$ spanning trees and $p - 1$ spanning 2-forests compatible with $\{u_3\}, \{u_1, u_2\}$!

When $p = 2$, this interpretation is much simpler as the equation simplifies to

$$c_2^{(2)}(G - v) \equiv [\alpha_4 \cdots \alpha_{|E|}] \, \Phi_{G-\{u,v\}}^{\{u_3\},\{u_1,u_2\}} \Psi_{G-\{u,v\}} \mod 2. \tag{3.1.1}$$

Now we only have one copy of each edge, and we are counting the number of edge bipartitions where one part is a spanning tree and the other is a spanning 2-forest compatible with $\{u_3\}, \{u_1, u_2\}$. We will usually denote these edge partitions as (ψ, ϕ). Additionally, as we are counting modulo 2, we really only care about the parity of this count. Thus one strategy to tackling the c_2 completion conjecture when $p = 2$ is to find fixed-point free involutions on the appropriate sets of edge bipartitions!

3.2. THE T-CASE

The first case we will look at is the T-case, which is also special in and of itself. It is the starting point for many of the arguments needed later on. Let G be a connected 4-regular graph, and let v and w be two adjacent vertices of G such that they share two common neighbours. Let $T = G - \{v, w\}$ be the graph obtained by removing vertices v and w from G, the first grey blob in Figure 3.1, with the neighbours of v and w labelled $\{a, b, c, d\}$ as in the figure. Here w has neighbours $\{a, b, c, v\}$, and v has neighbours $\{b, c, d, w\}$.

What's special about this case is when v and w share two common neighbours, these four vertices form a double triangle as defined in Section 2.5.2. From the results in Section 2.5.2, we know that double triangles are important since the c_2 is invariant under double triangle reductions. Now adding in decompleting at a double triangle vertex, as proved in [Hu+18] (see also Theorem 2.5.2), we can resolve the c_2 completion conjecture in the T-case for all values of p as a consequence of double triangle reduction. This proof uses a different argument, namely via proving an equivalence of certain graph polynomials. Even though we do have this alternative proof, it is still worthwhile to present the

enumerative argument from [Yea18b] in this section, as it showcases the proof techniques and overall framework that the S and R cases build upon.

Furthermore, another reason for the specialness of the T-case is that by using the same counting methods as for the $p = 2$ case, these results can be extended to all values of p [Yea18a]. While the S and R cases are not as simple, this extension gives some hope for generalizing the arguments needed in those cases for higher values of p.

(**3.2.1**) **Set-up.**

To start off, we define the particular sets of edge partitions that we are counting using the vertex bipartition that the spanning 2-forests are compatible with (see Definition 2.3.6). We will use similar definitions for the S and R cases.

Definition 3.2.1 — (Definition 3.1 of [Yea18b]) *Suppose P is a bipartition of $\{a, b, c, d\}$. Let $\mathcal{T}_P$ be the set of bipartitions (ψ, ϕ) of the edges of T such that ψ is a spanning tree and ϕ is a spanning 2-forest compatible with P. Let $t_P = |\mathcal{T}_P|$.*

Now, since we can choose the order of the edges adjacent to the 3-valent vertex we are reducing by in Equation (3.1.1), and thus choose the partition of its neighbours into a 1-part and 2-part for the 2-forest, in the T-case we can do so in such a way that exploits the property that v and w share two common neighbours.

Proposition 3.2.2 — (Proposition 3.2, 3.3 of [Yea18b]) *When v and w have common neighbours b, c,*

$$c_2^{(2)}(G - v) = t_{\{a\},\{b,c\}} \mod 2,$$

$$c_2^{(2)}(G - w) = t_{\{d\},\{b,c\}} \mod 2,$$

and thus we have

$$c_2^{(2)}(G - v) - c_2^{(2)}(G - w) = t_{\{a\},\{b,c,d\}} + t_{\{d\},\{a,b,c\}} \mod 2.$$

PROOF. In the $G - v$ case, from Equation (3.1.1) we can reduce with respect to vertex w, since now w is a 3-valent vertex, and choose the order of its adjacent edges such that we get vertex partition $\{a\}, \{b, c\}$ for the spanning 2-forest. Then, by the discussion in Section 3.1, this is equivalent to counting the number of ways to partition the edges of $G - \{v, w\} = T$ into a spanning tree and a spanning 2-forest compatible with $\{a\}, \{b, c\}$, which gives

$$c_2^{(2)}(G - v) \equiv [\alpha_4 \cdots \alpha_{|E(G-v)|}] \, \Phi_{G-\{v,w\}}^{\{a\},\{b,c\}} \Psi_{G-\{v,w\}} \equiv t_{\{a\},\{b,c\}} \mod 2.$$

Similarly, in the $G - w$ case, v is now a 3-valent vertex, and we choose vertex partition $\{d\}, \{b, c\}$ to get

$$c_2^{(2)}(G - w) \equiv t_{\{d\},\{b,c\}} \mod 2.$$

Finally, enumerating over all possibilities for the vertex set $\{a, b, c, d\}$ in the spanning 2-forests

$$t_{\{a\},\{b,c\}} = t_{\{a\},\{b,c,d\}} + t_{\{a,d\},\{b,c\}},$$

$$t_{\{d\},\{b,c\}} = t_{\{d\},\{a,b,c\}} + t_{\{a,d\},\{b,c\}},$$

37

and thus since $t_{\{a,d\},\{b,c\}}$ appears in both equations and we are working modulo 2

$$c_2^{(2)}(G-v) - c_2^{(2)}(G-w) = t_{\{a\},\{b,c\}} - t_{\{d\},\{b,c\}} \equiv t_{\{a\},\{b,c,d\}} + t_{\{d\},\{a,b,c\}} \quad \text{mod } 2.$$

∎

(**3.2.2**) **Swapping around a two-valent vertex.**

The main property of the T-case, that makes it much simplier to deal with, is that since v and w share two common neighbours, after removing both v and w we are left with two 2-valent vertices. As we are dealing with edge bipartitions involving spanning trees and forests of a particular form, these 2-valent vertices cannot be isolated in said forests, and thus we know exactly how the two edges of a 2-valent vertex must be distributed across the edge partition. Furthermore, we can swap these edges to obtain another valid edge partition.

Lemma 3.2.3 — (Lemma 4.1 of [Yea18b]) *Let V be a set of marked vertices of a connected graph G, and suppose $c \in V$ is a 2-valent vertex. Let (ψ, ϕ) be an edge partition of G where ψ is a spanning tree and ϕ is a spanning 2-forest that is compatible with a bipartition of V where each part (and thus forest) contains at least one of the vertices in $V \setminus \{c\}$.*

Then, of the the two edges incident to c, exactly one edge is in each part of (ψ, ϕ), and swapping which edge is in which part yields a new edge partition (ψ', ϕ') with the same properties, except possibly the bipartition of V that ϕ' is compatible with.

PROOF. Looking at the edge partition (ψ, ϕ), first, since ψ is a spanning tree of G there must be at least one edge in ψ that is adjacent to the vertex c. Now since ϕ is a spanning 2-forest in which each forest contains at least one of the vertices in $V \setminus \{c\}$, we have that in particular, c is not an isolated vertex in ϕ and must be connected to the rest of its forest via at least one edge. As c is a 2-valent vertex adjacent to at least one edge in part of the edge partition, exactly one edge incident to c must be in each of (ψ, ϕ).

Because c is a leaf in both ψ and ϕ, removing the edges incident to c disconnects and furthermore, isolates c from the rest of the tree and the 2-forest, respectively. Since both ψ and ϕ are spanning, they must contain both neighbours of c, and thus we can reconnect c via the opposite edges incident to c as in ψ and ϕ. This gives a new edge partition (ψ', ϕ') where ψ' is a spanning tree and ϕ' is a spanning 2-forest, the only difference between the two edge partitions being the swapping of the edges incident to c. However, in ϕ', c could be reconnected to a different tree than in ϕ thus changing the bipartition of V that ϕ' is compatible with. Specifically, either the bipartition of V remains the same as for ϕ or c swaps between the parts of the bipartition. ∎

With this observation that we can swap the edges around any 2-valent vertex to get two different edge partitions (ψ, ϕ) and (ψ', ϕ'), we can obtain fixed-point free involutions on $\mathcal{T}_P$ for some particular bipartitions P of $\{a, b, c, d\}$.

Theorem 3.2.4 — (from §6 of [Yea18b]) *There is a fixed-point free involution on*

$$\mathcal{T}_{\{a\},\{b,c,d\}} \cup \mathcal{T}_{\{a,b\},\{c,d\}},$$

and on

$$\mathcal{T}_{\{d\},\{a,b,c,\}} \cup \mathcal{T}_{\{a,b\},\{c,d\}}.$$

Thus we have

$$t_{\{a\},\{b,c,d\}} + t_{\{a,b\},\{c,d\}} \equiv 0 \quad \mathrm{mod}\, 2,$$
$$t_{\{d\},\{a,b,c\}} + t_{\{a,b\},\{c,d\}} \equiv 0 \quad \mathrm{mod}\, 2,$$

which together gives

$$t_{\{a\},\{b,c,d\}} + t_{\{d\},\{a,b,c\}} \equiv 0 \quad \mathrm{mod}\, 2.$$

PROOF. Consider the edge partitions in $\mathcal{T}_{\{a\},\{b,c,d\}} \cup \mathcal{T}_{\{a,b\},\{c,d\}}$. Since b is a 2-valent vertex in T, and both $\{a\},\{b,c,d\}$ and $\{a,b\},\{c,d\}$ are bipartitions of $\{a,b,c,d\}$ in which each part contains at least one of $\{a,c,d\}$, these edge partitions satisfy the conditions of Lemma 3.2.3.

Take any edge partition $(\psi, \phi) \in \mathcal{T}_{\{a\},\{b,c,d\}}$ and swap the edges incident to b between the two parts to get a new edge partition (ψ', ϕ'). Looking at the new spanning 2-forest ϕ', either both neighbours of b were in the same tree in ϕ, and thus b was reconnected to that same tree corresponding to the part $\{b,c,d\}$ in the vertex bipartition. Or the neighbours of b were in different trees in ϕ, and thus b was reconnected but to the tree coresponding to the part $\{a\}$. In the first case, (ψ', ϕ') is once again an edge partition in $\mathcal{T}_{\{a\},\{b,c,d\}}$. In the second case, (ψ', ϕ') in now an edge partition in $\mathcal{T}_{\{a,b\},\{c,d\}}$. Similarly, for edge partitions in $\mathcal{T}_{\{a,b\},\{c,d\}}$, swapping around b either gives a new edge partition in the same set (first case) or in $\mathcal{T}_{\{a\},\{b,c,d\}}$ (second case).

In either case, since we are swapping the edges incident to b in each part, (ψ', ϕ') must be different from (ψ, ϕ), and swapping around b once again returns us to (ψ, ϕ). Thus, this action of swapping around b gives a fixed-point free involution on $\mathcal{T}_{\{a\},\{b,c,d\}} \cup \mathcal{T}_{\{a,b\},\{c,d\}}$, and so the size of this set must be even, giving the equation

$$t_{\{a\},\{b,c,d\}} + t_{\{a,b\},\{c,d\}} \equiv 0 \quad \mathrm{mod}\, 2.$$

Now since c is also 2-valent in T, using the exact same argument on $\mathcal{T}_{\{d\},\{a,b,c\}} \cup \mathcal{T}_{\{a,b\},\{c,d\}}$ with c as the vertex we are swapping around, we obtain a fixed-point free involution on these edge partitions, and thus

$$t_{\{d\},\{a,b,c\}} + t_{\{a,b\},\{c,d\}} \equiv 0 \quad \mathrm{mod}\, 2.$$

Finally, notice that $t_{\{a,b\},\{c,d\}}$ appears in both equations, and since we are working modulo 2, adding the two equations gives

$$t_{\{a\},\{b,c,d\}} + 2\,t_{\{a,b\},\{c,d\}} + t_{\{d\},\{a,b,c\}} \equiv t_{\{a\},\{b,c,d\}} + t_{\{d\},\{a,b,c\}} \equiv 0 \quad \mathrm{mod}\, 2.$$

$\blacksquare$

From Proposition 3.2.2, the final equation in Theorem 3.2.4 is exactly what we need to prove the c_2 completion conjecture for $p = 2$ in the T-case.

Corollary 3.2.5 — *Let G be a connected 4-regular graph. Let v and w be adjacent vertices of G such that they share two common neighbours. Then,*

$$c_2^{(2)}(G - v) = c_2^{(2)}(G - w).$$

3.3. THE S-CASE

With the T-case complete, we move to the slightly more complicated S-case. Let G be a connected 4-regular graph, and let v and w be two adjacent vertices of G such that they share one neighbour. Let $S = G - \{v, w\}$ be the graph obtained by removing vertices v and w from G, the middle grey blob in Figure 3.1, with the neighbours of v and w labelled $\{a, b, c, d, e\}$ as in the figure. Here w has neighbours $\{a, b, c, v\}$, and v has neighbours $\{c, d, e, w\}$.

While the S-case is more complicated, it does use ideas that stem from the T-case. Firstly, as v and w still share a neighbour, after removing both vertices S will have a 2-valent vertex, and thus we can use the same swapping around a 2-valent vertex argument as in the T-case. In fact, originally in [Yea18b] this argument was first used for the S-case! However, it was not enough to complete the proof of the conjecture for $p = 2$. A different type of argument involving compatible cycles (see §5 of [Yea18b]) was given for the rest of the sets not covered by swapping around a 2-valent vertex, but under the condition that G had an odd number of vertices.

Secondly, this idea of swapping around a 2-valent vertex can be generalized to swapping around a particular vertex called the control vertex, under certain conditions. While first used in [Yea18b] for the R-case, we prove that this argument can be modified to apply in the S-case as well. This modification was able to cover all the sets from the compatible cycle argument without the odd vertex condition on G, and thus complete the proof of the c_2 completion conjecture for $p = 2$ in the S-case.

Being not as simple as the T-case but also perhaps not as hard as the R-case due to the presence of a 2-valent vertex, the S-case seems to be the perfect playground and stepping stone for trying to further extend these ideas for higher values of p.

To give a complete, self-contained proof of the S-case, we begin by presenting the set-up and some of the results from [Yea18b] with proof, and in Section 3.3.3 we introduce our new results which allows us to finish the S-case for $p = 2$!

(**3.3.1**) **Set-up.**

As with the T-case, we start with defining the particular sets of edge partitions that we are counting.

Definition 3.3.1 — (Definition 3.1 of [Yea18b]) *Suppose P is a bipartition of $\{a, b, c, d, e\}$. Let $\mathscr{S}_P$ be the set of bipartitions (ψ, ϕ) of the edges of S such that ψ is a spanning tree and ϕ is a spanning 2-forest compatible with P. Let $s_P = |\mathscr{S}_P|$.*

Once again, we can pick the vertex bipartitions for the spanning 2-forests to exploit the property that v and w share a neighbour.

Proposition 3.3.2 — (Proposition 3.2, 3.3 of [Yea18b]) *When v and w have one common neighbour c*

$$c_2^{(2)}(G - v) = s_{\{c\}, \{a, b\}} \quad \mod 2,$$

$$c_2^{(2)}(G - w) = s_{\{c\}, \{d, e\}} \quad \mod 2,$$

and thus we have

$$c_2^{(2)}(G - v) - c_2^{(2)}(G - w) = s_{\{c,d\},\{a,b,e\}} + s_{\{c,e\},\{a,b,d\}} + s_{\{a,b\},\{c,d,e\}} \tag{3.3.1}$$
$$+ s_{\{a,c\},\{b,d,e\}} + s_{\{b,c\},\{a,d,e\}} + s_{\{d,e\},\{a,b,c\}} \quad \text{mod } 2.$$

PROOF. Using the same argument as in the proof of Proposition 3.2.2, we can pick the vertex partition $\{c\}, \{a, b\}$ for the $G - v$ case and $\{c\}, \{d, e\}$ for the $G - w$ case to equivalently determine each c_2 via counting edge partitions into a spanning tree and a spanning 2-forest compatible with the respective vertex partition.

Then, enumerating over all possibilities for the vertex set $\{a, b, c, d, e\}$ in the spanning 2-forests, and noticing that $s_{\{c\},\{a,b,d,e\}}$ appears for both $G - v$ and $G - w$, we obtain the final equation by subtracting the expanded counts for $G - v$ and $G - w$, and simplifying modulo 2. ∎

Here is a nice visual representation of Equation (3.3.1) from [Yea18b] where we depict the vertex bipartition for the spanning 2-forest using different shapes:

(**3.3.2**) **Swapping around c.**

The main property of the S-case is that v and w share a neighbour, and in particular, after removing both v and w from G, we are left with a 2-valent vertex. Since we already know how to deal with 2-valent vertices, we immediately obtain the following theorem.

Theorem 3.3.3 — (Lemma 4.2 of [Yea18b]) *There is a fixed-point free involution on*

$$\mathscr{S}_{\{a,b,c\},\{d,e\}} \cup \mathscr{S}_{\{a,b\},\{c,d,e\}},$$

and thus we have

$$s_{\{a,b,c\},\{d,e\}} + s_{\{a,b\},\{c,d,e\}} \equiv 0 \quad \text{mod } 2.$$

PROOF. Consider the edge partitions in $\mathscr{S}_{\{a,b,c\},\{d,e\}} \cup \mathscr{S}_{\{a,b\},\{c,d,e\}}$. Since c is a 2-valent vertex in S, and both $\{a, b, c\}, \{d, e\}$ and $\{a, b\}, \{c, d, e\}$ are bipartitions of $\{a, b, c, d, e\}$ in which each part contains at least one of $\{a, b, d, e\}$, these edge partitions satisfy the conditions of Lemma 3.2.3.

Then, using the same argument as in the proof of Theorem 3.2.4, swapping the edges incident to c between the two parts of the edge partitions of interest gives a new edge partition with a spanning 2-forest either compatible with the same vertex bipartition as we started with or the vertex bipartition where c swaps between the parts of the original vertex bipartition. Once again we have a fixed-point free involution, this time on $\mathscr{S}_{\{a,b,c\},\{d,e\}} \cup \mathscr{S}_{\{a,b\},\{c,d,e\}}$, and so the size of this set must also be even

$$s_{\{a,b,c\},\{d,e\}} + s_{\{a,b\},\{c,d,e\}} \equiv 0 \quad \text{mod } 2.$$

∎

(a) x is in the 2-part after removing v (b) $x = v$ or x is in one of the singletons after removing v
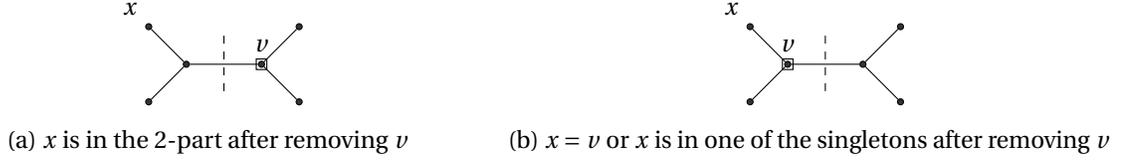
FIGURE 3.2: The two possible control vertices v depending on the specification for special vertex x for Lemma 3.3.4. The three unlabelled leaves are the vertices in $p \setminus \{x\}$. The dashed lines indicate the edges we will be picking for one of the involutions in the S-case and in the R-case, respectively. Some of the leaf edges are possibly contracted, with the exception that two leaf edges incident to the same vertex cannot simultaneously both be contracted.

Swapping around c takes care of two specific vertex bipartitions on the right-hand side of Equation (3.3.1), and what we are left with are sets of the form $\mathscr{S}_{\{c,*\},\{*,*,*\}}$. However, when we try to swap around c for these sets we obtain vertex bipartitions not in the equation nor ones that coincide with each other, like in the T-case with $t_{\{a,b\},\{c,d\}}$. For example, for $\{a,c\},\{b,d,e\}$, we would get the equation $s_{\{a,c\},\{b,d,e\}} + s_{\{a\},\{b,c,d,e\}} \equiv 0 \mod 2$ and similarly for the rest. To tackle this problem, we generalize the idea of swapping around a 2-valent vertex!

(**3.3.3**) **Swapping around a control vertex.**

For the remaining sets in Equation (3.3.1), we generalize the swapping argument to one using a vertex that is not 2-valent in S. While this idea of using control vertices is not new, we adapt it to be able to apply it to the S-case. In the process we develop a new swapping method involving multiple vertices, starting with using a control vertex in conjunction with a 2-valent vertex. Generalizing once again, the notion of using multiple control vertices will then be used for the R-case.

Recall that in the 2-valent vertex case, we are swapping the edges incident to this vertex between the two parts of the edge partition (ψ, ϕ). We can do so uniquely since under some requirements for (ψ, ϕ), this vertex is a leaf in both parts.

As we no longer have an appropriate 2-valent vertex, we need to define a different vertex to swap edges around. We do so under certain conditions such that the same vertex can be recovered after swapping an incident edge in both ψ and ϕ, and such that we can get a new 2-forest compatible with a vertex bipartition that we care about!

Lemma 3.3.4 — (generalizing Lemma 4.3 of [Yea18b]) *Let G be a connected 4-regular graph with two adjacent vertices removed, and let V be the set of neighbours of those vertices. Fix a vertex $x \in V$, which we call the **special vertex**. Suppose we have a bipartition P of V such that each part contains at least one of the vertices in V and one part is of the form $p = \{x, *, *, *\}$.*

Let (ψ, ϕ) be an edge partition of G where ψ is a spanning tree and ϕ is a spanning 2-forest compatible with P. Let t be the tree corresponding to p in the 2-forest ϕ. Then, there are two distinct vertices v such that removing v from t partitions $\{x, *, *, *\}$ into a 2-part, a singleton, and possibly a third part which is also a singleton.

If we further specify that x must satisfy one of the following properties, either

(i) *x is in the 2-part, or*
(ii) *x = v or x is in one of the singleton components,*

*then there is a unique vertex v, which we call the **control vertex**, that satisfies all the required properties. Figure 3.2 depicts the control vertex in each case. Furthermore, in t this control vertex v will either be 2-valent with v ∈ p or 3-valent and v ∉ p. In both cases v is a leaf in the spanning tree ψ.*

PROOF. Looking at the tree t and the subtree created by taking the union of all the paths in t between any two vertices in p, since p has four vertices this subtree will be of the form
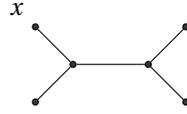
in which the leaves are vertices in p, the edges are paths in t, and some paths are possibly contracted.

First, we argue that this is indeed the correct form for the subtree. Since ψ is a spanning tree and we are partitioning the edges of G, the vertices in ϕ, and thus the non-leaves in the subtree, are at most 3-valent. Furthermore, in ϕ the vertices in P, and thus in p, are at most 2-valent. This immediately implies that we have the subtree form above, where some of the paths may be contracted. The valency constraints also mean that the path between the two non-leaf vertices cannot be contracted. Note trivially, the paths between two leaves adjacent to the same closest non-leaf vertex cannot both be contracted simultaneously. All possible configurations are drawn in Figure 6 of [Yea18b].

Now, because of the restrictions on the subtree configurations, removing any one of the two non-leaf vertices in the subtree above from t creates components that partitions p into a 2-part, a singleton, and possibly a third singleton. By the valency of the vertices in ϕ, these two vertices can only be 2-valent if they are in p or otherwise 3-valent. In either case, by the properties of G, they will be leaves in ψ.

Let v be one of the two non-leaf vertices in the subtree above. Finally, to get uniqueness of v we need to fix where one vertex $x \in p$ must be after removing v from t. In the first case, fixing x to be in the 2-part component, we get the control vertex v as pictured in Figure 3.2a. Notice here x could never be the control vertex v. In the second case, if we want x to be in a singleton component we choose the other non-leaf vertex to be v as in Figure 3.2b. However, with this choice of v, we could also have that $x = v$ (the path between x and v is contracted in t). Thus in this case, the property we are fixing is either $x = v$ or x is in one of the singleton components. ∎

Now we are ready to tackle the remaining sets in the Equation (3.3.1), which all are of the form $\mathscr{S}_{\{c,*\},\{*,*,*\}}$. Instead of dealing with these sets directly, since we know there is a fixed-point free involution via swapping around c (as it is a 2-valent vertex), we can equivalently show that there is a fixed-point free inovlution on sets of the form $\mathscr{S}_{\{*\},\{c,*,*,*\}}$ (the "swapped" version).

The involution uses a two-phase process, where we utilize that we can swap around c for certain edge bipartitions. For the others, we modify the involution used in Theorem 3.4.3, which uses the same idea of swapping edges incident to the control vertex between the two parts of (ψ, ϕ).

Theorem 3.3.5 — *There is a fixed-point free involution on*

$$\mathscr{S}_{\{a\},\{b,c,d,e\}} \cup \mathscr{S}_{\{b\},\{a,c,d,e\}} \cup \mathscr{S}_{\{d\},\{a,b,c,e\}} \cup \mathscr{S}_{\{e\},\{a,b,c,d\}},$$

and thus we have

$$s_{\{a\},\{b,c,d,e\}} + s_{\{b\},\{a,c,d,e\}} + s_{\{d\},\{a,b,c,e\}} + s_{\{e\},\{a,b,c,d\}} \equiv 0 \quad \mod 2.$$

PROOF. Consider the edge partitions in the union of sets of the form $\mathscr{S}_{\{*\},\{c,*,*,*\}}$ where the $*$'s are the vertices $\{a, b, d, e\}$ in any order. Since partitioning $\{a, b, c, d, e\}$ (the neighbours of the two adjacent vertices removed to obtain the graph S) into the form $\{*\}, \{c, *, *, *\}$ leaves each part containing at least one of $\{a, b, c, d, e\}$ and a 4-part containing c, these edge partitions satisfy the conditions of both Lemma 3.2.3 and Lemma 3.3.4. For Lemma 3.3.4, we are taking c to be the special vertex and specifying that c be in the 2-part after removing the control vertex from the approriate tree in the spanning 2-forest (the first property).

Let (ψ, ϕ) be an edge partition in any set of the form $\mathscr{S}_{\{x\},\{c,*,*,*\}}$, and let v be its control vertex. Let t be the tree in ϕ corresponding to part $\{c, *, *, *\}$ which also contains v. Since v is a leaf in the spanning tree ψ, there is exactly one edge incident to v in ψ, which we will call η_v, and let n_v be the neighbour of v in ψ. Since c is a 2-valent vertex, and thus also a leaf in ψ, let η_c be the edge incident to c in ψ and n_c its neighbour.

Now to describe the involution giving a new edge partition:

(1) *Swapping c stays in* – If $n_c \in t$, then swap the edges incident to c between ψ and ϕ.
(2) *Swapping c goes out* – Otherwise:
  (i) *Swapping v stays in* – If $n_v \in t$, so the control vertex and its neighbour in ψ are in the same tree of ϕ, let η be the edge incident to v in t in the path to n_v.
  Then, swap the edges η_v and η between ψ and ϕ.
  (ii) *Swapping v goes out* – If $n_v \notin t$, so the control vertex and its neighbour in ψ are in different trees of ϕ, let η be the edge incident to v in t in the path to c; in Figure 3.2a this is the edge towards the 2-part indicated by the dashed line.
  Then swap the edges η_v and η between ψ and ϕ, and finally swap the edges incident to c.

First, we show that we get a valid edge partition in a set of the form $\mathscr{S}_{\{*\},\{c,*,*,*\}}$. For (1), this is just a specific case of the swapping around a 2-valent vertex argument. As in Lemma 3.2.3, since c is a 2-valent vertex and is a leaf in both ψ and ϕ, we can swap the edges incident to c between the two parts to get (ψ', ϕ') where ψ' is still a spanning tree and ϕ' is still a spanning 2-forest. Looking at ϕ', as $n_c \in t$ and thus both neighbours of c were in the same tree in ϕ, c gets reconnected to t after the swap. Thus ϕ' is compatible with the same vertex bipartition as ϕ, and (ψ', ϕ') is in the same set $\mathscr{S}_{\{x\},\{c,*,*,*\}}$ as before.

For (2), we can use a similar argument to swapping around a 2-valent vertex, except replacing the 2-valent vertex with the control vertex v. From Lemma 3.3.4, we know that v is the unique vertex such that removing v from t partitions the vertices in part $\{c, *, *, *\}$ into a 2-part containing c, a singleton, and possibly a third singleton. Unlike the 2-valent argument, as v is no longer a leaf in t and we now have a choice of which edge incident to v in t we are swapping, we need to be careful of creating cycles when reconnecting v to the rest of t and make sure that we can recover the edge swapped and the same control vertex to indeed get an involution.

To deal with the possibility of creating cycles, we notice that the only way to get a cycle after swapping edges incident to v between ψ and ϕ is if the neighbour of v in ψ was in the same tree as v in ϕ and the edge picked to be swapped in ϕ was not on the cycle created by adding the edge incident to v in ψ to ϕ. Thus for (2i), if $n_v \in t$, then we pick the edge η to be the edge incident to v on the path to n_v in t. In this case all the neighbours of v are in the same tree in ϕ. Removing η from t breaks the tree t into two components, one with v and one containing n_v, splitting ϕ into three components. Adding η_v reconnects the two components from t via v and n_v to create a new spanning 2-forest ϕ'. As n_v and v were both in t, ϕ' is compatible with the same vertex bipartition as ϕ. Since v is a leaf in ψ, removing η_v from ψ and adding η to create ψ' maintains the spanning tree structure. Thus swapping the edges η_v and η between ψ and ϕ creates a new edge partition (ψ', ϕ') in the same set $\mathscr{S}_{\{x\},\{c,*,*,*\}}$ as we started with.

For (2ii) when $n_v \notin t$, we pick η to be the edge incident to v on the path to c in t. Removing η from t splits the tree into a component with v and a component which contains c and another vertex in part $\{c, *, *, *\}$, call it y, breaking ϕ into three components. Since $n_v \notin t$, adding η_v connects the component with v to the other tree in ϕ (corresponding to part $\{x\}$) creating a new spanning 2-forest ψ'. Once again as v is a leaf in ψ, removing η_v from ψ and adding η to create ψ' maintains the spanning tree structure. Thus swapping the edges η_v and η between ψ and ϕ creates a new edge partition (ψ', ϕ'), However, now ψ' is compatible with the vertex bipartition $\{c, y\}, \{x, *, *\}$. To get back to a vertex bipartition of the form $\{*\}, \{c, *, *, *\}$, we notice that we are in the case where $n_c \notin t$, and in particular, n_c was in the other tree of ϕ which is now connected to x. Since c can never be the control vertex, swapping the edges incident to c between ψ' and ϕ' gives a new edge partition (ψ'', ϕ'') now in the set $\mathscr{S}_{\{y\},\{c,x,*,*\}}$, which is of the correct form. Note that $y \neq x$ could be any of $\{a, b, d, e\}$.

In all three cases, as we are always changing the edges incident to c and/or v in ψ and ϕ, the new edge partitions can never be identical to (ψ, ϕ). Thus this transformation is fixed-point free.

Lastly, we prove that this transformation is indeed an involution by showing that the control vertex and the two edges being swapped remains the same in the new edge partition. Let (ψ', ϕ') be the transformed edge partition. For (1), since c is a 2-valent vertex, there is no choice of which edge is to be swapped and as both n_c and c are in the same tree in ϕ', we are in the same case of the transformation. Thus applying the transformation again returns us to (ψ, ϕ).

For (2), it will be easier to first formulate the control vertex in another way. We can define the control vertex v to be the first common vertex in t in the paths from exactly two of the vertices in part $\{c, *, *, *\} \setminus \{c\}$ to c. Under this formulation, we immediately see that any reconnecting of v via η_v does not change this property, and thus the control vertex remains the same after the swapping of edges incident to v. Additionally for (2ii), the swapping of edges incident to c afterwards also does not affect this property of v since n_c was in the tree connected to x, and thus once again the control vertex remains the same in ψ'.

For the edges, notice that (2i) occurs when all neighbours of v are in the same tree of ϕ, and thus by the choice of η, all the neighbours of v are still in the same tree of ϕ'. Additionally in ψ', η_v is exactly the edge incident to v on the cycle created by adding η, which is now the edge incident to v in ψ'. For (2ii), since we are reconnecting v to the other tree of ϕ via η_v, the new neighbour of v in ψ' is now in a different tree than v in ϕ'. Additionally in ψ', because n_c was in the tree connected to x in ϕ, after the two swaps η is exactly the edge incident to v on the path to c. In both cases (ψ', ϕ') remain in the same case of (2) as before with $\eta'_v = \eta$, as v is a leaf in ψ' and so there is only one choice of edge, and $\eta' = \eta_v$ as shown above. Thus, applying the transformation again returns us to (ψ, ϕ).

Putting everything together, we indeed have a fixed-point free involution in all cases. Finally, this holds for any edge partition in sets of the form $\mathscr{S}_{\{x\},\{c,*,*,*\}}$ where we are either staying in the same set $\mathscr{S}_{\{x\},\{c,*,*,*\}}$ or swapping to a different set $\mathscr{S}_{\{y\},\{c,*,*,*\}}$ where $y \neq x$ could be any of $\{a, b, d, e\}$. Thus our involution is on the union of sets of the form $\mathscr{S}_{\{*\},\{c,*,*,*\}}$,

$$\mathscr{S}_{\{a\},\{b,c,d,e\}} \cup \mathscr{S}_{\{b\},\{a,c,d,e\}} \cup \mathscr{S}_{\{d\},\{a,b,c,e\}} \cup \mathscr{S}_{\{e\},\{a,b,c,d\}},$$

and therefore, the size of this set must be even, giving the equation

$$s_{\{a\},\{b,c,d,e\}} + s_{\{b\},\{a,c,d,e\}} + s_{\{d\},\{a,b,c,e\}} + s_{\{e\},\{a,b,c,d\}} \equiv 0 \mod 2.$$

∎

As in our discussion before Theorem 3.3.5, since swapping around c gives fixed-point free involutions $\mathscr{S}_{\{x\},\{c,*,*,*\}} \cup \mathscr{S}_{\{c,x\},\{*,*,*\}}$ for $x \in \{a, b, d, e\}$, we immediately get

$$s_{\{a,c\},\{b,d,e\}} + s_{\{b,c\},\{a,d,e\}} + s_{\{c,d\},\{a,b,e\}} + s_{\{c,e\},\{a,b,d\}} \equiv 0 \mod 2$$

by adding all the equations together and using the result from Theorem 3.3.5.

Alternatively, notice that we could use the same fixed-point free involution on the union of sets of the form $\mathscr{S}_{\{c,*\},\{*,*,*\}}$ except with a slight modification. In (2) where the two neighbours of c are in different trees in the spanning 2-forest, we first swap around the edges of c, giving a new spanning 2-forest compatible with $\{*\}, \{c, *, *, *\}$, and proceed as before with finding the control vertex and edges to swap. Instead of swapping around c again in (2ii), we do so in (2i) to ensure we get edge partitions in sets of the form $\mathscr{S}_{\{c,*\},\{*,*,*\}}$. As everything else remains the same, we obtain the following result.

Corollary 3.3.6 — *There is a fixed-point free involution on*

$$\mathscr{S}_{\{a,c\},\{b,d,e\}} \cup \mathscr{S}_{\{b,c\},\{a,d,e\}} \cup \mathscr{S}_{\{c,d\},\{a,b,e\}} \cup \mathscr{S}_{\{c,e\},\{a,b,d\}},$$

and thus we have

$$s_{\{a,c\},\{b,d,e\}} + s_{\{b,c\},\{a,d,e\}} + s_{\{c,d\},\{a,b,e\}} + s_{\{c,e\},\{a,b,d\}} \equiv 0 \mod 2.$$

(**3.3.4**) **Completing the S-case.**

With the two swapping arguments, first with swapping around c and then with swapping around a control vertex, we have covered all the $\mathscr{S}$ sets in Equation (3.3.1)! In fact they are really the same argument as we can think of the 2-valent vertex c as acting as a special type of control vertex. Thus we have everything we need to prove the c_2 completion conjecture for $p = 2$ in the S-case.

Theorem 3.3.7 — *Let G be a connected 4-regular graph. Let v and w be adjacent vertices of G such that they share one common neighbour. Then,*

$$c_2^{(2)}(G - v) = c_2^{(2)}(G - w).$$

PROOF. Combining Proposition 3.3.2, Theorem 3.3.3 and Corollary 3.3.6 gives the result. ∎

3.4. The R-case

The last case we need to deal with is the R-case. Let G be a connected 4-regular graph, and let v and w be two adjacent vertices of G such that they do not share any neighbours. Let $R = G - \{v, w\}$ be the graph obtained by removing vertices v and w from G, the last grey blob in Figure 3.1, with the neighbours of v and w labelled $\{a, b, c, d, e, f\}$ as in the figure. Here w has neighbours $\{a, b, c, v\}$, and v has neighbours $\{d, e, f, w\}$.

The difficulty of the R-case stems from the fact that v and w no longer share any neighbours, and thus there is no distinguishing feature of R that we can readily exploit. However, the lack of specialness of the vertices $\{a, b, c, d, e, f\}$ is what lends itself to the symmetric nature of the R-case. To prove the conjecture for $p = 2$, we needed to use this symmetry to our advantage.

As we no longer have any 2-valent vertex to swap around, we look towards the more general control-vertex argument. Notice in the swapping around a control vertex argument for the S-case there was a symmetric flavour in how the new edge partitions that we ended up with could have been from any $\mathscr{S}$ set of a similar form to the original, but we didn't need to know exactly which one. Using a simplified version of the involution in the S-case, which was actually first discovered for the R-case, we use this symmetry to partially deal with the R-case.

However, like in the S-case, this particular control vertex argument was not enough. Once again an argument involving compatible cycles (see §5 of [Yea18b]) was used for the remaining sets when G had an odd number of vertices. To get rid of this parity requirement, a more complicated control vertex argument is necessary this time involving multiple control vertices.

Once again, to give a complete, self-contained proof of the R-case, we begin with the same set-up and partial results as from [Yea18b], and in Section 3.4.3 we present our new results, with allows us to finish the R-case for $p = 2$!

(**3.4.1**) **Set-up.**

As with the previous cases, we start with defining the sets of edge partitions that we are counting.

Definition 3.4.1 — (Definition 3.1 of [Yea18b]) *Suppose P is a bipartition of $\{a, b, c, d, e, f\}$. Let $\mathscr{R}_P$ be the set of bipartitions (ψ, ϕ) of the edges of R such that ψ is a spanning tree and ϕ is a spanning 2-forest compatible with P. Let $r_P = |\mathscr{R}_P|$.*

This time because of the symmetry of the neighbours of v and w, and since we are counting modulo 2, instead of picking specific vertex bipartitions for the spanning 2-forests we can add them all together.

Proposition 3.4.2 — (Proposition 3.2, 3.3 of [Yea18b]) *When v and w have no common neighbours*

$$c_2^{(2)}(G - v) = r_{\{a\},\{b,c\}} + r_{\{b\},\{a,c\}} + r_{\{c\},\{a,b\}} \mod 2,$$

$$c_2^{(2)}(G - w) = r_{\{d\},\{e,f\}} + r_{\{e\},\{d,f\}} + r_{\{f\},\{d,e\}} \mod 2,$$

47

and thus we have

$$c_2^{(2)}(G-v) - c_2^{(2)}(G-w) = r_{\{a\},\{b,c,d,e,f\}} + r_{\{b,c\},\{a,d,e,f\}} \qquad (\textbf{3.4.1})$$
$$+ r_{\{b\},\{a,c,d,e,f\}} + r_{\{a,c\},\{b,d,e,f\}}$$
$$+ r_{\{c\},\{a,b,d,e,f\}} + r_{\{a,b\},\{c,d,e,f\}}$$
$$+ r_{\{d\},\{a,b,c,e,f\}} + r_{\{e,f\},\{a,b,c,d\}}$$
$$+ r_{\{e\},\{a,b,c,d,f\}} + r_{\{d,f\},\{a,b,c,e\}}$$
$$+ r_{\{f\},\{a,b,c,d,e\}} + r_{\{d,e\},\{a,b,c,f\}} \quad \mod 2.$$

PROOF. Like in the proof of Proposition 3.2.2, we can pick specific vertex partitions for the $G-v$ and $G-w$ cases to equivalently determine each c_2 via counting edge partitions into a spanning tree and a spanning 2-forest compatible with the respective vertex partition. However, since the same argument holds for all vertex partitions of $\{a,b,c\}$ and $\{d,e,f\}$ into the form $\{*\},\{*,*\}$, we have that

$$c_2^{(2)}(G-v) \equiv r_{\{a\},\{b,c\}} \equiv r_{\{b\},\{a,c\}} \equiv r_{\{c\},\{a,b\}} \quad \mod 2,$$
$$c_2^{(2)}(G-w) \equiv r_{\{d\},\{e,f\}} \equiv r_{\{e\},\{d,f\}} \equiv r_{\{f\},\{d,e\}} \quad \mod 2.$$

Since we are working modulo 2, we can add each of the three choices together

$$c_2^{(2)}(G-v) \equiv r_{\{a\},\{b,c\}} + r_{\{b\},\{a,c\}} + r_{\{c\},\{a,b\}} \quad \mod 2,$$
$$c_2^{(2)}(G-w) \equiv r_{\{d\},\{e,f\}} + r_{\{e\},\{d,f\}} + r_{\{f\},\{d,e\}} \quad \mod 2.$$

Then, enumerating over all possibilities for the vertex set $\{a,b,c,d,e,f\}$ in the spanning 2-forests, we obtain the final equation by subtracting the expanded counts for $G-v$ and $G-w$ and simplifying modulo 2. To illustrate this, expanding the counts for $r_{\{a\},\{b,c\}}$ gives

$$r_{\{a\},\{b,c\}} = r_{\{a\},\{b,c,d,e,f\}}$$
$$+ r_{\{a,d\},\{b,c,e,f\}} + r_{\{a,e\},\{b,c,d,f\}} + r_{\{a,f\},\{b,c,d,e\}}$$
$$+ r_{\{a,d,e\},\{b,c,f\}} + r_{\{a,d,f\},\{b,c,e\}} + r_{\{a,e,f\},\{b,c,d\}}$$
$$+ r_{\{a,d,e,f\},\{b,c\}}.$$

Now notice that the terms in the middle two lines appear in the expanded counts for the terms in the $G-w$ case, for example, $r_{\{a,d\},\{b,c,e,f\}}$ and $r_{\{a,e,f\},\{b,c,d\}}$ both appear in the expansion of $r_{\{d\},\{e,f\}}$ and similarily for the other four terms. Thus we would be left with $r_{\{a\},\{b,c,d,e,f\}}$ and $r_{\{b,c\},\{a,d,e,f\}}$ after the subtraction.

As this holds for all $r_{\{*\},\{*,*\}}$, where the $*$'s are partitioning either $\{a,b,c\}$ or $\{d,e,f\}$, after subtracting the $G-v$ and $G-w$ counts we are left with terms of the form $r_{\{*\},\{*,*,*,*,*\}}$ and $r_{\{y,z\},\{x,*,*,*\}}$, where $\{x,y,z\}$ is either $\{a,b,c\}$ or $\{d,e,f\}$. ■

(3.4.2) Swapping around a control vertex.

Immediately from Equation (3.4.1) we see that there are 6 terms corresponding to sets of the form $\mathscr{R}_{\{y,z\},\{x,*,*,*\}}$ where $\{x,y,z\}$ is either $\{a,b,c\}$ or $\{d,e,f\}$. The thing to note is that these vertex bipartitions for the 2-forests are exactly in the form where we can apply Lemma 3.3.4, this time using the other specification for the special vertex x. As we already have an idea of how to deal with a control vertex from this lemma, we can use a simplified version of the involution in Theorem 3.3.5.

Theorem 3.4.3 — (Lemma 4.4 of [Yea18b]) *There is a fixed-point free involution on*

$$\mathscr{R}_{\{a,b\},\{c,d,e,f\}} \cup \mathscr{R}_{\{a,c\},\{b,d,e,f\}} \cup \mathscr{R}_{\{b,c\},\{b,d,e,f\}}$$
$$\cup \mathscr{R}_{\{d,e\},\{a,b,c,f\}} \cup \mathscr{R}_{\{d,f\},\{a,b,c,e\}} \cup \mathscr{R}_{\{e,f\},\{a,b,c,d\}},$$

and thus we have

$$r_{\{a,b\},\{c,d,e,f\}} + r_{\{a,c\},\{b,d,e,f\}} + r_{\{b,c\},\{b,d,e,f\}}$$
$$+ r_{\{d,e\},\{a,b,c,f\}} + r_{\{d,f\},\{a,b,c,e\}} + r_{\{e,f\},\{a,b,c,d\}} \equiv 0 \mod 2.$$

PROOF. Consider the edge partitions in the union of sets of the form $\mathscr{R}_{\{y,z\},\{x,*,*,*\}}$ where $\{x, y, z\}$ is either the trio $\{a, b, c\}$ or $\{d, e, f\}$, and the $*$'s are the other three vertices in $\{a, b, c, d, e, f\} \setminus \{x, y, z\}$. Since we can always distinguish the vertex x that is alone from its trio, we can choose x uniquely to be the special vertex. Thus these edge partitions satisfy the conditions of Lemma 3.3.4. This time because of the special form of the vertex bipartitions, we will specify for x the second property from the lemma, that x is either the control vertex or x is in one of the singleton components after removing the control vertex from its tree in the spanning 2-forest.

Let (ψ, ϕ) be an edge partition in any set of the form $\mathscr{R}_{\{y,z\},\{x,*,*,*\}}$, and let v be its control vertex. Let t be the tree in ϕ corresponding to $\{x, *, *, *\}$, which also contains v. Since v is a leaf in ψ, let η_v be the edge incident to v and n_v the neighbour of v in ψ.

Now to describe the involution which gives a new edge partition (ψ', ϕ'):

(i) *Swapping stays in* – If $n_v \in t$, so the control vertex and its neighbour in ψ are in the same tree of ϕ, let η be the edge incident to v in t in the path to n_v.
Then, swap the edges η_v and η between ψ and ϕ.
(ii) *Swapping goes out* – Otherwise if $n_v \notin t$, so the control vertex and its neighbour in ψ are in different trees of ϕ, let η be the edge incident to v in t in the path to the component of $t - v$ that contains exactly two vertices of $\{x, *, *, *\}$; in Figure 3.2b this is the edge towards the 2-part indicated by the dashed line.
Then, swap the edges η_v and η between ψ and ϕ.

Notice that this is essentially phase (2) in Theorem 3.3.5, just without the extra swapping of edges in (2ii), and thus most of the proof still holds for this simplified involution. Firstly, as we are always changing which edge is incident to v in ψ and ϕ, $(\psi', \phi') \neq (\psi, \phi)$, and so this transformation is fixed-point free.

Now for (i), we can argue exactly as we did for (2i) in Theorem 3.3.5 to show that (ψ', ϕ') stays a valid edge partition in the same set $\mathscr{R}_{\{y,z\},\{x,*,*,*\}}$ as (ψ, ϕ). Furthermore, it was shown that if v is still the control vertex after the transformation, which we will prove shortly, then (ψ', ϕ') remains in case (i) and applying the transformation again brings us back to (ψ, ϕ).

For (ii), a little more work is needed to show that (ψ', ϕ') is in a valid $\mathscr{R}$ set but the general framework is the same. By the choice of η, removing η from t splits the tree into two components that partitions $\{x, *, *, *\}$ into the form $\{x, x'\}, \{*, *\}$ for some $x' \in \{a, b, c, d, e, f\} \setminus \{x, y, z\}$. Since $n_v \notin t$, adding the edge η_v connects the component from t containing v to the other tree of ϕ (corresponding to part $\{y, z\}$), to create a new spanning 2-forest ϕ'. In particular, as v is in the component corresponding to $\{x, x'\}$, ϕ' is now compatible with the vertex bipartition $\{*, *\}, \{x', x, y, z\}$. As v is a leaf in ψ, removing η_v from ψ
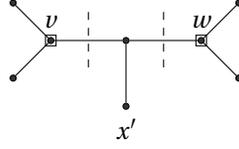
FIGURE 3.3: The two control vertices v and w for Lemma 3.4.4. The unlabelled leaves and x' are all vertices in p. The dashed lines indicate the edges we will be picking for one of the involutions in the R-case. Some of the leaf edges are possibly contracted, with the exception that the edges of two leaves adjacent to the same vertex cannot simultaneously both be contracted.

and adding η to create ψ' maintains the spanning tree structure. Thus we have that (ψ', ϕ') is a valid edge partition in the set $\mathscr{R}_{\{*,*\},\{x',x,y,z\}}$. Notice that as $\{x,y,z\}$ is either $\{a,b,c\}$ or $\{d,e,f\}$, the other trio must be $\{x',*,*\}$ in any order, and so $\{*,*\},\{x',x,y,z\}$ is in the form we want.

Once again if v is remains the control vertex, v and its neighbour in ψ' are in different trees of ϕ', and thus we stay in case (ii). As x' is now the special vertex, η_v is exactly the edge incident to v connecting the component with $\{x',x\}$ and the component with $\{y,z\}$ in ϕ', that is $\eta' = \eta_v$. Thus applying the transformation again returns us to (ψ, ϕ).

Finally what's left to prove is that the control vertex v stays the same across the transformation. As in the proof of Theorem 3.3.5, we can formulate the control vertex in an alternate way. This time v is the first common vertex in t in all three paths from x to the other vertices in part $\{x,*,*,*\}$. Under this formulation, we immediately see that this property does not change after the transformation, where now v is the first common vertex in the tree corresponding to part $\{x',x,y,z\}$ in ϕ' from x' to the other three vertices. Thus v remains the control vertex in (ψ', ϕ').

Putting everything together, we indeed have a fixed-point free involution on the union of sets of the form $\mathscr{R}_{\{y,z\},\{x,*,*,*\}}$ where $\{x,y,z\}$ is either $\{a,b,c\}$ or $\{d,e,f\}$,

$$\mathscr{R}_{\{a,b\},\{c,d,e,f\}} \cup \mathscr{R}_{\{a,c\},\{b,d,e,f\}} \cup \mathscr{R}_{\{b,c\},\{b,d,e,f\}}$$
$$\cup \mathscr{R}_{\{d,e\},\{a,b,c,f\}} \cup \mathscr{R}_{\{d,f\},\{a,b,c,e\}} \cup \mathscr{R}_{\{e,f\},\{a,b,c,d\}},$$

and therefore, the size of this set must be even, giving the equation

$$r_{\{a,b\},\{c,d,e,f\}} + r_{\{a,c\},\{b,d,e,f\}} + r_{\{b,c\},\{b,d,e,f\}}$$
$$+ r_{\{d,e\},\{a,b,c,f\}} + r_{\{d,f\},\{a,b,c,e\}} + r_{\{e,f\},\{a,b,c,d\}} \equiv 0 \mod 2.$$

∎

(**3.4.3**) **Two control vertices.**

For the remaining 6 sets in Equation (3.4.1), we are no longer in a situation where we can apply Lemma 3.3.4 since these sets are of the form $\mathscr{R}_{\{x\},\{*,*,*,*,*\}}$ for $x \in \{a,b,c,d,e,f\}$. What we need is a new notion of "control vertex", this time for a tree with five marked vertices instead of four. However, as we no longer have any distinguishable vertices in this 5-part, we could not appropriately define one unique vertex as the control vertex. Rather two control vertices are necessary to get an involution on the correct sets.

Lemma 3.4.4 — *Suppose we have a bipartition of $\{a, b, c, d, e, f\}$ of the form $\{x\}, \{*, *, *, *, *\}$. Let (ψ, ϕ) be an edge partition in $\mathscr{R}_{\{x\}, \{*, *, *, *, *\}}$ where ψ is a spanning tree and ϕ is a spanning 2-forest compatible with $\{x\}, \{*, *, *, *, *\}$. Let t be the tree corresponding to part $p = \{*, *, *, *, *\}$ in the 2-forest ϕ.*

*Then, there is a unique pair of non-adjacent vertices v and w called the **control vertices**, such that removing v and w from t partitions p into all singletons. Figure 3.3 depicts the two control vertices. Furthermore, in t each control vertex will either be*

- *2-valent and in $\{a, b, c, d, e, f\} \setminus \{x\}$, or*
- *3-valent and not a vertex in $\{a, b, c, d, e, f\}$.*

In all cases, the control vertices are leaves in the spanning tree ψ.

PROOF. In a similar fashion to the proof of Lemma 3.3.4, we start by looking at the tree t and the subtree created by taking the union of all the paths in t between any two vertices in p. Since p has five vertices this subtree will be of the form
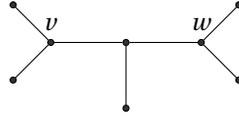
where the leaves are vertices in p, the edges are paths in t, and some of the paths may be contracted.

Since ψ is a spanning tree, the vertices in ϕ, and thus t, are at most 3-valent, and the vertices in $\{a, b, c, d, e, f\}$, and thus p, are at most 2-valent. In particular, we have that the above is the only possible form for the subtree, where some of the paths may be contracted. Note trivially, the paths between two leaves adjacent to the same non-leaf vertex cannot be contracted simultaneously. Now, the valency restrictions also mean that in the subtree form above, the paths from v and w to the middle non-leaf vertex cannot be contracted. Thus $v \neq w$, and furthermore, they are not adjacent in ϕ i.e. there must be at least two edges in the path from v to w. By the valency of the vertices in ϕ, v and w can only be 2-valent if they are in the set $\{a, b, c, d, e, f\}$ or otherwise 3-valent, making them both leaves in the spanning tree ψ. As R has more than two vertices, v and w are also not adjacent in ψ, and thus in R the control vertices are non-adjacent.

Finally, removing v and w from t clearly partitions p into all singletons and these are the unique pair of vertices which do so. The number of components of $t - \{v, w\}$ will depend on whether v or w are in the set $\{a, b, c, d, e, f\}$. ∎

Now that we've established the notion of control vertices, we can use a swapping argument once more for one final fixed-point free involution to cover the remaining sets in Equation (3.4.1). As there are two vertices to swap around, like in Theorem 3.3.5, we need a more complicated involution than previously.

Theorem 3.4.5 — *There is a fixed-point free involution on*

$$\mathscr{R}_{\{a\}, \{b,c,d,e,f\}} \cup \mathscr{R}_{\{b\}, \{a,c,d,e,f\}} \cup \mathscr{R}_{\{c\}, \{a,b,d,e,f\}}$$
$$\cup \mathscr{R}_{\{d\}, \{a,b,c,e,f\}} \cup \mathscr{R}_{\{e\}, \{a,b,c,d,f\}} \cup \mathscr{R}_{\{f\}, \{a,b,c,d,e\}},$$

and thus we have

$$r_{\{a\},\{b,c,d,e,f\}} + r_{\{b\},\{a,c,d,e,f\}} + r_{\{c\},\{a,b,d,e,f\}}$$
$$+ r_{\{d\},\{a,b,c,e,f\}} + r_{\{e\},\{a,b,c,d,f\}} + r_{\{f\},\{a,b,c,d,e\}} \equiv 0 \mod 2.$$

PROOF. By construction the edge partitions in the union of sets of the form $\mathscr{R}_{\{x\},\{*,*,*,*,*\}}$, where $x \in \{a,b,c,d,e,f\}$, and the $*$'s are the rest of the vertices $\{a,b,c,d,e,f\} \setminus \{x\}$, satisfy the conditions of Lemma 3.4.4. Let (ψ,ϕ) be an edge partition in any set of the form $\mathscr{R}_{\{x\},\{*,*,*,*,*\}}$, and let v,w be its control vertices. Let t be the tree in ϕ corresponding to part $\{*,*,*,*,*\}$ which contains v and w. Since the control vertices are leaves in ψ, let η_v and η_w be the edges incident to v and w in ψ, respectively, and let n_v and n_w be their neighbours.

Consider the following involution giving a new edge partition (ψ',ϕ'):

(1) *Swapping either stays in* – If $n_v \in t$ or $n_w \in t$:
   (i) *v stays in* – If $n_v \in t$, so control vertex v and its neighbour in ψ are in the same tree of ϕ, let η_1 be the edge incident to v in t in the path to n_v.
      Then, swap the edges η_v and η_1 between ψ and ϕ.
   (ii) *w stays in* – If $n_w \in t$, so control vertex w and its neighbour in ψ are in the same tree of ϕ, let η_2 be the edge incident to w in t in the path to n_w.
      Then, swap the edges η_w and η_2 between ψ and ϕ.
   Here we are using non-exclusive or, so if $n_v, n_w \in t$, then swap around both v and w as above.
(2) *Swapping both goes out* – Otherwise we have $n_v, n_w \notin t$, where the control vertices and their neighbours in ψ are in different trees of ϕ. Let η_1 be the edge incident to v in t in the path to w, and let η_2 be the edge incident to w in t in the path to v; in Figure 3.3 these are the edges indicated by the dashed lines.
   Then, swap edges η_v and η_w with η_1 and η_2 between ψ and ϕ.

Firstly, as we are always changing which edges are incident to v and/or w, (ψ',ϕ') must be different from (ψ,ϕ). Thus this transformation is fixed-point free.

For (1), both cases are symmetric so without loss of generality assume $n_v \in t$. Then, removing edge η_1 from ϕ breaks t into two components and because of how η_1 was chosen, n_v is now in a different component than v. As v and its neighbour in ψ, n_v, are in the same tree of ϕ, adding edge η_v reconnects the two components from t. By the choice of η_1, we cannot create any cycles by adding η_v, and thus ϕ' is a spanning 2-forest which is compatible with the same vertex bipartition as ϕ. Since v is a leaf in ψ, removing edge η_v and adding edge η_1 to create ψ' maintains the spanning tree structure. Thus (ψ',ϕ') is a valid edge partition in the same $\mathscr{R}$ set as (ψ,ϕ). We also have if the control vertices stay the same, (ψ',ϕ') will still have the property that v and its neighbour in ψ' are in the same tree of ϕ', and thus applying the transformation again returns to (ψ,ϕ).

When both $n_v, n_w \in t$, since $v \neq w$ and they are not adjacent, the four edges in the swap are all distinct; $\eta_1 \neq \eta_2$ and $\eta_v \neq \eta_w$. Thus we can simultaneously swap edges η_v and η_w with edges η_1 and η_2, respectively. Note that we need to swap around both control vertices as there is no way to distinguish v from w, and thus no way to ensure we get an involution if we arbitrarily picked one to swap edges around. Applying the above swapping argument twice, we have that (ψ',ϕ') is a valid edge partition in the same $\mathscr{R}$ set as before and applying the transformation again returns to (ψ,ϕ).

For (2) when $n_v, n_w \notin t$, once again we know that the four edges in the swap are all distinct by the same reasoning as when $n_v, n_w \in t$. Then, removing edges η_1 and η_2 breaks t into three subtrees (thus breaking ϕ into four components); one with v, one with w, and one with the vertex $x' \in \{a, b, c, d, e, f\} \setminus \{x\}$ as in Figure 3.3. Now since v and n_v are in different trees of ϕ, adding η_v connects the subtree from $t - \{\eta_1, \eta_2\}$ with v to the tree in ϕ corresponding to part $\{x\}$. Similarly, as w and n_w are in different trees of ϕ, adding η_w connects the subtree with w also to the tree with x in ϕ.

As the last subtree, which contains x', never gets connected to any other component, this transformation creates a new spanning 2-forest ϕ' which is compatible with the vertex bipartition $\{x'\}, \{x, *, *, *, *\}$. Since v, w are leaves in ψ, removing edges η_v and η_w from ψ disconnects the control vertices and adding edges η_1 and η_2 reconnects them, creating ψ' which is once again a spanning tree. Thus (ψ', ϕ') is a valid edge partition in $\mathscr{R}_{\{x'\},\{x,*,*,*,*\}}$, which is in the correct form. Once again, if the control vertices stay the same, then their neighbours in ψ will be in a different tree of ϕ' landing us back in (2). Thus applying the transformation again brings us back to (ψ, ϕ).

What's left of the proof is to make sure v and w are still the control vertices in (ψ', ϕ'). Notice we can formulate the control vertices as the two vertices in t such that for a specific x' in part $\{*, *, *, *, *\}$, each vertex is the last common vertex on the paths from x' to the exactly two of the other vertices in the part. The vertex x' is as depicted in Figure 3.3. Under this formulation, we immediately see that in (2) as ϕ' is created by connecting subtrees with v and w to the tree corresponding to part $\{x\}$ in ϕ, the vertex x now acts as the x' for ϕ'. Thus v and w remain the control vertices for (ψ', ϕ').

For (1), depending on where n_v or n_w were in the tree t, the vertex acting as x' in ϕ' may no longer be the same x' from ϕ. However, one can check that in all possible situations, the property for the control vertices is still satisfied by v and w, and thus v and w are still the control vertices.

Finally putting everything together, we indeed get a fixed-point free involution on the union of sets of the form $\mathscr{R}_{\{x\},\{*,*,*,*,*\}}$,

$$\mathscr{R}_{\{a\},\{b,c,d,e,f\}} \cup \mathscr{R}_{\{b\},\{a,c,d,e,f\}} \cup \mathscr{R}_{\{c\},\{a,b,d,e,f\}}$$
$$\cup \mathscr{R}_{\{d\},\{a,b,c,e,f\}} \cup \mathscr{R}_{\{e\},\{a,b,c,d,f\}} \cup \mathscr{R}_{\{f\},\{a,b,c,d,e\}},$$

and therefore, the size of this must be even

$$r_{\{a\},\{b,c,d,e,f\}} + r_{\{b\},\{a,c,d,e,f\}} + r_{\{c\},\{a,b,d,e,f\}}$$
$$+ r_{\{d\},\{a,b,c,e,f\}} + r_{\{e\},\{a,b,c,d,f\}} + r_{\{f\},\{a,b,c,d,e\}} \equiv 0 \mod 2.$$

$\blacksquare$

As a remark on why two contol vertices were necessary, looking at Figure 3.3 one would first be tempted to pick the unlabelled non-leaf vertex as the control vertex, call it u, and cut the edge towards x'. However, in some edge partitions, this path may be contracted with u being x' itself. In this case there is no obvious way to choose an edge incident to u to remove.

To mitigate this, one would then decide to cut the two edges incident to u that is not towards x'. However, in this case by valency arguments, u is a leaf in the corresponding spanning tree, and thus there is only one edge available to swap two edges with, which is not possible. Thus, we cannot pick u to be the control vertex.

Looking to v or w, as there are no special vertices in this 5-part of the vertex bipartition, there would be no way to distinguish between v and w to pick just one control vertex. Even if there were, only picking one of v or w to swap edges around would not swap to a spanning 2-forest compatible with a vertex bipartition that we care about. Therefore, we really did need to pick both vertices v and w to be control vertices!

Notice that we can think of this involution involving two control vertices as a generalized version of the involution used in Theorem 3.3.5. For that involution, the 2-valent vertex c acts as a second "control vertex".

(**3.4.4**) **Completing the R-case.**

While there were no special vertices in $\{a, b, c, d, e, f\}$ that we could readily use to swap between specific vertex bipartitions, we were able to take advantage of the symmetric nature of the R-case and use control vertices to find involutions where we only cared about the form of the vertex bipartition we may be swapping our edge partitions to. With the two swapping arguments, one involving one control vertex and the other involving two control vertices, we have everything we need to prove the c_2 completion conjecture for $p = 2$ in the R-case.

Theorem 3.4.6 — *Let G be a connected 4-regular graph. Let v and w be adjacent vertices of G such that they do not share any common neighbours. Then,*

$$c_2^{(2)}(G - v) = c_2^{(2)}(G - w).$$

PROOF. Together Proposition 3.4.2, Theorem 3.4.3 and Theorem 3.4.5 gives the result. ∎

3.5. COMPLETING THE $p = 2$ CASE

To prove the c_2 completion conjecture for $p = 2$, in all three cases the underlying idea remained the same: finding a vertex, or vertices, and picking particular edges incident to them to swap between the two parts of an edge partition, where this action gives a fixed-point free involution on specific sets of edge partitions.

In the T-case, we found that we could use the two 2-valent vertices to swap edges around. In the S-case, we reused the 2-valent vertex swapping argument and generalized it to give the notion of swapping around a particular vertex called the control vertex. We found that for a second involution, we needed a two-phase swapping process where we used both the control vertex and a 2-valent vertex. Finally in the R-case, we reused the control vertex argument and found that we needed to extend it to include two control vertices for a second involution, this extension itself being a generalized version of the second involution in the S-case.

We note here that the two new bijections, giving Theorems 3.3.5 and 3.4.5 in the S and R cases, respectively, were implemented in Sage [Sage] to verify our results on some small graphs.

Assembling everything together, we finally complete the c_2 completion conjecture for $p = 2$!

Theorem 3.5.1 — *Let G be a connected 4-regular graph, and let v and w be vertices of G. Then,*
$$c_2^{(2)}(G - v) = c_2^{(2)}(G - w).$$

PROOF. As in the discussion in Section 3.1, since G is connected there is path between any two vertices of G, and thus it suffices to prove the conjecture when v and w are adjacent vertices. For any two non-adjacent vertices, we could then just follow any path between them, getting a chain of equivalences to obtain the required result.

Then, as G is a 4-regular graph, the four possibilities for the neighbours of v and w are:

- v and w share all neighbours, or
- v and w have exactly two common neighbours – T-case, or
- v and w have only one common neighbour – S-case, or
- v and w do not share any neighbours – R-case.

When v and w share all neighbours, notice that $G - v$ and $G - w$ are isomorphic. Thus, trivially, their c_2's are equivalent. When v and w share some neighbours or none at all, Corollary 3.2.5, Theorem 3.3.7, and Theorem 3.4.6 give the result in the T, S, and R cases, respectively. This completes the proof. ∎

II

LE DIAGRAMS

In the second part of this thesis we move to a different story but one which is still very much in the same flavour as Part I, on the emergence of combinatorics and using combinatorial techniques to study aspects of scattering amplitudes. This time, instead of looking at an arithmetic invariant related to Feynman diagrams through enumerating certain edge bipartitions, we are looking at a geometric story of triangulations through a simple combinatorial map where one side of the map is related to on-shell diagrams.

We start with introducing the main objects of interest, the positive Grassmannian and some of the combinatorial objects in bijection with the cells that decompose it. Finishing off this introductory chapter, in Section 4.3.3 we set-up the motivation for the question we investigate in Chapter 5, which looks at the map mentioned above through the lens of a different combinatorial object. The exposition from this chapter follows mainly from [Pos06; LPW20].

4.1. GRASSMANNIA

The Grassmannian is one of those classical geometric objects in mathematics that has been extensively studied due to its nice structure and its many connections to different areas of mathematics, including (of course) combinatorics, algebraic and differential geometry, and representation theory. For one, the Grassmannian is at the heart of **Schubert calculus**, which aims to solve problems in enumerative geometry via the Grassmannian, Schubert varieties, and flag varieties, and which lies at the intersection of combinatorics and algebraic geometry. An example of one of the early problems studied via Schubert calculus is [KL72]: "How many lines in 3-space, in general, intersect four given lines?" (of which the answer is generically 2). As an added benefit, the Grassmannian can be described very simply.

Definition 4.1.1 — *For $0 \leq k \leq n$, the (real) **Grassmannian** $\mathrm{Gr}_{k,n}(\mathbb{R})$ is the space of all k-dimensional (linear) subspaces of $\mathbb{R}^n$.*

We can think of the elements V of the Grassmannian as represented by real $k \times n$ matrices A of rank k modulo invertible row operations, where V is the subspace spanned by the row vectors of A. That is,

$$\mathrm{Gr}_{k,n}(\mathbb{R}) \cong \mathsf{GL}(k,\mathbb{R}) \backslash \mathsf{Mat}^*(k,n) \, ,$$

where $\mathsf{Mat}^*(k,n)$ is the set of all real $k \times n$ matrices of rank k. The dimension of $\mathrm{Gr}_{k,n}$ is $k(n-k)$. We often drop the reference to the field $\mathbb{R}$ and just refer to the real Grassmannian as $\mathrm{Gr}_{k,n}$. As a geometric object, we can view the Grassmannian through a couple of different lenses (which we do not go through here): as a compact smooth manifold, as a projective variety, as a scheme, and as a CW-complex.

Example 4.1.2 — $\mathrm{Gr}_{1,n}$ *consists of all lines through the origin in $\mathbb{R}^n$, or viewing through matrices all $1 \times n$ matrices modulo scaling, which is equivalent to projective space $\mathbb{P}^{n-1}$.*

For algebraic combinatorialists, the interest in the Grassmannian lies in the various beautiful connections to combinatorics arising from its decompositions. Through the classic Schubert decomposition of the Grassmannian into **Schubert cells**, which can be indexed by partitions, we are led to some familiar combinatorial machinery such as Young tableaux, Schur functions, and Schubert polynomials. A standard reference for many of these topics is [Ful97].

From a different decomposition, as first described by Gelfand, Goresky, MacPherson, and Serganova in [Gel+87], and which we will present here, we are led to matroids and matroid polytopes, opening up a different connection to combinatorics.

First, let's fix some notation. Let $[n]$ denote the set $\{1, \ldots, n\}$ and $\binom{[n]}{k}$ be the set of all k-element subsets of $[n]$. Given a $k \times n$ matrix A of rank k and an $I \in \binom{[n]}{k}$, let $p_I(A) := \det(A_I)$, where A_I is the $k \times k$ matrix of A restricted to the columns indexed by I. We call this a **maximal minor** of A. For $V \in \text{Gr}_{k,n}$ represented by A, the **Plücker coordinates** of V are then defined to be $p_I(V) := p_I(A)$ for all $I \in \binom{[n]}{k}$, where we note that $p_I(V)$ does not depend on the choice of representative matrix A.

Now, thinking about $V \in \text{Gr}_{k,n}$ as given by the n column vectors of A which together span $\mathbb{R}^k$, since A is of rank k, we can then look at all the k-subsets of these n column vectors such that they form a basis for $\mathbb{R}^k$. In particular, each of these subsets $I \in \binom{[n]}{k}$ must have the property that $p_I(A) \neq 0$. Viewing through the combinatorial lens, in other words we can construct a representable matroid $\mathcal{M}$ over the ground set $[n]$ which has bases

$$M(A) := \left\{ I \in \binom{[n]}{k} \; : \; p_I(A) \neq 0 \right\},$$

and where A represents $\mathcal{M}$ over $\mathbb{R}$. For a standard reference on matroids, we refer the reader to [Oxl11].

Going back to the Grassmannian, we can then divide the elements of the Grassmannian based on which Plücker coordinates are non-zero, into **matroid strata**, also known as the Gelfand-Serganova strata. That is, for any $M \subset \binom{[n]}{k}$ we can define

$$\mathcal{S}_M := \left\{ V \in \text{Gr}_{k,n} \; : \; p_I(V) \neq 0 \text{ if and only if } I \in M \right\}.$$

Equivalently, if A is a $k \times n$ matrix representing V, then the condition reduces to $M = M(A)$ where $M(A)$ is as defined above. Notice that M defines a representable matroid of rank k, where here we are taking the convention of defining a matroid by its bases. Thus we get the following (disjoint) decomposition, called the **matroid stratification**, also known as the Gelfand-Serganova stratification, of the Grassmannian,

$$\text{Gr}_{k,n} = \bigsqcup_{\text{representable matroids } M} \mathcal{S}_M.$$

However, unfortunately Mnëv's Universality Theorem [Mnë88] tells us that the structure of these $\mathcal{S}_M$ can be very complicated, in fact as complicated as any algebraic variety!

(**4.1.1**) **The positive Grassmannian.**

Instead of looking at the full Grassmannian, Postnikov in [Pos06] initiated the study of a certain subset of the Grassmannian called the positive Grassmannian, through giving a combinatorial description of

its cells, called positroids, which turned out to have a much nicer geometric structure. This opened the door to the extensive study of the positive Grassmannian, both combinatorially and through the multitude of emerging connections with other branches of mathematics and physics. Here we just define the positive Grassmannian and in the next section, we will delve into some aspects of its decomposition.

As an analogue to totally positive and totally non-negative matrices, we can extend this notion of positivity to the matrices underlying the Grassmannian.

Definition 4.1.3 — (Definition 3.1 of [Pos06]) *We say that a $k \times n$ matrix A of rank k is **totally non-negative**, resp. **totally positive**, if all of its maximal minors are non-negative i.e. $p_I(A) \geq 0$, resp. all positive $p_I(A) > 0$, for all $I \in \binom{[n]}{k}$. We then say that $V \in \mathrm{Gr}_{k,n}(\mathbb{R})$ is **totally non-negative**, resp. **totally positive**, if V is represented by a $k \times n$ matrix A that is totally non-negative i.e. $p_I(V) \geq 0$, resp. totally positive $p_I(V) > 0$, for all $i \in \binom{[n]}{k}$.*

*The **totally non-negative (TNN) Grassmannian**, also referred to as the **positive Grassmannian**, $\mathrm{Gr}_{k,n}^{\geq 0}$ is the set of all totally non-negative $V \in \mathrm{Gr}_{k,n}(\mathbb{R})$. The **totally positive Grassmannian** $\mathrm{Gr}_{k,n}^{>0}$ is the set of all totally positive $V \in \mathrm{Gr}_{k,n}(\mathbb{R})$.*

Note that the positive Grassmannian $\mathrm{Gr}_{k,n}^{\geq 0}$ is a closed subset of the (full) Grassmannian $\mathrm{Gr}_{k,n}$ of dimension $k(n-k)$, which is also the dimension of $\mathrm{Gr}_{k,n}$. For the rest of this chapter, we will focus on the positive Grassmannian.

We end this section with an example of an element in the positive Grassmannian.

Example 4.1.4 — *Consider the following 2×4 matrix of rank 2, which represents an element V in $\mathrm{Gr}_{2,4}$:*

$$A = \begin{bmatrix} 1 & 0 & -1 & -2 \\ 0 & 1 & 2 & 3 \end{bmatrix}.$$

Then looking at the Plücker coordinates, for the following order on the 2-subsets of $[4]$:

$$\binom{[4]}{2} = \{12, 13, 14, 23, 24, 34\},$$

we have the sequence

$$p_I(A) = 1, 2, 3, 1, 2, 1,$$

where as a sample calculation

$$p_{34}(A) = \det \begin{bmatrix} -1 & -2 \\ 2 & 3 \end{bmatrix} = 1.$$

In particular, $p_I(A) \geq 0$ for all $I \in \binom{[4]}{2}$, and thus V is an element in the positive Grassmannian $\mathrm{Gr}_{2,4}^{\geq 0}$. As we actually have $p_I(A) > 0$ for all I, V is also in the totally positive Grassmannian $\mathrm{Gr}_{2,4}^{>0}$. Here we have that $M(A) = \binom{[4]}{2}$, which is a uniform matroid!

4.2. POSITROIDS AND RELATED COMBINATORIAL OBJECTS

Like how the Grassmannian could be subdivided into its matroid strata, analogously we can do the same with the positive Grassmannian, this time based on which maximal minors are positive.

Definition 4.2.1 — (Definition 3.2 of [Pos06]) *Let M be a subset of $\binom{[n]}{k}$, and define*

$$S_M := \mathscr{S}_M \cap \mathrm{Gr}_{k,n}^{\geq 0} = \left\{ V \in \mathrm{Gr}_{k,n}^{\geq 0} \; : \; p_I(V) > 0 \text{ for all } I \in M \text{ and } p_I(V) = 0 \text{ for all } I \in \binom{[n]}{k} \setminus M \right\}.$$

*If $S_M \neq \emptyset$, we call M a **positroid**, and S_M is its **positroid cell**.*

Note that M is a rank k representable matroid, represented by a $k \times n$ matrix with non-negative maximal minors, with the term "positroid" arising from abbreviating "positive matroid". However, the positroid is not invariant under matroid isomorphism but is invariant under cyclic shifts of elements. Additionally, notice that the totally positive Grassmannian corresponds to the uniform matroid $M = \binom{[n]}{k}$ and is the unique top-dimensional cell (of dimension $k(n-k)$) of the positive Grassmannian. From Example 4.1.4, we have that the $M(A) = \binom{[4]}{2}$ is actually a positroid!

In other words, we are partitioning the positive Grassmannian based on the Plücker coordinates that are strictly positive. We call this (disjoint) decomposition the **positroid stratification** of the positive Grassmannian

$$\mathrm{Gr}_{k,n}^{\geq 0} = \bigsqcup_{M \text{ is a positroid}} S_M.$$

What's nice about the geometric structure of these positroid cells is that they are indeed cells.

Theorem 4.2.2 — (Theorem 3.5 of [Pos06], Theorem 5.4 of [PSW09]) *Each positroid cell S_M is homeomorphic to an open ball. The decomposition of $\mathrm{Gr}_{k,n}^{\geq 0}$ into the union of positroid cells is a CW-complex.*

For us the interest in positroid cells lies in its surprisingly rich combinatorial structure, arising from the numerous families of combinatorial objects that Postnikov in [Pos06] showed index these cells. Other than positroids, these objects include decorated permutations, Le diagrams, Grassmann necklaces, and equivalence classes of reduced plabic graphs. We present these objects and one of the bijections between them, giving us a way to label positroid cells by their associated combinatorial objects.

(**4.2.1**) **Decorated permutations.**

We start with the simplest family of combinatorial objects called decorated permutations.

Definition 4.2.3 — *A **decorated permutation** of $[n]$ is a permutation π on $[n]$ where every fixed-point is designated (coloured) as a **loop** (black), denoted by $\pi(i) = \underline{i}$, or a **co-loop** (white), denoted by $\pi(i) = \overline{i}$. An **anti-excedance** of π is an element i such that either $\pi^{-1}(i) > i$ or i is a co-loop. We refer to $\pi^{-1}(i)$ as the **position** of i.*

Example 4.2.4 — (from [LPW20]) *The following is a decorated permutation of* [8] *in two-line notation:*

$$\pi = \left(\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & \underline{2} & 5 & 1 & 6 & 8 & \overline{7} & 4 \end{array} \right),$$

which has 3 anti-excedances, $1, 4, 7$, *as these elements when viewed on the bottom line, do not exceed their corresponding element in the top line and are not loops. Here* π *has one loop,* 2, *and one co-loop,* 7.

To show that a specific type of decorated permutation is in bijection with the cells of the positive Grassmannian, Postnikov utilized another object called the **Grassmann necklace** $\mathscr{I}$ which is directly in bijection with positroids and can be read off from the bases of the positroid. Decorated permutations can then be seen as expressing the information from Grassmann necklaces in a more compact way. We refer the reader to §16 of [Pos06] for more details.

Theorem 4.2.5 — (Lemma 16.2 of [Pos06]) *There is a bijection between decorated permutations* π *of* [n] *with k anti-excedances and Grassmann necklaces* $\mathscr{I}$ *of type* (k, n). *Therefore, decorated permutations of* [n] *with k anti-excedances index the cells of* $\mathrm{Gr}_{k,n}^{\geq 0}$, *and we denote by* S_π *the positroid cell indexed by* π.

Going back to Example 4.2.4, this means that π indexes the cell S_π of $\mathrm{Gr}_{3,8}^{\geq 0}$.

(**4.2.2**) **Le diagrams and pipe dreams.**

While Grassmann necklaces can directly see the bases of the positroid, and decorated permutations are simple and succinct objects that encode positroids, one property that both do not easily see is the dimension of the associated positroid cell. For this, we need the next family of combinatorial objects called Le diagrams.

Definition 4.2.6 — (Definition 6.1 of [Pos06]) *A ⅃-diagram (Le diagram), is a filling D of a Young diagram of shape* λ *with* 0*'s and* +*'s such that D avoids the ⅃-configuration:*

$$
\begin{array}{c}
\boxed{+} \\
\vdots \\
\boxed{+} \cdots \boxed{0}
\end{array}
$$
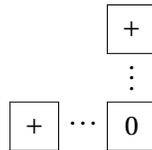
That is, no 0 *has both a* + *above and to the left of it, which we refer to as the ⅃-condition.*
For $0 \leq k \leq n$ *we say that the ⅃-diagram D is of* **type (k,n)** *if the shape* λ *fits inside a* $k \times (n-k)$ *rectangle.*

An example of a ⅃-diagram D is given in Figure 4.1a, which we will see shortly is the ⅃-diagram associated to the decorated permutation in Example 4.2.4. D is indeed a ⅃-diagram as it avoids the ⅃-configuration, and it is of type $(3, 8)$ since the shape of D fits inside a $3 \times 5 = 3 \times (8 - 3)$ rectangle.

In §20 of [Pos06], Postnikov gave two bijections between ⅃-diagrams and decorated permutations, the first through associating a series of other objects (hook diagrams, networks, and plabic graphs) to ⅃-diagrams. The second, which we describe here, uses an algorithm from §19 of [Pos06] going through a slightly different object called pipe dreams.

Given a ⌐-diagram D of type (k, n), we associate a decorated permutation π_D on $[n]$ as follows:

(i) In D, we replace each 0 with a cross $+$ and each $+$ with an elbow joint $\diagdown$:

$$\boxed{0} \longmapsto \boxplus$$

$$\boxed{+} \longmapsto \diagdown$$
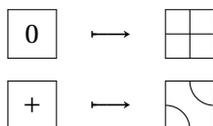
(ii) Notice that we can view the south-east (SE) border of D as a lattice path with n steps, and thus label the edges with $1, \ldots, n$ from the top-right corner to the bottom-left corner.

(iii) Then label the edges of the north-west (NW) border of D so that, viewing D as a grid, the rows and columns have the same labels (the opposite horizontal/vertical edges are labelled the same). We call this diagram P the **pipe dream** associated to D.

(iv) To get the decorated permutation π associated to P and D, we follow the "pipes" of P from the SE border to the NW border. That is, $\pi(i) = j$ if the pipe starts at i and ends at j. A horizontal pipe starting and ending at i, where i labels vertical edges, is denoted a co-loop $\pi(i) = \bar{i}$. A vertical pipe starting and ending at i, where i labels horizontal edges, is denoted a loop $\pi(i) = \underline{i}$.

To illustrate this algorithm, we use an example.

Example 4.2.7 — *Given the ⌐-diagram D of type $(3, 8)$ in Figure 4.1a, we label the SE and NW borders with $1, \ldots, 8$ as above, and replace the 0's and $+$'s with crosses and elbow-joints. This gives the pipe dream P in Figure 4.1b, where arrows are added to show the direction to follow the pipes for the permutation.*

(a) A ⌐-diagram D of type $(3, 8)$.

(b) The pipe-dream P associated to D in Figure 4.1a.
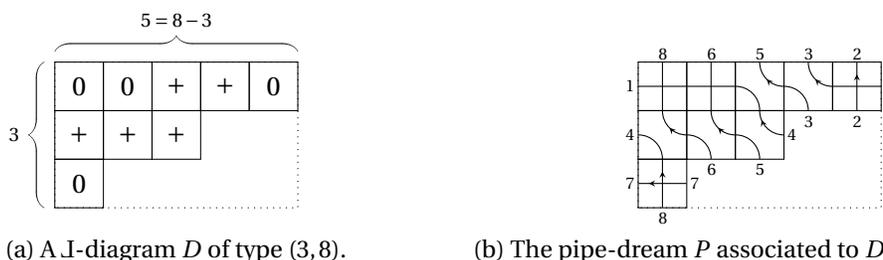
FIGURE 4.1: The ⌐-diagram and pipe dream associated to Example 4.2.4. The arrows on the pipes indicate the direction to follow to read off the decorated permutation.

To read off the decorated permutation, we follow the pipes in the direction of the arrows to get

$$\pi_D = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & \underline{2} & 5 & 1 & 6 & 8 & \overline{7} & 4 \end{pmatrix},$$

which is the decorated permutation from Example 4.2.4! Note that π_D is a decorated permutation on $[8]$ with 3 anti-excedances.

From the example, we make a few observations about this map $D \mapsto \pi_D$, where P is the associated pipe dream to D:

– Rows of all 0's in D correspond to horizontal pipes in P, and thus these are the co-loops in π_D. In Example 4.2.7 this would be the row labelled 7 in Figure 4.1.

- Similarly, columns of all 0's in D correspond to vertical pipes in P, and thus these are the loops in π_D. In Example 4.2.7 this would be the column labelled 2 in Figure 4.1.
- Because of the way the elbow joints are drawn, notice that pipes in P can only ever go to the left and up from its starting point. Thus, pipes starting from i that end on vertical edges j must have $i \geq j$, with $i = j$ being a horizontal pipe. Whereas pipes starting from i and ending on horizontal edges j have $i < j$. That is, the labels of the rows (vertical edges) of P exactly correspond to the anti-excedances of π_D. In Example 4.2.7, this would be the rows labelled $1, 4, 7$ in Figure 4.1, which indeed are the anti-excedances of π_D.
- Finally, avoiding the ⅃-configuration in D corresponds to pipes in P crossing at most once, and furthermore once two pipes cross, they will also no longer touch (via an elbow joint).

In turns out that this map is indeed a bijection, and furthermore one of the reasons to look at ⅃-diagrams is because of how easily they display the dimension of the positroid cell that they index. In particular, the dimension of the corresponding positroid cell is determined by counting the number of $+$'s in the ⅃-diagram.

Theorem 4.2.8 — (Theorem 6.5, Corollary 20.1, Theorem 20.3 of [Pos06]) *The map $D \mapsto \pi_D$ is a bijection from ⅃-diagrams of type (k, n) to decorated permutations of $[n]$ with k anti-excedances. Therefore, ⅃-diagrams of type (k, n) index the cells of $\mathrm{Gr}_{k,n}^{\geq 0}$. Furthermore, let S_D be the positroid cell indexed by the ⅃-diagram D. Then the dimension of the cell S_D is the number of $+$'s in D.*

Going back to Example 4.2.7, this means that D indexes the cell S_D of $\mathrm{Gr}_{3,8}^{\geq 0}$, which corresponds to the cell S_π from Example 4.2.4, and the dimension of S_D is 5 since there are 5 $+$'s in D .

(**4.2.3**) **Plabic graphs.**

The final family of combinatorial objects that we will define (without going into detail) are plabic graphs, where the term "plabic" comes from abbreviating "planar bicoloured". These graphs play an important role in the foundations for the original correspondences of the above objects to positroid cells as well as the basis of many of the applications and connections from positroids to other fields.

Definition 4.2.9 — (Definition 11.5 of [Pos06]) *A **plabic graph** is an undirected planar graph G drawn inside a disk with the following properties:*

- *n boundary vertices on the boundary of the disk labelled clockwise with $1, \ldots n$,*
- *some number of internal vertices, which are coloured either black of white, and*
- *each boundary vertex is incident to a single edge; if this edge is incident to a leaf, the leaf is called a **lollipop**.*

From associating ⅃-diagrams to plabic graphs, Postnikov showed that what we really want to be looking at are equivalence classes of **reduced plabic graphs**, which are specific types of plabic graphs that can be obtained via certain local moves (transformations) on plabic graphs. That is, these equivalence classes also index the cells of the positive Grassmannian! In particular, like ⅃-diagrams, reduced plabic graphs also see the dimension of the cell, via the number of faces of the graph minus one. We refer the reader to §11–13, 20 of [Pos06] and §14 of [LPW20] for a more thorough treatment of plabic graphs.

Continuing with our running example in bijection with the same positroid, we present the reduced plabic graph in bijection with the ⅃-diagram D from Example 4.2.7.

Example 4.2.10 — *In Figure 4.2, we draw the reduced plabic graph G associated to Examples 4.2.4, 4.2.7 in two different ways. The top representation hints at the bijection with ⅃-diagrams, and the bottom representation hints at the connection to on-shell diagrams.*


FIGURE 4.2: A reduced plabic graph G, drawn in two ways.

Very briefly to see that the decorated permutation does indeed line up with the previous examples, follow the labelled boundary vertices turning (maximally) right at every black vertex and left at every white vertex. Some of these paths are drawn on G with the arrows indicating the direction to follow. Lollipops correspond to fixed points and we colour the fixed points accordingly, that is, black lollipops are loops, and white lollipops are co-loops.

Here G is of type $(3,8)$ as G has 8 boundary vertices, and $k = 3$ is computed via

$$k = \#edges - \#black\ vertices - \sum_{white\ vertices\ v} (\deg(v) - 1) = 11 - 2 - 6 = 3.$$

Thus the equivalence class of G indexes the cell S_G of $\mathrm{Gr}_{3,8}^{\geq 0}$, which corresponds to S_π and S_D from Examples 4.2.4, 4.2.7. The dimension of the cell S_G indexed by G is then $\#faces - 1 = 5$, which lines up with the dimension of S_D.

From our three examples, we have that the decorated permutation π from Example 4.2.4, the ⅃-diagram D from Example 4.2.7, and the reduced plabic graph G from Example 4.2.10 (and its equivalence class) are all in bijection with each other and with the same positroid. Thus they index the same positroid cell, which we label by S_π, S_D, or S_G according to which combinatorial object is of interest!

4.3. T-DUALITY ON DECORATED PERMUATIONS

Through the recent extensive study of the combinatorial objects related to positroids, the positive Grassmannian (like its full counterpart) has been shown to have deep connections to many different areas of mathematics and physics.

To name a few, first purely combinatorially, we can study positroids themselves as a special class of matroids. We can also study the numerous related combinatorial objects that arise directly from

positroids and those arising from the many applications. Some examples include oriented matroids, polytopes and polyhedral subdivisions, non-crossing partitions, and lattice paths. On the more algebraic and geometric side, there are connections to cluster algebras and quantum algebras, and naturally we could study the positive Grassmannian in the spirit of Schubert calculus, through varieties, flags, and symmetric functions. Finally, on the physics side there have been applications to KP solitons, types of asymmetric exclusion processes, and scattering amplitudes (via the amplituhedron and Wilson loop diagrams). We refer the reader to [Pos06; Pos18; KW13; CW07; LPW20; PSBW21] and the references therein for further details.

We will highlight and focus on the combinatorics related to one specific application to physics, namely to scattering amplitudes in $\mathcal{N} = 4$ SYM theory. In particular, the story from [LPW20] starts with looking at the images of the positive Grassmannian under two different maps.

(**4.3.1**) **The hypersimplex.**

The first map of interest is the classic moment map, which was used in [Gel+87] to study the Grassmannian and its decomposition into matroids. Let A be a $k \times n$ matrix representing an element of $\mathrm{Gr}_{k,n}$. The **moment map** μ from $\mathrm{Gr}_{k,n}$ to $\mathbb{R}^n$ is defined as

$$\mu(A) := \frac{\sum_{I \in \binom{[n]}{k}} |p_I(A)|^2 e_I}{\sum_{I \in \binom{[n]}{k}} |p_I(A)|^2}.$$

Under this moment map, the object that arises as the image of both the Grassmannian and the positive Grassmannian is the well-known hypersimplex.

Definition 4.3.1 — *For $I \in \binom{[n]}{k}$, let $e_I := \sum_{i \in I} e_i \in \mathbb{R}^n$ where the e_i are the standard basis vectors of $\mathbb{R}^n$. The **(k,n)-hypersimplex** $\Delta_{k,n}$ is the convex hull of all such points e_I,*

$$\Delta_{k,n} := \mathrm{conv}\left\{ e_I \; : \; I \in \binom{[n]}{k} \right\}.$$

Note that $\Delta_{k,n}$ lives inside $\mathbb{R}^n$ and is of dimension $n - 1$.

As the hypersimplex is a convex polytope, a natural question that arises is how can one triangulate or subdivide this polytope using the moment map and in relation with the positroid cells that decompose the positive Grassmannian?

The start to an answer to this question lies in objects called **positroid polytopes**, which are matroid polytopes where the matroid is a positroid. Analogous to how hypersimplex is defined, given a matroid we can naturally define a polytope using its bases. That is, for a positroid M on $[n]$ as defined by its bases, its positroid polytope Γ_M is

$$\Gamma_M := \mathrm{conv}\{e_I \; : \; I \in M\} \subset \mathbb{R}^n.$$

In particular, why we care about these polytopes is because it turns out that positroid polytopes and the closure of the image of positroid cells under the moment map coincide!

Theorem 4.3.2 — (Proposition 7.10 of [TW15])
Let M be a positroid and S_M its associated positroid cell. Then $\Gamma_M = \overline{\mu(S_M)}$.

Now the question becomes, when does a collection of positroid polytopes (with positroid cells) triangulate the hypersimplex? Here we want the positroid polytopes to be **generalized triangles** of $\Delta_{k,n}$, which restricts to positroid cells of dimension $n-1$ and where μ is injective on the cells. This is one of the questions that [LPW20] looked at.

(**4.3.2**) **The amplituhedron.**

On the other side of the story, as part of an ongoing program on reformulating the Feynman diagram approach in quantum field theory which culminated in the book [AH+16], a surprising connection arose between the work of physicists in the context of scattering amplitudes in $\mathcal{N} = 4$ SYM theory and work on the positive Grassmannian. What the physicists described as **on-shell diagrams**, which are analogs to Feynman diagrams, turned out to be exactly the **plabic graphs** as described by Postnikov [Pos06] (as defined in Section 4.2.3) and in relation to positroid cells.

Motivated by this connection as well as observations on the BCFW recursion relations in tree-level amplitudes, Arkani-Hamed and Trnka [AHT14] introduced a new geometric object called the amplituhedron in hopes of encoding these tree-level $\mathcal{N} = 4$ SYM amplitudes through its "volume".

Definition 4.3.3 — (§9 of [AHT14]) *For $k + m \leq n$, the (tree) **amplituhedron** $\mathscr{A}_{n,k,m}(Z)$ is defined as the image of the positive Grassmannian under the map*

$$\tilde{Z} : \mathrm{Gr}^{\geq 0}_{k,n} \longrightarrow \mathrm{Gr}_{k,k+m}$$
$$A \longmapsto AZ$$

where A is a $k \times n$ matrix representing an element in $\mathrm{Gr}^{\geq 0}_{k,n}$, and Z is a totally positive $n \times (k+m)$ matrix.

Note that $\mathscr{A}_{n,k,m}$ lives inside the Grassmannian $\mathrm{Gr}_{k,k+m}$ and has full dimension km. We will focus on the $m = 2$ case which is considered a toy-model for the more physically relevant $m = 4$ case. However, we note that there are other special cases which also have interesting connections to combinatorics. As an example, when Z is a square $n \times n$ matrix, we recover the positive Grassmannian $\mathrm{Gr}^{\geq 0}_{k,n}$. When $k = 1$ and $m = 2$, $\mathscr{A}_{n,1,2}$ is a polygon in $\mathbb{P}^2$, and more generally $\mathscr{A}_{n,1,m}$ is a cyclic polytope living inside $\mathbb{P}^m$.

While not a polytope, the same question from the hypersimplex can be asked about the amplituhedron: based on the positroid stratification of the positive Grassmannian, how can one "triangulate" the amplituhedron? Relating back to scattering amplitudes, these triangulations should then be related to the different decompositions of the amplitude in the on-shell representation. Some aspects of these questions were previously looked at in [KWZ20].

For the amplituhedron, the objects of interest are **Grasstopes**, which are defined to be the closure of the image of positroid cells under the $\tilde{Z}$ map. The term as used in [PSBW21] is the abbreviation of "Grassmann polytopes", Then we can analogously define **generalized triangles** of $\mathscr{A}_{n,k,m}$ as Grasstopes restricted to km-dimensional positroid cells and where $\tilde{Z}$ is injective on the cells. Once again, one could ask: when does a collection of Grasstopes triangulate the amplituhedron?

(**4.3.3**) **T-duality.**

For us, one of the interesting results from [LPW20] was based on a different question, one connecting the two stories about triangulations. As the positive Grassmannian has this nice positroid cell decomposition, another natural question to ask is if there is any relationship between the images of the positroid cells under the previous two maps, the moment map and $\tilde{Z}$, and how they would relate back to the full objects. Following these lines of questioning and based on numerical evidence, Lukowski, Parisi, and Williams [LPW20] found a surprising connection between the hypersimplex and the $m = 2$ amplituhedron, which was encapsulated by a very simple map directly on positroid cells that they called T-duality.

The name "T-duality" comes from its correspondence to the T-duality in string theory, which very simply put is a type of equivalence of theories where as one example momentum in a certain string theory is interchanged with winding numbers in its T-dual theory. In particular, this map is named as a nod to being a manifestation of T-duality in the context of the geometry of $\mathcal{N} = 4$ SYM scattering amplitudes and the $m = 2$ amplituhedron.

Definition 4.3.4 — (Definition 5.1 of [LPW20]) *The **T-duality** map from loopless decorated permutations on* $[n]$ *to co-loopless decorated permutations on* $[n]$ *is defined as*

$$\pi \longmapsto \hat{\pi}$$
$$(a_1, a_2, \ldots, a_n) \longmapsto (a_n, a_1, \ldots, a_{n-1})$$

*where the permutations are written in one-line notation, and any fixed points in $\hat{\pi}$ are declared to be loops. That is, for a given loopless π, we have $\hat{\pi}(i) = \pi(i - 1)$ where all fixed points are loops, and we call $\hat{\pi}$ the **T-dual** decorated permutation.*

Lemma 4.3.5 — (Lemma 5.2, Proposition 5.17 of [LPW20]) *The T-duality map $\pi \mapsto \hat{\pi}$ is a bijection between loopless decorated permutations on* $[n]$ *with $k + 1$ anti-excedances and co-loopless decorated permutations on* $[n]$ *with k anti-excedances.*

Equivalently, the T-duality map is a bjection between loopless positroid cells S_π of $\mathrm{Gr}_{k+1,n}^{\geq 0}$ and co-loopless positroid cells $S_{\hat{\pi}}$ of $\mathrm{Gr}_{k,n}^{\geq 0}$. Furthermore, we have the following dimensional relationship

$$\dim(S_{\hat{\pi}}) - 2k = \dim(S_\pi) - (n - 1),$$

where in particular, if $\dim(S_\pi) = n - 1$ then $\dim(S_{\hat{\pi}}) = 2k$.

While this dimensional relationship was proven to exist between T-dual loopless cells of $\mathrm{Gr}_{k+1,n}^{\geq 0}$ and co-loopless cells of $\mathrm{Gr}_{k,n}^{\geq 0}$, it was not clear where it was coming from. The need to understand this relationship motivated the question of viewing T-duality as a map on ⅃-diagrams, in which one very easily sees the dimension of the positroid cells they index. This problem is the focus of Chapter 5.

The magic in this very simple shift map on decorated permuations is that it was exactly what was needed to go between triangulations of the $(n - 1)$-dimensional hypersimplex $\Delta_{k+1,n}$ and of the $2k$-dimensional $m = 2$ amplituhedron $\mathscr{A}_{n,k,2}$! This duality then explained the mysterious appearance of $\binom{n-2}{k}$ in the number of cells needed to triangulate both these objects. In [LPW20], the connection was first shown for a specific type of triangulation.

Proposition 4.3.6 — (Theorem 6.5, Proposition 6.6 of [LPW20]) *The T-duality map provides a bijection between BCFW triangulations (resp. dissection) of the hypersimplex $\Delta_{k+1,n}$ and BCFW triangulations (resp. dissection) of the amplituhedron $\mathscr{A}_{n,k,2}$.*

More specifically, a collection $\{S_\pi\}$ of cells of $\mathrm{Gr}_{k+1,n}^{\geq 0}$, as constructed in [LPW20], gives a triangulation (resp. dissection) of $\Delta_{k+1,n}$ if and only if the collection $\{S_{\hat{\pi}}\}$ of cells of $\mathrm{Gr}_{k,n}^{\geq 0}$ gives a triangulation (resp. dissection) of $\mathscr{A}_{n,k,2}$.

Furthermore, if S_π is a generalized triangle of $\Delta_{k+1,n}$, then $S_{\hat{\pi}}$ is a generalized triangle of $\mathscr{A}_{n,k,2}$.

Very recently, Parisi, Sherman-Bennett, and Williams [PSBW21] extended the work done in [LPW20], building on the theory of generalized triangles, and were able to prove the main conjecture in [LPW20] which strengthens the above statement to all triangulations and dissections. Their proof utilized looking at the T-duality map via plabic graphs and plabic tilings.

Theorem 4.3.7 — (Conjecture 6.9 of [LPW20], Theorem 11.6 of [PSBW21]) *Let $\{S_\pi\}$ be a collection of cells of $\mathrm{Gr}_{k+1,n}^{\geq 0}$. Then this collection gives a triangulation (resp. dissection) of $\Delta_{k+1,n}$ if and only if the collection $\{S_{\hat{\pi}}\}$ of cells of $\mathrm{Gr}_{k,n}^{\geq 0}$ gives a triangulation (resp. dissection) of $\mathscr{A}_{n,k,2}$.*

A natural question that arises from T-duality (Definition 4.3.4) is: what does this map look like diagrammatically, that is, on the level of ⅃-diagrams? How are the ⅃-diagrams related?

Recall from Section 4.2.2 that given a decorated permutation π on $[n]$ with k anti-excedances, we get an associated ⅃-diagram D of type (k, n) where:

- D has shape λ which fits in a $k \times (n - k)$ rectangle,
- co-loops are rows of all 0's, while loops are columns of all 0's,
- $\dim(S_D)$ is the number of $+$'s in D, where S_D is the positroid cell indexed by D, and
- anti-excedances of π label the rows of the pipe dream associated to D (including empty rows).

The direction of T-duality we will look at is $\hat{\pi} \mapsto \pi$, where $\hat{\pi}$ is a co-loopless permutation on $[n]$ with k anti-excedances and π is a loopless permutation on $[n]$ with $k + 1$ anti-excedances. On ⅃-diagrams, we are looking at the map from $\hat{D} \mapsto D$, where $\hat{D}$ is the ⅃-diagram associated to $\hat{\pi}$ and D is the ⅃-diagram associated to π. As cells, recall from Section 4.3.3 that $S_{\hat{\pi}}$ is a co-loopless cell of $\mathrm{Gr}_{k,n}^{\geq 0}$ and S_{π} is a loopless cell of $\mathrm{Gr}_{k+1,n}^{\geq 0}$ where their dimensions relate via $\dim(S_{\hat{\pi}}) - 2k = \dim(S_{\pi}) - (n - 1)$. One advantage of looking at this map on ⅃-diagrams, as we will see, is how simple it becomes to relate the dimensions through counting the $+$'s on either side.

We start with an explicit algorithm for constructing D given a $\hat{D}$, with an example given at the end in Section 5.1.5, and then move to a more visual perspective that showcases the structure of the $+$'s of D. We end with proving that the algorithm does give the correct ⅃-diagram, as given by π.

5.1. The bijection on the level of Le diagrams

(**5.1.1**) **Notation.**

Throughout this chapter, we will refer to the rows and columns of a ⅃-diagram D by the same labelling as that which gives the associated decorated permutation (i.e from its pipe dream). Recall, we are viewing the SE border of the shape λ as a lattice path of n steps from the top-right corner to the bottom-left corner of the $k \times (n - k)$ rectangle. The edges of this path are then labelled from 1 to n. Boxes in D, and the $k \times (n - k)$ rectangle, will then be referred to by their coordinates (i, j) under this labelling. A box is considered as "existing" if it is a valid box to be filled within the shape λ. Note that all such valid boxes have $i < j$. An example of this notation is given in Example 5.1.1. Finally since we are dealing with two ⅃-diagrams, one on each side of the map, we refer to the "corresponding" row/column as the row/column with the same label on the opposite side of the map.

We will also order boxes in columns from top to bottom and in rows from right to left, in accordance with labels going from smallest to largest. For rows/columns, having $i < j$ would mean i is to the left

and/or above j. For boxes, "first" refers to right/top-most in a row/column, "last" refers to left/bottom-most in a row/column and "next" refers to the next box to the left/below in a row/column.

Example 5.1.1 — *Going back to Example 4.2.7, we have the following labelling of the ⅃-diagram D:*
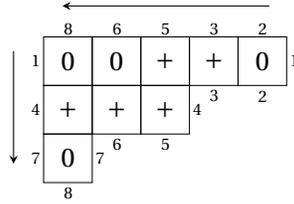
where there are +'s in boxes $(1,3)$, $(1,5)$, $(4,5)$, $(4,6)$, and $(4,8)$. Notice box $(4,3)$ does not exist since it is outside of the shape λ of D. By the nature of the labelling, we also never have boxes of the form (i,i). The arrows indicate the direction of the ordering of boxes, going from first to next to last.

(**5.1.2**) **Preliminaries and shape of** D**.**

We start with a co-loopless decorated permutation $\hat{\pi}$ on $[n]$ with k anti-excedances, which we denote in two-line notation as

$$\hat{\pi} = \left(\begin{array}{cccc} 1 & 2 & \cdots & n \\ a_n & a_1 & \cdots & a_{n-1} \end{array} \right).$$

Let $\{b_1,\ldots,b_k\}$ be the k anti-excedances of $\hat{\pi}$ ordered such that $b_1 < \cdots < b_k$ (recall that b_u is an anti-excedance if $\hat{\pi}^{-1}(b_u) > b_u$, where note we can't have $\hat{\pi}(b_u) = \bar{b}_u$ since $\hat{\pi}$ is co-loopless). In particular, let $i_u = \hat{\pi}^{-1}(b_u)$ be the position of b_u. Then we have $b_u = \hat{\pi}(i_u) = a_{i_u-1}$ for $1 \le u \le k$.

We denote the associated ⅃-diagram by $\hat{D}$ with shape $\hat{\lambda}$. Since $\hat{\pi}$ is co-loopless, we have that every row in $\hat{D}$ has ≥ 1 +'s. In particular,

$$\hat{\lambda} = \left(\hat{\lambda}_1, \cdots, \hat{\lambda}_k \right), \quad \text{where } \hat{\lambda}_u = (n-k) - (b_u - u) \text{ for } 1 \le u \le k$$

and every $\hat{\lambda}_u \ge 1$. We also have that the rows of $\hat{D}$ are thus labelled by these b_u, as in Figure 5.1.
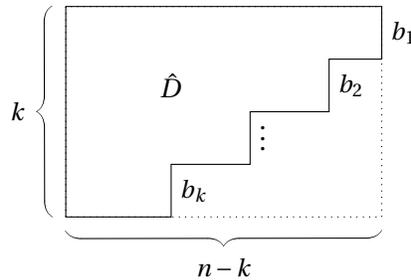
FIGURE 5.1: The shape and row labels of ⅃-diagram $\hat{D}$ associated to $\hat{\pi}$. Here we have illustrated when $b_1 = 1$. When $b_1 \ne 1$, there are columns of size 0 labelled $1,\ldots,b_1 - 1$ (these would be loops), indicated by a horizontal line extending the top (north) border of the diagram. Note that there are no rows of size 0 as $\hat{\pi}$ is co-loopless.

We want to give an algorithm that produces a ⅃-diagram D of type $(k+1, n)$ and dimension $\dim(S_{\hat{D}}) - 2k + (n-1)$ with associated decorated permutation π of $[n]$ given by

$$\pi = \begin{pmatrix} 1 & 2 & \cdots & n \\ a_1 & a_2 & \cdots & a_n \end{pmatrix},$$

which has $k+1$ anti-excedances and is loopless. The loopless condition means that D has to have ≥ 1 +'s in each of the $n - (k+1)$ columns.

First, by looking at T-duality as a map on permutations, we can already determine the shape λ of D which only cares about the anti-excedances of π. Consider the following:

- a_n is not an anti-excedance of $\hat{\pi}$ since $\hat{\pi}$ is co-loopless, $\hat{\pi}(1) = a_n$ and $1 \not> a_n$. In particular, a_n is the label of a column in $\hat{D}$.

- Under T-duality, b_u for $1 \leq u \leq k$ stays an anti-excedance as $i_u > b_u = \hat{\pi}(i_u) = \pi(i_u - 1)$. Since π is loopless, we have either $\pi(i_u - 1) < i_u - 1$ or $\pi(i_u - 1) = i_u - 1$ where $i_u - 1$ must be a co-loop. In either case, b_u is an anti-excedance of π.

- a_n is always an anti-excedance of π since π is loopless.

- There are no other anti-excedances of π since $\pi(i) = \hat{\pi}(i+1) \geq i+1 > i$ for all $1 \leq i < n$ with $i + 1 \notin \{i_1, \ldots, i_k\}$.

Based on these observations, we have that $\{b_1, \ldots, b_k\} \cup \{a_n\}$ are the $k+1$ anti-excedances of π. In particular, the labels of the rows of D (including rows of length 0) are thus exactly the same as $\hat{D}$, with the addition of a_n. Then, the shape of D can be constructed from $\hat{D}$ by removing the column labelled a_n and inserting in a row labelled a_n in the appropriate position, maintaining the order of the labels of the new boundary lattice path, see Figure 5.2.
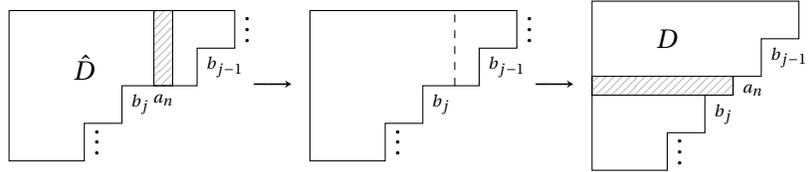
FIGURE 5.2: Steps to construct the shape of D from $\hat{D}$. Remove column a_n from $\hat{D}$ and insert a row labelled a_n where the dashed line is, making sure that the new boundary path is in the correct order. Here j is the index such that $b_{j-1} < a_n < b_j$.

The shape λ of D is then (including 0 sized parts)

$$\lambda = (\lambda_1, \cdots, \lambda_{k+1}) \quad \text{where} \quad \lambda_u = \begin{cases} \hat{\lambda}_u - 1 & 1 \leq u \leq j-1, \\ n - (k+1) - (a_n - j) & u = j, \\ \lambda_u = \hat{\lambda}_{u-1} & j+1 \leq u \leq k+1, \end{cases}$$

and j is the index such that $b_{j-1} < a_n < b_j$. If $a_n > b_u$ for all $1 \leq u \leq k$, then let $j = k+1$. Here we have $0 \leq \lambda_u \leq n - (k+1)$ for all $1 \leq u \leq k+1$, and at most $k+1$ non-zero parts, as needed. Thus the order of rows (and anti-excedances) of D is $\{b_1, \ldots, b_{j-1}, a_n, b_j, \ldots, b_k\}$, as in Figure 5.3. Note that we always have either $b_1 = 1$ or $a_n = 1$, and thus the first row of D will always be labelled with 1 and is a full row of size $\lambda_1 = n - (k+1)$ (which also must hold as π is loopless).
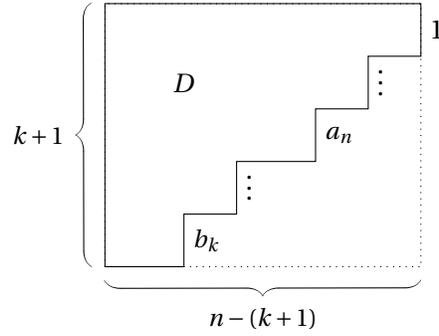
$\mathrm{F}\,\mathrm{I}\,\mathrm{G}\,\mathrm{U}\,\mathrm{R}\,\mathrm{E}$ 5.3: The shape and row labels of ⅃-diagram D associated to π. Here we have illustrated when $b_1 = 1, a_n \neq 1, n$. When $a_n = n$, we would have a row of size 0 labelled by n (and thus is a co-loop) which would be indicated by a vertical line extending the west border of the diagram. Note that there are no columns of size 0 since π is loopless.

(**5.1.3**) **The algorithm.**

Now that we have the shape of D, we give an explicit algorithm for filling in D such that we get the ⅃-diagram with associated decorated permutation π. We will fill D row by row, from right to left, based on the corresponding row and rows below in $\hat{D}$. For row a_n, since a_n was a column in $\hat{D}$, we will consider its "corresponding row" in $\hat{D}$ as a row of all 0's of the same length as in D and placed in the same position as D (in-between rows b_{j-1} and b_j so that the order of labels of the SE border is maintained).

Let L_u be the column containing the leftmost $+$ in each row labelled b_u of $\hat{D}$ (which is in box (b_u, L_u)) for $1 \leq u \leq k$. Let W_u be the row directly below row b_u, and let W_n be the row directly below a_n, if they exist. Otherwise, for the last row b_k, define $W_k = n + 1$, or if a_n is the last row, then define $W_n = n + 1$. Concretely we have

$$W_u = \begin{cases} b_{u+1} & 1 \leq u < k, u \neq j - 1 \\ a_n & u = j - 1 \\ b_{j+1} & u = n, j \neq k + 1 \\ n+1 & u = k, j \neq k + 1 \text{ or } u = n, j = k + 1 \end{cases}$$

In particular, $W_u - 1$ is the column right before the next row below, or for the last row, $W_u - 1 = n$. Finally, we say that a 0 is **restricted** if there is a $+$ in a box to its left, in the same row, and **unrestricted** otherwise. In Example 5.1.1, the only restricted 0 is in box $(1, 2)$.

There are three different row types to consider for D:

(I) Rows b_u such that the box (b_u, a_n) either does not exist, or does not have a $+$ in $\hat{D}$.

(II) Rows b_u such that the box (b_u, a_n) has a $+$ in $\hat{D}$.
   Note: These rows will always be above row a_n.

(III) Row a_n.

We start by outlining the steps for row type (I); the algorithm for row types (II) and (III) will be slight modifications based on these steps. Note that row types (I) and (III) actually follow the same steps if we consider the "leftmost" $+$ in the "corresponding row" to a_n in $\hat{D}$ as occurring in column $L_n > n$.

Algorithm for filling in rows of D **— Main steps, and for row type (I)**

Fill boxes in row b_u from right to left, if they exist, as follows:

Step 1. For columns $b_u + 1$ to $W_u - 1$, fill boxes in row b_u with $+$'s,
   i.e. boxes $(b_u, b_u + 1), \ldots, (b_u, W_u - 1)$ are filled with $+$'s.

Step 2. For columns $W_u + 1$ to $L_u - 1$, fill boxes in row b_u with $+$'s under the conditions defined below.
   Otherwise, fill the boxes with 0's.
   i.e. fill boxes $(b_u, W_u + 1), \ldots, (b_u, L_u - 1)$ with $+$'s if they satisfy the conditions.

Step 3. Starting from column L_u, fill the rest of the boxes in row b_u with 0's,
   i.e. fill boxes (b_u, L_u) and leftwards to the end of the row with 0's.
   Note: if $W_u > L_u$, then Step 1 only fills boxes until column $L_u - 1$ and Step 2 is skipped.

Note that the algorithm can be run in parallel for each row.

Step 2 Conditions **—** Fill a box in row b_u at (b_u, ℓ) with a $+$ if:

(i) In $\hat{D}$, there is a $+$ in box (b_u, ℓ).

(ii) In $\hat{D}$, there is a $+$ in some row below, say at (b_m, ℓ) where $m > u$, such that there are only unrestricted 0's in column ℓ in-between rows b_u and b_m. For example, the circled $+$'s in Figure 5.4 satisfy this condition. Note: In the case that $m = u + 1$, the condition holds trivially. Another way to phrase this condition: Look for $+$'s in column ℓ in rows below b_u where all rows in-between have no $+$'s to the left of column ℓ (their leftmost $+$ is before column ℓ).

(iii) In $\hat{D}$, column ℓ only has unrestricted 0's below row b_u. For example, the shaded columns in Figure 5.4 satisfy this condition. In other words in column ℓ, all the rows below b_u has their leftmost $+$ before column ℓ, i.e. their last $+$ has already passed.
   Note: This is a special case of (ii).
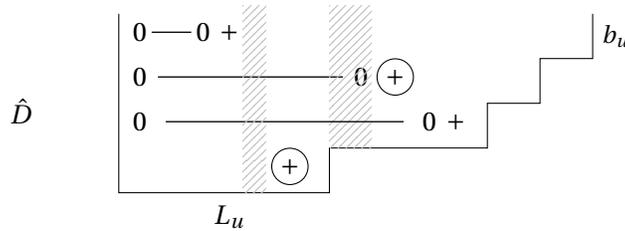
FIGURE 5.4: Example of conditions in Step 2(ii) and (iii). For row b_u, the columns of the two circled $+$'s in this $\hat{D}$ satisfy condition (ii) as all boxes above the $+$ (until row b_u) are filled with unrestricted 0's. Thus in D, a $+$ would be filled in row b_u in those two columns. The column of the non-circled $+$ not in row b_u would fail this condition because of the row directly above it. The shaded columns indicate the columns that satisfy condition (iii) since all the boxes below row b_u are filled with unrestricted 0's.

Modifications for row types (II) and (III) —

For row type (II):

Steps 1 & 2. For columns $b_u + 1$ until row a_n, follow Steps 1 & 2 of row type (I),
i.e. fill boxes $(b_u, b_u + 1), \ldots, (b_u, a_n - 1)$ as in row type (I).

Step 3. Starting from column $a_n + 1$, fill the rest of the boxes in row b_u following only Step 2(i),
i.e. fill boxes $(b_u, a_n + 1)$ and leftwards until the end of the row using Step 2(i).
Note: These boxes have the exact same filling as in the corresponding row in $\hat{D}$.

For row type (III):

Steps 1 & 2. Follow Steps 1 & 2 of row type (I) for the whole row a_n.
Recall, the "corresponding row" to a_n in $\hat{D}$ is a row of all 0's of the same length, in the same position as in D. Think of the leftmost $+$ in this row as occurring at (a_n, L_n) for some $L_n > n$. Under this convention, replace all instances of b_u with a_n and L_u with L_n in the main steps.
Note: Step 2(i) never applies for this row.

From these steps, we can already start to see some structure to these diagrams D filled through this algorithm. Later in Section 5.2, we will give a different perspective on how to build these diagrams that showcases the structure of the $+$'s.

Some first observations are:

– Step 1 tells us that there will always be a string of $+$'s at the beginning of a row (until the next row below) in one of two ways:

FIGURE 5.5: The beginning of a row using Step 1. On the left, we have the case when $L_u > W_u$ and so $+$'s are filled in columns $b_u + 1 \leq \ell \leq W_u - 1$. On the right, we have the case when $L_u < W_u$, in which Step 2 is skipped, and thus $b_u + 1 \leq \ell \leq L_u - 1$.

– In particular, Step 1 also tells us that the bottom-most (non-empty) row of D will be either:

FIGURE 5.6: The last row using Step 1. On the left, we have the case when $b_k > a_n$ is the last row, and so $+$'s are filled until column $L_u - 1$, with 0's filling the rest from Step 3. On the right, we have the case when $a_n > b_k$, and Step 1 fills the whole row with $+$'s.

- Step 2(ii) and (iii) tells us that for row type (III), there will always be a string a $+$'s at the left end of this row:

$$a_n \;\boxed{+ \;\cdots\; +}$$

- Step 2(iii) also tells us that the string of $+$'s filled by satisfying this condition always comes after (is to the left of) a $+$ that is filled based on Step 2(ii).

- Step 3 of row types (I) and (II) tells us that there will always be a string of 0's at the left end of row b_u after column L_u:

$$b_u \;\boxed{0 \!-\!\!-\! 0}$$
$$L_u$$

Finally, what's interesting about this algorithm is that for any particular row, only a part of $\hat{D}$ is looked at, namely it looks recursively at the row itself and the first rows below for which the leftmost $+$ has not yet been passed.

(**5.1.4**) D **is indeed a ⅃-diagram.**

To even be able to consider this algorithm, we first verify that under this filling D avoids the ⅃-configuration and thus is a valid ⅃-diagram.

Theorem 5.1.2 — *Under this filling, D is a ⅃-diagram of type $(k+1, n)$.*

PROOF. We show that if a ⅃-configuration occurs in D, then one must occur in $\hat{D}$ as well!

Without loss of generality, suppose in D we have the following

$$\boxed{+} \leftarrow \text{row i}$$
$$\vdots$$
$$\boxed{+} \cdots \boxed{0} \leftarrow \text{row j}$$
$$\uparrow \qquad \uparrow$$
$$\text{col. m} \quad \text{col. } \ell$$

where the boxes indicated by the dots are filled with 0's. There are two cases for row j,

 (i) either there's a corresponding row j in $\hat{D}$, or
(ii) $j = a_n$.

First, based on our observations from Steps 1 and 3, the three boxes in this configuration must have been filled in Step 2 for rows i and j, regardless of row type. In particular, that means row j is not the last row, and furthermore there must be boxes in columns $\ell, \ldots, m$ below row j in both D and $\hat{D}$.

Looking at the $+$ at (j, m) in D, since it was filled based on a condition in Step 2, in each case we have:

(i) The last $+$ in the corresponding row in $\hat{D}$ must be to the left of or at column m, and thus in $\hat{D}$ there is a $+$ at some (j, L) where $L \geq m$.

(ii) In this case, the box must have been filled by Step 2(ii), and thus in $\hat{D}$, there is some first row b below j such that its last $+$ is to the left of or at column m, and thus there is a $+$ at (b, L) where $L \geq m$. That is, all rows in-between j and b must have boxes in column m filled with unrestricted 0's i.e. their last $+$ is to the right of m.
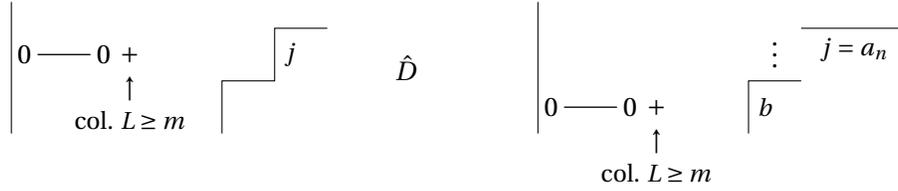
FIGURE 5.7: The two cases for j in $\hat{D}$ based on a $+$ at (j, m) in D. On the left, we have case (i), and on the right, we have case (ii). Here b is the first row below $j = a_n$ where all boxes in column m in-between row b and column j are filled with unrestricted 0's.

Next, looking at the 0 at (j, ℓ), in each case we have:

(i) Because of Step 2(i) and the last $+$ of row j in $\hat{D}$ is at some column $L \geq m > \ell$, we know that there is no $+$ in $\hat{D}$ at (j, ℓ), and thus it is filled with 0.

(ii) Since we are assuming boxes $(j, \ell + 1), \ldots, (j, m - 1)$ are also filled with 0's in D, we must have that all rows in-between j and b in $\hat{D}$ have unrestricted 0's filled in those columns. As otherwise Step 2(ii) would have placed a $+$ in D somewhere in those boxes. But then this also means that in $\hat{D}$, (b, ℓ) must be filled with a 0 since otherwise Step 2(ii) would be satisfied, placing a $+$ at (j, ℓ) in D.

Finally we look at the $+$ at (i, ℓ). The $+$ at (j, m) in D tells us that Step 2(iii) could not have been the reason for this $+$ since in $\hat{D}$, the last $+$ of row j or row b, for each case respectively, have not passed yet. Thus this $+$ must have come from satisfying Step 2(i) or (ii) for row i (regardless of type) which means in $\hat{D}$ in column ℓ, there must be a $+$ in rows i or below. The 0 at (j, ℓ) in D then tells us that in $\hat{D}$, there cannot be any $+$'s in rows below j in column ℓ or else Step 2(ii) would have been satisfied for (j, ℓ). Then, regardless of which case we are in, there must be a $+$ in $\hat{D}$ in column ℓ above row j.

Putting all these together, notice in either case we have constructed a ⌐-configuration occurring in $\hat{D}$. But this cannot happen as $\hat{D}$ is a ⌐-diagram! Thus the ⌐-configuration cannot occur in D and D is indeed a valid ⌐-diagram. Furthermore, from the shape λ of D, we get that D is a ⌐-diagram of type $(k + 1, n)$ as needed. ∎

(**5.1.5**) **An example.**

To illustrate this algorithm, we work through an example step by step.

Suppose we had the following ⅃-diagram $\hat{D}$ of type $(4, n)$ and dimension $2k = 8$, where all unmarked boxes are filled with 0's. The circled $+$'s indicate the last $+$'s in each row.
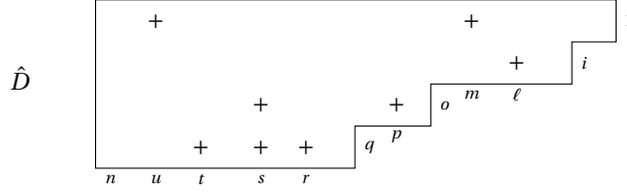
First, since in $\hat{D}$ we have that $\hat{\pi}(1) = a_n = m$, the shape of D is
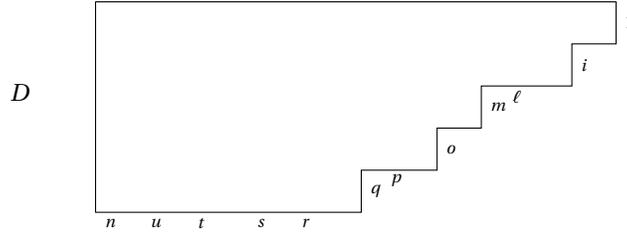
Here we have the following variables,

$$
\begin{array}{lllll}
b_1 = 1 & b_2 = i & a_n = m & b_3 = o & b_4 = q \\
L_1 = u & L_2 = \ell & L_n > n & L_3 = s & L_4 = t \\
W_1 = i & W_2 = a_n & W_n = o & W_3 = q & W_4 = n + 1
\end{array}
$$

where recall that b_u are the rows of $\hat{D}$, L_u is the column of the last $+$ in row b_u (with the last $+$ for a_n thought of as at some $L_n > n$) and W_u is the next row below b_u (or $n + 1$ if its the last row).

To fill D, we follow the algorithm row by row, starting with row 1. As row 1 has a $+$ in column $a_n = m$ in $\hat{D}$, it is of row type (II) and we obtain the filling:
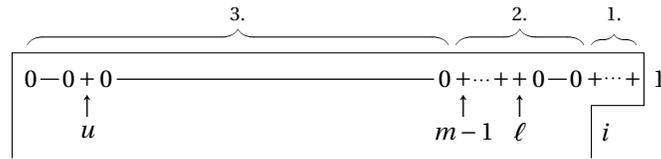
1. From Step 1, we fill boxes $(1, 2), \ldots, (1, i - 1)$ with $+$'s as the last $+$ has not yet passed ($L_1 = u$) and the next row below is row i.

2. From Step 2, we fill boxes $(1, i + 1), \ldots, (1, m - 1)$ with $+$'s according to the conditions, and 0's otherwise. We stop at column $m - 1$ since $a_n = m$, and we are following row type (II).

   (i) Row 1 has no $+$'s in these boxes in $\hat{D}$ so Step 2(i) is not satisfied.

   (ii) Step 2(ii) fills a $+$ in box $(1, \ell)$, as the $+$ at (i, ℓ) in $\hat{D}$ satisfies this condition trivially since there are no rows in-between rows 1 and i.

   (iii) Step 2(iii) then fills $+$'s in boxes $(1, \ell + 1), \ldots, (1, m - 1)$, since the $+$ at (i, ℓ) in $\hat{D}$ was in fact the last $+$ of row i.

3. Finally, Step 3 only fills a $+$ in box $(1, u)$, with the rest filled with 0's, since in $\hat{D}$ there is only one $+$ after column $a_n = m$ which is in column u.

Next, row i does not have a $+$ in column m in $\hat{D}$ and thus is of row type (I). As the last (and only) $+$ of row i is at (i, ℓ) in $\hat{D}$, we have that $W_2 = m > L_2 = \ell$ and so Step 2 is skipped. Then, Step 1 fills boxes $(i, i+1), \ldots (i, \ell - 1)$ with $+$'s, and Step 3 fills the rest of the boxes with 0's, to obtain:

$$0 \overline{} 0 + \cdots + \boxed{i}$$

Moving on to row $m = a_n$ which is of row type (III), recall that we are considering "row" a_n in $\hat{D}$ to be a row of all 0's with the same length and position as in D, and having its last $+$ at some column $L_n > n$.

$$+ \cdots\cdots + + 0 - 0 + 0 \overline{} 0 + 0 - 0 + \cdots + \boxed{m}$$

with arrows pointing to t, s, p, and o.

1. From Step 1, we fill boxes $(m, m+1), \ldots, (m, o-1)$ with $+$'s as the last $+$ has not yet passed $(L_n > n)$ and the next row below is row o.

2. From Step 2, we fill the rest of the boxes in the row with $+$'s according to the conditions and 0's everywhere else. As we are in row type (III), we skip Step 3.

   (i) Step 2(i) never applies for this row.

   (ii) Step 2(ii) fills a $+$ in boxes (m, p), (m, s) and (m, t). These columns correspond to $+$'s in $\hat{D}$ in rows below m such that all the boxes above are unrestricted 0's.

   (iii) Step 2(iii) then fills $+$'s in boxes from $(m, t+1)$ to the end of the row, since the $+$ at (q, t) in $\hat{D}$ is the leftmost $+$ of all rows below m and thus all columns after t are filled with unrestricted 0's below m.

After row a_n, all the rows below will be of type (I). For row o we get the filling:

$$0 \overline{} 0\,0 - 0 + 0 - 0 + \cdots\cdots + \boxed{o}$$

with arrows pointing to $s-1$, r, q, p.

Here, Step 1 fills the beginning of row o with $+$'s until the next row below, which is row q, as the last $+$ in row o in $\hat{D}$ is in column $L_3 = s$. Note that we don't consider the $+$ in box (q, p) in $\hat{D}$. Step 2 fills a $+$ only at box (o, r) through condition (ii), and 0's everywhere else until column s. Step 3 then fills the rest of the row, starting from column s, with 0's.

Finally, we have row q. As its the last row, we skip Step 2 (recall we are considering $W_4 = n+1$) and since the last $+$ of row q is in box (q, t) in $\hat{D}$, we have:

$$0 \overline{} 0 + \cdots\cdots\cdots + \boxed{q}$$

with t below.

Putting everything together, we get the following ⅃-diagram D:

$$D$$
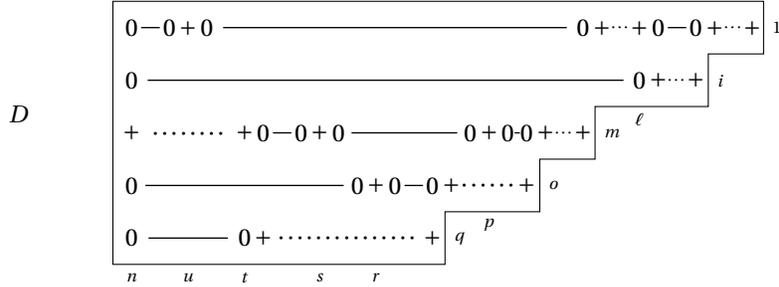
To briefly check that this D is indeed the right diagram for the given $\hat{D}$:

- D is a ⅃-diagram of type $(5, n) = (k+1, n)$ (notice it avoids the ⅃-configuration and has 5 rows).
- Every column has at exactly one $+$ except columns p, r, s, u, which have two $+$'s. Thus D is loopless and we have dimension (recall this is the number of $+$'s)

$$\dim(S_D) = (n-5) + 4 = n - (k+1) + k = n - 1,$$

  as there are $n-5$ columns and 4 columns with an extra $+$. This gives the correct relation since we wanted $\dim S_{\hat{D}} - 2k = \dim(S_D) - (n-1)$, recalling that $\dim(S_{\hat{D}}) = 8 = 2k$.

- One can check that the associated decorated permutation π to D, in two-line notation, is

$$\begin{pmatrix} 1 \cdots i-1 \cdots \ell-1 \cdots m-1 \cdots o-1 \cdots p-1 \cdots q-1 \cdots r-1 \cdots s-1 \cdots t-1 \cdots u-1 \cdots n \\ 2 \quad \ell \quad \textcircled{i} \quad u \quad p \quad s \quad r \quad \textcircled{o} \quad t \quad \textcircled{q} \quad \textcircled{1} \quad \textcircled{m} \end{pmatrix},$$

  where the $\cdots$ denotes $\pi(a) = a+1$ for all a in-between the explicitly written values. Notice π is loopless and has 5 anti-excedances (circled). Shifting the bottom-line one to the right (and wrapping around) exactly gives $\hat{\pi}$, the decorated permutation associated to $\hat{D}$.

To compare with $\hat{D}$, we redraw D to illustrate the structure of the $+$'s (as we will discuss in the next section). All the unmarked boxes are filled with 0's.

$$D \qquad\qquad\qquad \hat{D}$$
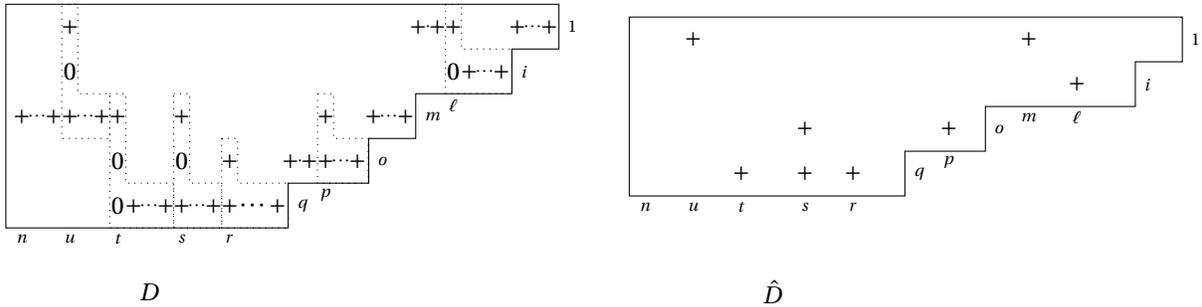
FIGURE 5.8: An example of D vs. $\hat{D}$, where D (left) is filled by the algorithm for the given $\hat{D}$ (right). In D, the dotted lines outlining shapes of $+$'s and 0's are to illustrate the structure of the $+$'s. Comparing to the original $\hat{D}$, notice that there is one of these shapes per column of $+$'s in $\hat{D}$, except column $a_n = m$.

5.2. A MORE VISUAL PERSPECTIVE

While an algorithm is nice to have, it doesn't illuminate the underlying structure of the $+$'s in D, which we are starting to see (Figure 5.8). Perhaps a more enlightening, and surprising, viewpoint is to look at how the $+$'s in $\hat{D}$ give rise to certain shapes, which when glued together fully characterize D (see Theorem 5.2.5). That is, without going through the algorithm, we can reconstruct D using these shapes. Whereas the algorithm can be seen as viewing D through each row, here we are viewing D through each column. Under this perspective, it becomes clearer how a reverse map on ⅃-diagrams would work.

(**5.2.1**) **Building blocks.**

The first aspect of the surprising structure that arose is that D is composed of 2 distinct shapes; the main one being what we will call $\mathbb{L}$-shapes and also strings of $+$'s. In fact, the strings of $+$'s can be considered as a special type of $\mathbb{L}$-shape where there is only the horizontal part.

We will use the same notation as in Section 5.1 for ⅃-diagrams, see Example 5.1.1. Recall that boxes in a ⅃-diagram are referred to by their coordinates (i, j), which denotes row i and column j. We also interchangeably use "first" to mean top/right-most, "last" to mean bottom/left-most, and "passed" to mean in some column to the right of the current column in consideration, with "not yet passed" meaning to the left of or at the current column.

Recall that $\hat{D}$ is a co-loopless ⅃-diagram of type (k, n), shape $\hat{\lambda}$, and with decorated permutation $\hat{\pi}$. We are denoting its rows by $b_1, \ldots, b_k$ and distinguishing the column a_n, which is defined by $\hat{\pi}(1) = a_n$. The algorithm in Section 5.1 is constructing a loopless ⅃-diagram D of type $(k + 1, n)$, shape λ, and with decorated permutation π, as given by T-duality (Definition 4.3.4) for the given $\hat{\pi}$. The rows of D are $\{b_1, \ldots, b_k, a_n\}$, with row a_n inserted as in Figure 5.2. From the algorithm, there are three row types to consider for D: (I) rows with no $+$'s in column a_n, (II) rows with $+$'s in column a_n and (III) row a_n.

To create these $\mathbb{L}$-shapes, we will look at columns $\ell \neq a_n$ in $\hat{D}$ with at least one $+$. As such, there are some facts we need to know about these columns:

– In $\hat{D}$, since every row contains at least one $+$ (as co-loopless) and we have the ⅃-condition, all 0's below a $+$ in column ℓ must be unrestricted, i.e. these rows have their last $+$ to the right of ℓ:
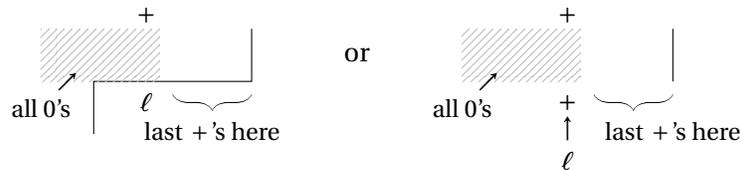
– For $+$'s in column ℓ at (b, ℓ):
  – If $a_n < b$ then b is of row type (I) as (b, a_n) is not a box within the shape $\hat{\lambda}$.
  – If $b < a_n < \ell$ then b is of row type (II) since there is a $+$ at $(1, a_n)$ and so the ⅃-condition forces there to be a $+$ at (b, a_n).
  – If $\ell < a_n$, then b can be of row type (I) or (II).

We refer to $+$'s in $\hat{D}$ that are not a leftmost (last) $+$ in its row as **non-last**.

Theorem 5.2.1 — *For every column $\ell \neq a_n$ in $\hat{D}$ with at least one $+$, in D there is a corresponding shape*


$$s_\ell + t_\ell + 1 \\ +\text{'s} \qquad \begin{array}{l} \boxed{+} \leftarrow \ row\ b_T \\ \quad \boxed{+ \cdots +} \leftarrow \ row\ b_B \end{array}$$

col. ℓ col. m
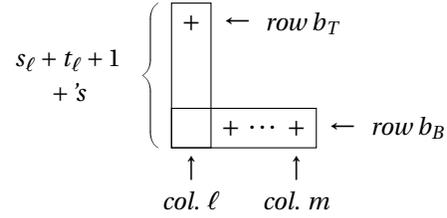
*which we call a $\mathbb{L}$-**shape**, and which has the following properties:*

- *There is always a $+$ in box (b_T, ℓ), with some sequence of 0's and $+$'s below.*
- *There is a string of $+$'s in boxes $(b_B, m), \ldots, (b_B, \ell - 1)$, possibly empty (if $m = \ell$).*
- *In $\hat{D}$, there are s_ℓ non-last $+$'s in column ℓ.*
- *In $\hat{D}$, there are t_ℓ last $+$'s in rows of type (II) in column ℓ.*
- *There is one more $+$, not counted by s_ℓ or t_ℓ, that is in row a_n or in row $b_T \neq f, a_n$ in column ℓ, where f is the row of the topmost $+$ in column ℓ in $\hat{D}$.*
- *In $\hat{D}$, $m - 1$ is either the label of the first column with at least one $+$ to the right of ℓ, or the first row above ℓ, whichever comes first.*

Moreover, we can exactly characterize the $+$'s in the $\mathbb{L}$'s, and these are the only $+$'s in these columns in D.

PROOF. Let $\ell \neq a_n$ be a column in $\hat{D}$ with at least one $+$. Let (f, ℓ) be the first (topmost) $+$ in this column and (g, ℓ) be the last (bottom-most).

First, to determine the horizontal string of $+$'s in the $\mathbb{L}$, we look at $\hat{D}$ around the $+$ at (g, ℓ). Using the first fact from above, if there are boxes below (g, ℓ), there must be a first column call it $m - 1$, to the right of ℓ with at least one $+$. Furthermore, in this column there must be $+$'s in rows g or below. That is all columns in-between $m - 1$ and ℓ and in rows g and below, are filled with only 0's. By the ⅃-condition, these columns are then actually columns of all 0's.

all 0's

$\hat{D}$ row $g \rightarrow$ $+$ $\}$ $+$'s here in col. $m - 1$

ℓ $m - 1$
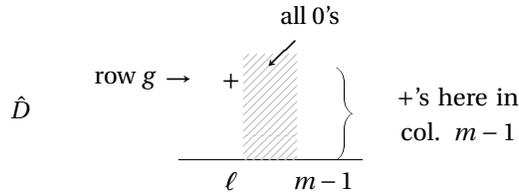
If there are no rows below (g, ℓ), then we let $m - 1 = g$ if (g, ℓ) is the first $+$ in its row, or let $m - 1$ be the column such that $(g, m - 1)$ is the first $+$ to the right of (g, ℓ).

$$\hat{D} \qquad \frac{+\,0\,-\,0\,+}{\ \ell \qquad m-1} \Big|\, g$$

$\uparrow$
either $+$ at $(g, m - 1)$ or let $m - 1 = g$

81
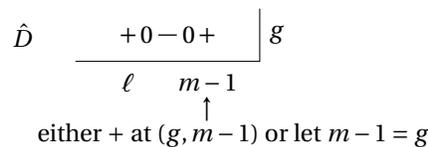
If row g is of type (I) or of type (II) with $\ell < a_n$, then the boxes $(g, \ell - 1), \ldots, (g, m)$ are filled with a string of $+$'s in D by Step 2(iii) if there are rows below (g, ℓ), or by Step 1 if there are no rows below (recall that row type (II) follows the same steps as row type (I) until the column a_n). Thus we let $b_B = g$.

If row g is of type (II) with $g < a_n < \ell$, then notice in D, the row a_n will be inserted in-between g and ℓ. In this case, we have that there will always be a first column $m - 1$ to the right of ℓ with at least one $+$ in $\hat{D}$, as $m - 1$ is at least a_n. Note that we can have $m - 1 = a_n$.
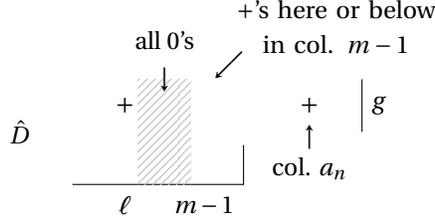
Now we repeat a similar argument as before, except for boxes $(a_n, \ell - 1), \ldots, (a_n, m)$. These are filled with a string of $+$'s in D by Step 1 for row type (III), if a_n becomes the row right above ℓ, or by Step 2(iii) otherwise. Thus we let $b_B = a_n$.

Furthermore, we argue that these are the only $+$'s in columns m to $\ell - 1$ in D. Notice that from the algorithm, for columns like m in $\hat{D}$ filled only with 0's, the only way that a box in this column in D can be filled with a $+$ is through some Step 1 or Step 2(iii) for any row type. So we just need to show that if say box (b_B, m) is filled with a $+$, then no other rows in column m satisfy these two conditions. First note that Step 1 can only affect the very last box of column m and that to have (b_B, m) filled with a $+$ means the last $+$ in row b_B, say at (b_B, L), must not have passed yet ($L > m$).

Right away we see that for rows b below b_B in column m, if (b_B, m) was filled through Step 1, then there's nothing to check as b_B was the last row. If (b_B, m) was filled through Step 2(iii), then all rows b must have their last $+$ to the right of m. But then we cannot be in Step 1 nor Step 2(iii) for (b, m) since the last $+$ in row b has already passed.
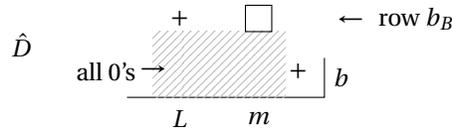
For any rows b above b_B, we can never be in Step 1 since these are not the last row in the column. We also can never satisfy Step 2(iii) for (b, m) since this requires all rows below b to have their last $+$'s to the right of m, which does not hold for row b_B, and now we are done.

Thus we get for the horizontal part of the $\mathbb{L}$ in D:

- $b_B = g$, or $b_B = a_n$ if row g is of type (II) and $g < a_n < \ell$.
- m, possibly equal to ℓ, corresponds to the column such that all columns in-between ℓ and $m - 1$ in $\hat{D}$ are filled with only 0's. That is $m - 1$ is either the first column with at least one $+$ to the right of ℓ, or $m - 1$ is a row.
- For $m \neq \ell$, boxes $(b_B, \ell - 1), \ldots, (b_B, m)$ are filled with $+$'s, and these are the only $+$'s in columns $\ell - 1$ to m. Otherwise if $m = \ell$, there are no boxes in the horizontal part.

Next we look at the vertical part of the $\mathbb{L}$, including box (g, ℓ), which corresponds to the column ℓ. First, using the first fact from above once again, this time for $+$'s in column ℓ, we see that any row b below f with a 0 filled at box (b, ℓ) in $\hat{D}$ must have its last $+$ in some column to the right of ℓ. Note here we are not considering $b = a_n$. Thus for these rows, Step 3 for row types (I) and (II) will fill box (b, ℓ) in D with a 0 as well. For the rows b with a $+$ filled at (b, ℓ) in $\hat{D}$, in D box (b, ℓ) is filled with a $+$ through Step 2(i) for row type (I) if it is a non-last $+$, and Step 2(i) and Step 3 for row type (II), regardless if it is a last $+$ or not. Note, if (b, ℓ) is a last $+$ in row b of type (I), then Step 3 fills this box in D with a 0. This gives $s_\ell + t_\ell$ $+$'s in column ℓ in D, determined exactly by the $+$'s in column ℓ in $\hat{D}$, where s_ℓ is the number of non-last $+$'s in column ℓ of $\hat{D}$ and t_ℓ is the number of last $+$'s in rows of type (II) in column ℓ of $\hat{D}$.

To determine row b_T, we look at $\hat{D}$ above the $+$ at (f, ℓ). As all boxes directly above (f, ℓ) are filled with 0's in $\hat{D}$, from the algorithm, the only way to fill these boxes (b, ℓ) in D with a $+$ is through Step 2(ii).

If row f is type (I) or of type (II) with $\ell < a_n$ then we look for the first row above f for which in $\hat{D}$ its last $+$ has not yet passed. Note that such a row must always exist if f is of type (II) with $\ell < a_n$, and this row will also be of type (II) with $\ell < a_n$. If such a row b exists and it is of type (I) or type (II) with $\ell < a_n$, then Step 2(ii) for row b would place a $+$ at (b, ℓ) in D and we let $b_T = b$. All rows in-between b and f would have had their last $+$ to the right of ℓ and so these rows are filled with a 0 in column ℓ in D.

If no such row exists, then either there are no rows above f and thus $a_n = 1$ (if $f = 1$ then f is of row type (II) with $a_n < \ell$) or all rows above f have their last $+$ to the right of column ℓ and thus $a_n < \ell$. In both cases, f must be of row type (I) and with $a_n < f$ since in the second case, if $f < a_n < \ell$, then by the $\lrcorner$-condition there would be a $+$ at (f, a_n) contradicting our assumption of f's row type. If a row b exists but is of type (II) with $\ell > a_n$, in which case $a_n > b$, then once again because of the $\lrcorner$-condition we actually have $a_n < f$ (as f must be of type (I)). In either case, we get that Step 2(ii) for row a_n will fill box (a_n, ℓ) with a $+$ since all rows in-between a_n and f would have all their last $+$'s already passed and furthermore all these rows are filled with a 0 in column ℓ in D. Thus we let $b_T = a_n$.

If f is of type (II) with $\ell > a_n$, in which case we also have $a_n > f$, then by the $\lrcorner$-condition any rows b above f with $+$'s to the left of column ℓ would have to have a $+$ at (b, a_n) making them of row type (II) as well. The rest of the rows, would have their last $+$ to the right of ℓ. Then Step 3 for row types (I) and (II) would fill all the rows b above f with a 0 at (b, ℓ). Since f is of type (II), Step 2(i) would fill (f, ℓ) with a $+$, and thus we let $b_T = f$.

One detail left over is that for boxes in-between g and b_B in column ℓ, by the $\lrcorner$-condition all these rows must have their last $+$ already passed (or else it contradicts g) in $\hat{D}$ and thus Step 3 for row types (I) and (II) would fill these boxes with a 0.

Thus we get for the vertical part of $\mathbb{L}$ in D:

- There are three cases for b_T. Either b_T is the first row above f, and below a_n if $f > a_n$, with a $+$ to the left of column ℓ in $\hat{D}$, or $b_T = f$ if row f is of type (II) with $f < a_n < \ell$, or $b_T = a_n < f$.
- There is always a $+$ at (b_T, ℓ).
- There is a $+$ in box (b, ℓ) if there is a non-last $+$ at (b, ℓ) in $\hat{D}$, or if there is a last $+$ but b is of row type (II), for rows b from f to g in $\hat{D}$. Thus there is a total of $s_\ell + t_\ell$ $+$'s in boxes $(f, \ell), \ldots, (g, \ell)$ in D, not considering row a_n.
- We always have $b_T \le f$ and $b_B \ge g$, and thus these $+$'s are all within the vertical part of the $\mathbb{L}$.
- Any rows in-between b_T and f in column ℓ must be filled with 0's, and similarly for g and b_B.

Finally, we show that there are exactly $s_\ell + t_\ell + 1$ +'s in column ℓ, in the vertical part of the $\mathbb{L}$. To do this, we look at the different cases of b_B, b_T and how row a_n affects the $\mathbb{L}$. We will show that in fact $b_T < b_B$ and the extra $+$ will be either in a_n or we have $b_T \neq f, a_n$. Doing case analysis on a_n:

- If $a_n < f \leq g < \ell$, then $b_B = g$, and $b_T = a_n$ or b_T is the first row above f, below a_n with a $+$ to the left of column ℓ.
  In either case the extra $+$ is at (b_T, ℓ) and we have $b_T < b_B$.

- If $f < a_n < g < \ell$, then $b_B = g$ and $b_T = f$.
  In this case, Step 2(ii) for row a_n fills (a_n, ℓ) with a $+$ since there is at least one more $+$ in rows below a_n in column ℓ in $\hat{D}$, with all rows in-between filled with unrestricted 0's by the $\lrcorner$-condition.

- if $f \leq g < a_n < \ell$, then $b_B = a_n$ and $b_T = f$.
  Here the extra $+$ is at $(b_B, \ell) = (a_n, \ell)$.

- Lastly if $f \leq g < \ell < a_n$, then $b_B = g$ and $b_T \neq f, a_n$.
  Here the extra $+$ is at (b_T, ℓ).

Thus we are done since these are the only cases, and in each we have $b_T < b_B$ with exactly $s_\ell + t_\ell + 1$ +'s in the vertical part of the $\mathbb{L}$ with a $+$ at (b_T, ℓ).

One final detail to check is that these are the only +'s in column ℓ. Since we always have that $b_T \leq f$ and $b_T \geq g$ with $b_T < b_B$, if there was a $+$ at (a, ℓ) in D, for rows $a < b_T$ or $a > b_B$, we can show this contradicts the choice of b_T or b_B.

For $b < b_T \leq f$, the $+$ at (b, ℓ) must have been filled through Step 2(ii) for row b which means that row b's last $+$ in $\hat{D}$ has not yet passed. This also means that any rows c below $b < c < f$ must have their last +'s to the right of column ℓ. But this contradicts when $b_T \neq f$ since in these cases, b_T has their last $+$ to the left of column ℓ (for $b_T = a_n$ remember we are thinking of row a_n as having last $+$ in a column $> n$). Thus $b_T = f$, which means that f is of row type (II) with $a_n > f$. But because of the $\lrcorner$-condition, all rows above f, in particular b, must also be of type (II) with $a_n > b$, in which case the algorithm would not have filled (b, ℓ) with a $+$. Thus all rows b above b_T must be filled with a 0 at (b, ℓ).

For $b > b_B \geq g$, the $+$ at (b, ℓ) must have been filled through Step 2(iii) for row b. Note, it could not have been filled through Step 1 since by the $\lrcorner$-condition, that would mean there was a $+$ at (b, ℓ) in $\hat{D}$ originally, contradicting g (since row b's last $+$ in $\hat{D}$ must have been to the left of column ℓ and there's a $+$ at (g, ℓ)). Once again, Step 2(iii) means that row b's last $+$ in $\hat{D}$ has not yet passed. By the $\lrcorner$-condition for $\hat{D}$, this would force $b = a_n$ or else we would contradict g, and thus $b_B = g$. However, as there is a box at (b, ℓ), we actually have $\ell > b = a_n > b_B = g$. Looking at our case analysis above, this contradicts the choice of b_B, which should have been a_n. Thus all rows b below b_B must be filled with a 0 at (b, ℓ).

This concludes the proof. ∎

Notice these $\mathbb{L}$-shapes cover all columns from $\hat{D}$, ignoring a_n, with at least one $+$ and the columns of all 0's to the right of these. What's left are the (consecutive) columns of all 0's to the left of a column with at least one $+$ until the next row below or the end of the diagram, or if in-between two rows there are only columns of all 0's. These columns are covered by strings of +'s, which is actually just the horizontal part of an $\mathbb{L}$-shape.

Corollary 5.2.2 — *For consecutive columns of all 0's in $\hat{D}$, in D there is a corresponding string of + 's*

$$\boxed{+ \ \cdots\cdots \ +}$$

and these are the only + 's in these columns.

PROOF. From the proof of Theorem 5.2.1, we saw that this statement holds for consecutive columns of all 0's in $\hat{D}$ in-between two columns with at least one +, or in-between the first column to the left of a row with a least one + and the beginning of that row. What's left are consecutive columns of all 0's in-between two consecutive rows, and there are no columns with at least one + in-between these rows, and those in-between the next row below or the end of the diagram and a column with at least one +. Both of these are really special cases of what we proved in Theorem 5.2.1.

Say we are considering columns with last row a and the next row below is row b or if we are at the end of the diagram we can think of the "next row below" as at $n + 1$. Say $m - 1 = a$ for the first case or $m - 1 > a$ is the last column with at least one + in $\hat{D}$ for the second case, and we are considering columns c of all 0's for all $m - 1 < c < b$, or $m - 1 < c < n + 1$. Then we can imagine a column associated to b or $n + 1$, where the last row in the column is a and it extends to the top of the diagram. We place + 's in this "column" in the rows in $\hat{D}$ where there are + 's to left of column $b - 1$, that is their last + 's have not yet passed and these rows are not of type (II) with $a_n < m - 1$. Then we can think of this "column" as the column ℓ in Theorem 5.2.1.

If there are no + 's in this "column", i.e. all last + 's are before m, then we must have $a_n < m$ in which case Step 2(iii) for row a_n would place + 's at (a_n, c) for all c. In particular, this always occurs if we are at the end of the diagram.

If there are + 's in this "column", then once again we look at g, the row of the last + i.e. the first row above b where it's last + has not yet passed. If $g = a$, then Step 1 for row a fills (a, c) with + 's for all c. If $g < a$, then Step 2(iii) for row g fills (g, c) with + 's for all c.

Once again, we can argue in the same way as Theorem 5.2.1 that these are the only + 's in columns c. ∎

From Theorem 5.2.1 and Corollary 5.2.2, we immediately get the following statement.

Corollary 5.2.3 — *There is at least one + in every column of D. More specifically,*

(i) *there is exactly one + in columns corresponding to those in $\hat{D}$ with no + 's, and*
(ii) *there are $s_\ell + t_\ell + 1$ + 's in columns ℓ corresponding to those in $\hat{D}$ with at least one +, with s_ℓ and t_ℓ as defined in Theorem 5.2.1.*

As an example of the different cases for the $\mathbb{L}$-shapes and strings of + 's, we encourage the reader to look at and verify the above statements for the example given in Figure 5.8.

Now that we have our building blocks, what's left is to piece them together.

(**5.2.2**) **Glueing these shapes together.**

The other surprising aspect is the fact that there is even a structure to D and how these shapes glued together. Furthermore, in different parts of D which we will call sections, these shapes consistently glued together in the same way.

Definition 5.2.4 — *Given a Young diagram D of shape $\lambda = (\lambda_1, \cdots, \lambda_k)$, we partition D into k rectangles called **sections** where each section has dimension $j \times (\lambda_j - \lambda_{j+1})$ for $1 \le j \le k$, and is bounded by two rows (one possibly empty). We let $\lambda_{k+1} = 0$.*
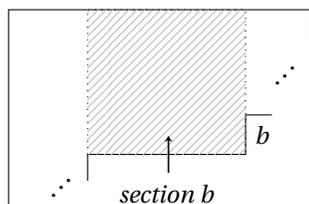
We name each section by its last row.
Here we include empty sections into the count of k (when $\lambda_j = \lambda_{j+1}$ for some j).

Theorem 5.2.5 — *D is made up of glueing together these $\mathbb{L}$-shapes and strings of $+$'s where in each (non-empty) section b, we have a chain of shapes in the following form:*
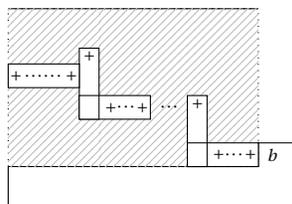
More precisely, we have the following properties for each section:

- *There is at least one shape, either an $\mathbb{L}$ or a string of $+$'s, in each section.*
- *There is either none or exactly one string of $+$'s.*
- *There can be any number of $\mathbb{L}$-shapes, including 0.*
- *$\mathbb{L}$-shapes are glued on the left to other $\mathbb{L}$-shapes at the last $+$ in the vertical part of a previous $\mathbb{L}$, except the first $\mathbb{L}$ in the chain which is glued to row b.*
- *The string of $+$'s is either glued on the left to the last $\mathbb{L}$-shape in the chain at the last $+$ in the vertical part, or glued to row b, in which case it fills the whole width of a section as such:*
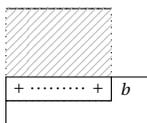
The rest of the boxes in D (the shaded regions) are filled with 0's.

PROOF. Putting together Theorem 5.2.1 and Corollary 5.2.2, D is made up of the following two types of blocks, one consisting of $\mathbb{L}$-shapes where ℓ corresponds to a column in with at least one $+$ in $\hat{D}$, and the other consists of strings of $+$'s which corresponds to consecutive columns of all 0's in $\hat{D}$.

In the blocks, the shaded regions represent boxes filled with all 0's, which extends to fill the rest of the rows in columns m to ℓ in the shape λ. The bolded lines represent the border of λ.

By construction we cannot have a row b occurring in-between columns m to ℓ for either shape, as that would violate Step 1 for row b regardless of type since there are no $+$'s in columns m to $\ell - 1$ in $\hat{D}$ and thus the last $+$ for row b must be to the left of column $\ell - 1$. For example, we cannot have:
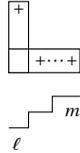
In fact, we actually know more about these blocks since they cannot span across multiple rows. That is, each section of D is made up of a chain of $\mathbb{L}$-shapes and a string of $+$'s as such
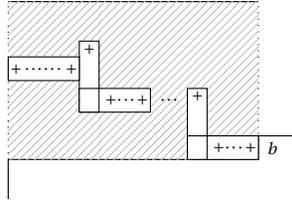
since we know the following properties:

– $\mathbb{L}$-shapes start from a column with at least one $+$ in $\hat{D}$ and extend to the right to the previous column of $+$'s or the start of a row. In particular, if there is a column with at least one $+$ in section b of $\hat{D}$, then there will be a first (right-most) $\mathbb{L}$ in section b of D which has $m - 1 = b_B = b$:
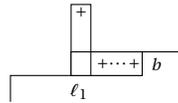
Here ℓ_1 is the first column of $+$'s in $\hat{D}$ in section b.
– Strings of $+$'s (not in the $\mathbb{L}$-shapes) extend to the left until the start of the next row below, and to the right until a column from $\hat{D}$ with at least one $+$. In the special case that section b in $\hat{D}$ consists of columns of all 0's, then the string of $+$'s extends to fill the whole bottom row b of the section (by Step 1 of the algorithm):
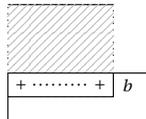
What's left is to show is where the shapes glue together. Consider a particular section b. First, we already saw the special case when a section contains only columns of all 0's, we can assume there is at least one $\mathbb{L}$ in section b. We also saw that the first $\mathbb{L}$ in the section will always glue to the beginning of the row. As we saw in Corollary 5.2.2 that string's of $+$'s can be thought of as special cases of $\mathbb{L}$'s, all we need to show is how $\mathbb{L}$'s glue on the left to other $\mathbb{L}$'s.

Consider an $\mathbb{L}$-shape with vertical part in column ℓ (or with no vertical part) and has right-most and bottom-most box at (b_B, m). And we want to glue to this the previous $\mathbb{L}$ with vertical part in column $m-1$, in rows i to j. We want to see where b_B is in relation this previous $\mathbb{L}$:
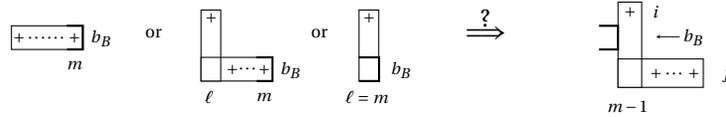
If (b_B, m) is filled with a $+$, since D is a ⅃-diagram and $(i, m-1)$ is always filled with a $+$, we either have $b_B < i$, or $i \le b_B \le j$ and $(b_B, m-1)$ is also filled with a $+$. We also know that if g is the last $+$ in column ℓ in $\hat{D}$, then $b_B \ge g$. Since b_B is either g or a_n if $g < a_n < \ell$, all rows c below b_B (in section b) must have their last $+$ to the right of column m and in particular no such c is a_n. Furthermore, no such row c is of type (II) since this would mean a_n is below row c with $c < a_n < m-1$. But then this would contradict what b_B is, as this inequality suggests $b_B = a_n$ since $g \le b_B < b$ and $m-1 < \ell$. Thus all rows b below b_B must be of type (I), with last $+$ in a column $< m$. By Step 3 of the algorithm for row c, regardless if the last $+$ is in column $m-1$ or not, this means $(b, m-1)$ is filled with a 0. Since this holds for all rows c below b_B in section b, we must have that b_B is the row of the last $+$ in column $m-1$ i.e. in the previous $\mathbb{L}$:
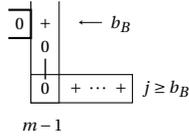
If (b_B, m) is filled with a 0, we are in the special case where $m = \ell$. We also have $b_B \ne a_n$, since in the proof of Theorem 5.2.1 we saw that if a_n was in the vertical part of a $\mathbb{L}$ then (a_n, ℓ) would be filled with a $+$. In particular, this means row b_B is of type (I) with its last $+$ in $\hat{D}$ exactly at (b_B, m). But now we have the same analysis for the rows c below b_B as the $+$ case, since by the ⅃-condition, these rows must have their last $+$'s to the right of column m. Once again, this gives that b_B is the row of the last $+$ in column $m-1$.
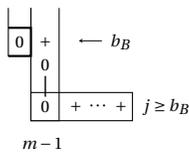
Thus we get that $\mathbb{L}$-shapes, and therefore also strings of $+$'s, glue to the left of $\mathbb{L}$-shapes at the last (bottom-most) $+$ in the vertical part of the $\mathbb{L}$.

Putting everything together, we get that every section b of D looks like one of the following, giving the properties stated in the statement, making up the whole diagram D!
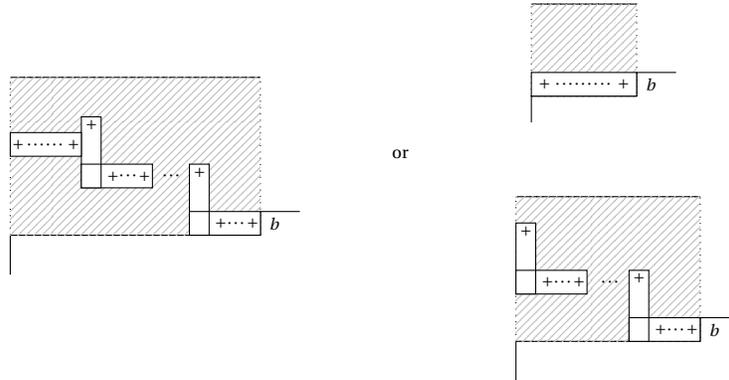

FIGURE 5.9: The possible chains of shapes in each section of D. Generally (on the left), we have a chain of $\mathbb{L}$-shapes with a string of $+$'s on the very left. In the case that section b in $\hat{D}$ consists of columns of all 0's, we only have a string of $+$'s filling the whole bottom row of the section (top-right figure). In the case that there is a column of $+$'s in $\hat{D}$ right before the next row below, there is only a chain of $\mathbb{L}$-shapes (bottom-right figure). The shaded regions are filled with 0's.

$\blacksquare$

(**5.2.3**) **Iterating the algorithm.**

If D is co-loopless, that is π has no fixed points, then the algorithm can be applied again, this time starting with D, to obtain a $\lrcorner$-diagram $\bar{D}$. Assuming that our algorithm is correct (which will be proven in Section 5.3), the associated decorated permutation to $\bar{D}$ would be:

$$\bar{\pi} = \begin{pmatrix} 1 & \cdots & n-1 & n \\ a_2 & \cdots & a_n & a_1 \end{pmatrix}$$

that is from the original $\hat{\pi}$, the permutation is shifted to the left twice.

First, we know that the shape of $\bar{D}$ corresponds to removing the column $a_1 = \pi(1)$ from D and adding in a row labelled a_1 in the appropriate place. For the filling of $\bar{D}$, from Theorem 5.2.5 we know that we really only need to look at what happens the $\mathbb{L}$-shapes. Applying Theorem 5.2.1 to an $\mathbb{L}$-shape gives:
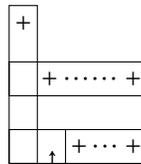
could be $+$ or 0

where now every $+$ in the horizontal string of $+$'s turns into its own $\mathbb{L}$, just with no horizontal part.

5.3. Proof of the algorithm

Finally, we show that the algorithm in Section 5.1.3 actually gives us the correct ⌐-diagram. We already have that D is a ⌐-diagram of type $(k+1, n)$ from Theorem 5.1.2, so what's left is to show that D is loopless, has the right dimension, and has π as its associated decorated permutation.

(**5.3.1**) **Dimension and looplessness of D.**

Other than verifying that the algorithm is correct, using the visual perspective of the algorithm, we can clearly see how the dimension of D arises in relation to the dimension of $\hat{D}$. Namely, it comes from having at least one $+$ in every column of D, with the number of additional $+$'s being exactly the number of non-last $+$'s in $\hat{D}$.

Theorem 5.3.1 — *Under this filling, D is loopless and has dimension* $\dim(S_D) = \dim(S_{\hat{D}}) - 2k + (n-1)$.

PROOF. For this proof, we only need Corollary 5.2.3 and refer to (i) and (ii) from that statement.

First, it follows automatically from having $\geq 1\ +$ in every column of D that D is loopless. For the dimension S_D, (i) and the $+1$ in (ii) gives one $+$ in every column of D for a total of $n - (k+1)\ +$'s.

Summing up what's left over in (ii) gives:

$$
\begin{aligned}
\sum_\ell s_\ell + t_\ell &= \sum_\ell s_\ell + \sum_\ell t_\ell \\
&= \left(\text{\# of non-last } +\text{'s in } \hat{D} - \text{\# of non-last } +\text{'s in col. } a_n \text{ of } \hat{D}\right) \\
&\quad + \left(\text{\# of non-last } +\text{'s in col. } a_n \text{ of } \hat{D}\right) \\
&= \text{\# of non-last } +\text{'s in } \hat{D} \\
&= \dim(S_{\hat{D}}) - k
\end{aligned}
$$

where the sums run over columns $\ell \neq a_n$ in $\hat{D}$ with at least one $+$. The equalities come from:

- Summing over s_ℓ gives the number of non-last $+$'s in each column in $\hat{D}$ excluding column a_n.
- Summing over t_ℓ gives the number of rows of type (II) in $\hat{D}$ with last $+$'s not in column a_n, which in other words is the number of non-last $+$'s in column a_n.
- As $\hat{D}$ is co-loopless and there are k rows all with at least one $+$, there are k last $+$'s out of a total of $\dim(S_{\hat{D}})$, which gives us the last equality.

Putting the two together gives:

$$
\begin{aligned}
\text{\# of } +'s \text{ in } D &= \sum_{\text{cols in (i)}} 1 + \sum_{\text{cols in (ii)}} s_\ell + t_\ell + 1 \\
&= n - (k+1) + \dim(S_{\hat{D}}) - k \\
&= \dim(S_{\hat{D}}) - 2k + (n-1)
\end{aligned}
$$

which verifies that $\dim(S_D) = \dim(S_{\hat{D}}) - 2k + (n-1)$ as needed. ∎

(**5.3.2**) D **corresponds to the correct decorated permutation.**

Recall that $\hat{D}$ is a co-loopless Ⅎ-diagram of type (k, n) associated to the co-loopless decorated permutation $\hat{\pi}$ with k anti-excedances, where

$$\hat{\pi} = \begin{pmatrix} 1 & 2 & \cdots & n \\ a_n & a_1 & \cdots & a_{n-1} \end{pmatrix}.$$

Applying the algorithm in Section 5.1 constructs a loopless Ⅎ-diagram D of type $(k + 1, n)$, with a particular dimensional relationship to $\hat{D}$. We finally prove that D, as constructed, is the Ⅎ-diagram associated to the decorated permutation π, as given by the T-duality map from $\hat{\pi} \mapsto \pi$ (see Section 4.3.3).

Theorem 5.3.2 — *Under this filling, the associated decorated permutation to D is*

$$\pi = \begin{pmatrix} 1 & 2 & \cdots & n \\ a_1 & a_2 & \cdots & a_n \end{pmatrix},$$

where π is loopless and has $k + 1$ anti-excedances.

PROOF. Since D is loopless and of type $(k + 1, n)$, its associated decorated permutation π is also loopless and will have $k + 1$ anti-excedances. Now, what we want to prove is that

$$\begin{cases} \pi(i) = \hat{\pi}(i + 1) = \begin{cases} a_i \neq i + 1 & \text{for non-fixed points } i + 1 \text{ of } \hat{\pi} \\ i + 1 & \text{for fixed points } i + 1 \text{ of } \hat{\pi} \end{cases} & \text{for } i \neq n \\ \pi(n) = \hat{\pi}(1) = a_n \end{cases}$$

Recall that to get from D to its decorated permutation, we go through its pipe dream (see Section 4.2.2). Starting from the label i on the SE border, we follow its pipe until it reaches a label j on the NW border of D, indicating that $\pi(i) = j$. To read off the pipes from D, we start with pipes going leftwards if i is a row and upwards if i is a column. Then boxes with 0's correspond to the pipes continuing straight in the direction it was going, while boxes with $+$'s correspond to the pipes turning. The pipes turn leftwards if it was originally going up, or upwards if originally going to the left. See Figure 5.10 for an example of this process.
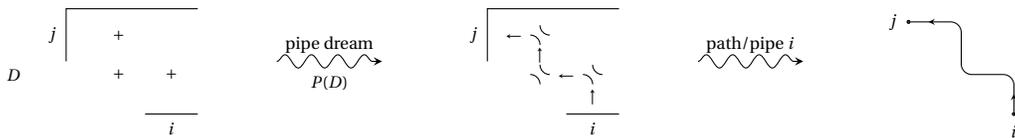

FIGURE 5.10: An example of reading off the decorated permutation π from a Ⅎ-diagram D. On the left, we have the portion of D that affects label i, where unmarked boxes are filled with 0's. Translating to its pipe dream $P(D)$, we get the middle figure, where unmarked boxes are filled with crosses $+$ and the arrows indicate the direction of reading the pipe. On the right, we draw the pipe, also referred to as path, corresponding to i. That is, we have $\pi(i) = j$. Here we have depicted when $j < i$.

We will also refer to the pipe starting at i as the corresponding path for $\pi(i)$, or just i, where the turns are indicating where the $+$'s are. In the following figures, the shaded areas refer to boxes filled with 0's and are used to highlight explicit strings of 0's.

First, for the easiest case of $\pi(n)$, we look at the possibilities of what n corresponds to in D:

– If n is a row in D (with no boxes), and in particular $\pi(n) = n$:

we must have that $a_n = n$ since $\hat{D}$ being co-loopless (no empty rows or rows of all 0's) means that n was a column in $\hat{D}$ with at least one box. Since the only change in rows and columns is through a_n, we get $\pi(n) = n = a_n$ as needed.

– If n is a column in D, where previously in $\hat{D}$ n was a column of all 0's i.e. $\hat{\pi}(n) = n$, then Corollary 5.2.2 and its proof tells us there is a horizontal string of $+$'s at the end of the diagram and in particular they will be in row a_n. Thus we get n is a column in D with exactly one $+$ at (a_n, n), which gives $\pi(n) = a_n$ as needed.
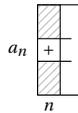
– Finally, if n is a column in D (and thus $n \neq a_n$) and previously n was a column in $\hat{D}$ with at least one $+$ (i.e. $\hat{\pi}(n) \neq n$), then by Theorem 5.2.1 there is an $\mathbb{L}$-shape whose vertical part is in column n. Since we are at the last column of the diagram, we have that the $+$'s in the vertical part of the $\mathbb{L}$ exactly comes from last $+$'s in rows of type (II) in $\hat{D}$, with an extra $+$ in row a_n (since otherwise there needs to be some row whose last $+$ is in some column $> n$). However for a row b to be of type (II), i.e. with a $+$ at (b, a_n) in $\hat{D}$, we must have $a_n > b$. Thus the extra $+$ at (a_n, n) must be the bottom-most $+$ in column n in D. In particular, this means $\pi(n) = a_n$ as needed.
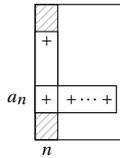
Next, for the case with $i+1$ is a fixed point of $\hat{\pi}$ for $i \neq n$, that is $i+1 \neq a_n$ is a column of all 0's in $\hat{D}$, we look at what i corresponds to in D:

– If i is a row in D, regardless of row type for i, Step 1 places a $+$ at $(i, i+1)$ and this is the only $+$ in column $i+1$ in D. This gives $\pi(i) = i+1$ as needed.
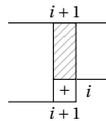
– If i is a column in D where previously in $\hat{D}$ i was a column of all 0's, then Corollary 5.2.2 tells us there is a horizontal string of $+$'s in columns $i, i+1$ in the same row, and these are the only $+$'s, which gives $\pi(i) = i+1$ as needed.
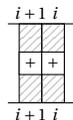
- Finally, if i is a column in D (and thus $i \neq a_n$) where previously in $\hat{D}$ i was a column with at least one $+$, then by Theorem 5.2.1 there is an $\mathbb{L}$-shape with vertical part in column i. Since $i+1$ is a column of all 0's in $\hat{D}$, we know that the one $+$ in column $i+1$ of D corresponds to the rightmost $+$ of either a horizontal string of $+$'s or an $\mathbb{L}$-shape. In either case, by Theorem 5.2.5, this shape must glue to the left of the $\mathbb{L}$ in column i at the last $+$ in column i, say at (b_L, i). Thus we get $+$'s at (b_L, i) and $(b_L, i+1)$ with all 0's below in column i and above in column $i+1$, which gives $\pi(i) = i+1$ as needed.
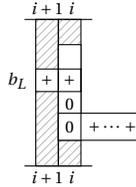
Lastly, we have the hardest case of when $i+1$ is not a fixed point of $\hat{\pi}$ for $i \neq n$, that is in $\hat{D}$ either $i+1$ is a column with at least one $+$ or $i+1$ is a row. Say $\hat{\pi}(i+1) = j$ for some $j \neq i+1$ (we also have $j \neq a_n$). We want to show that $\pi(i) = j$. This is trickier as now there can be a complicated path in $\hat{D}$ to get to j and it is not immediately clear how this translates in D for i. In the subsequent proofs, Theorem 5.2.1, Corollary 5.2.2, and Theorem 5.2.5 will be used without explicit citation.

Figure 5.10, when viewed as for $\lrcorner$-diagram $\hat{D}$ with decorated permutation $\hat{\pi}$ and for $i+1$ instead of i, gives an example of a path when $i+1$ is not a fixed point. In general, every path for a non-fixed point $i+1$ must start with one $+$ in row/column $i+1$ and then is built from alternating between two $+$'s in the same row and two in the same column (where they share a $+$), or vice versa, until row/column j where the path ends with one $+$ in row/column j.

Consider the following general path in $\hat{D}$ which goes through $m+1$ columns of $+$'s,
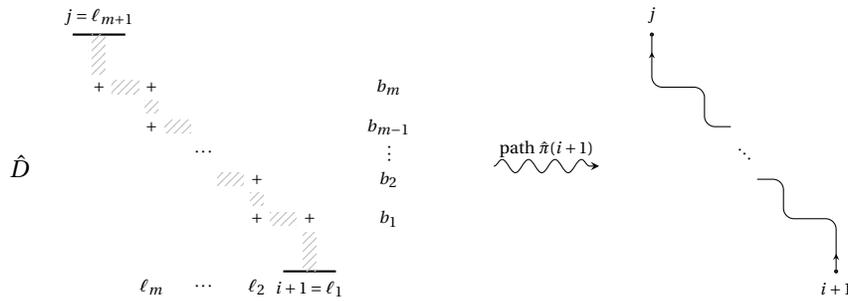
where ℓ_j denotes the columns of $+$'s and b_j denotes the rows of $+$'s. Here, note that $i+1 > b_1 > \cdots > b_m$ and $\ell_1 < \cdots < \ell_{m+1}$. If instead $i+1$ is a row, there is an additional $+$ in the first column of $+$'s at $(i+1, \ell_1)$ where $i+1 > b_1$. If instead j is a row, there is an additional $+$ in the last column of $+$'s at (j, ℓ_{m+1}) where $j < b_m$.

First, notice that by the $\lrcorner$-condition for $\hat{D}$, there are two areas for $+$'s where $\hat{D}$ must be filled with all 0's, depicted below by the shaded region around one $+$.

One area extends below to the next row of +'s or the border of the diagram, and to the left until the end of the diagram. The other area extends on the right to the next column of +'s or the border of the diagram, and above until the top of the diagram. In particular, the shaded areas in-between two +'s in a path must be filled with all 0's, where for the column of +'s these 0's extend to the left until the end of the diagram, and for the row of +'s they extend to the top of the diagram.

Now to tackle the problem at hand, we split the paths $\hat{\pi}(i+1)$ in $\hat{D}$ based on its relation to a_n and consider them separately. As an example of the different types of paths, see Figure 5.11.

(i) $i+1 < j < a_n$: the path in $\hat{D}$ is completely to the right of a_n and j is a column.

(ii) $j < i+1 \le a_n$ or $i+1 \le a_n < j$: the path passes through column a_n and thus either the last + in the path is before a_n (and j is a row), or the path contains +'s in column a_n.

(iii) $j < a_n < i+1$ or $a_n < i+1 < j$: the path passes through where row a_n in D will be.

(iv) $a_n < j < i+1$: the path is completely below where row a_n in D will be and j is a row.
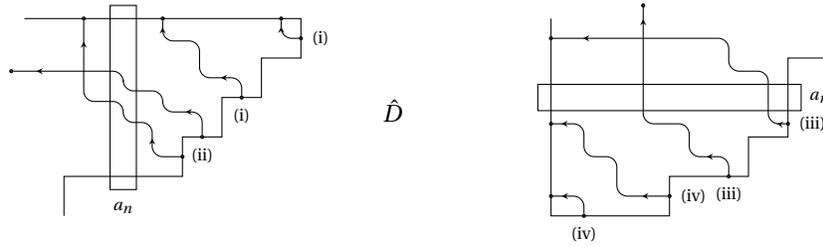
FIGURE 5.11: Examples of the four types of paths in $\hat{D}$ considered in the proof of π. Each path is labelled by its type. On the left, the rectangle indicates where the column a_n is in $\hat{D}$. On the right, the rectangle indicates where the row a_n will be in D.

Note all the paths in (ii) with +'s to the left of a_n must contain +'s in column a_n because of the ⌐-condition for $\hat{D}$, since we know there is a + at $(1, a_n)$ if a box exists there. In the special case of $a_n = 1$, paths in (i) and (ii) do not exist.

Starting with (i), we are in the case where:
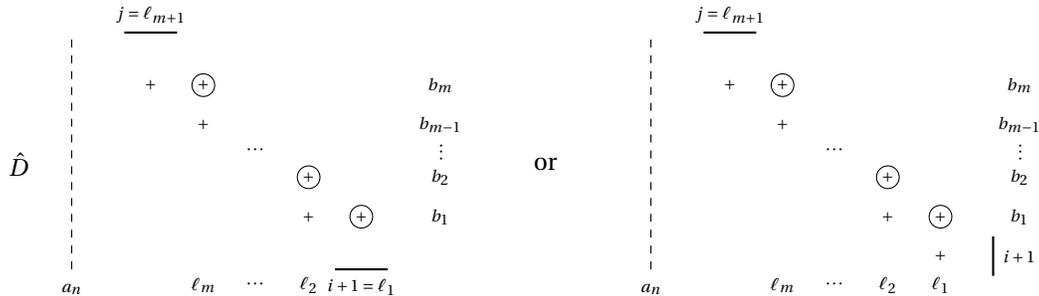
Notice that the top + in each column in the path, except possibly j, is not the last + in their respective rows in $\hat{D}$. In the figure above, these are the circled +'s. In particular this means that in the $\mathbb{L}$'s corresponding to these columns in D, there is also a + in these positions, $(b_1, \ell_1), \dots (b_m, \ell_m)$; in the figures below these are the circled +'s.

We also notice that since $a_n > j$, all $\mathbb{L}$-shapes corresponding to columns $\ell_1,\ldots,\ell_m, j$ each have their $b_T \neq f, a_n$ where f is the row of the first $+$ in $\hat{D}$ in that column. Now let's look at two consecutive columns ℓ_s, ℓ_{s+1}, for $s \neq m$.
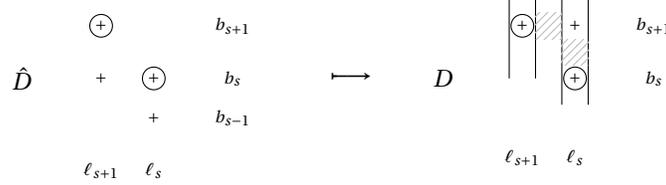
We know that any $+$'s in column ℓ_s in-between rows b_s, b_{s+1} must be last $+$'s in their row because of the $\lrcorner$-condition for $\hat{D}$. Thus, these rows are of type (I) and any last $+$ at (b, ℓ_s) for $b_s > b > b_{s+1}$ corresponds to a 0 at (b, ℓ_s) in D. If there are $+$'s in column ℓ_s above b_{s+1} in $\hat{D}$, then by the $\lrcorner$-condition there must be a $+$ at (b_{s+1}, ℓ_s) in $\hat{D}$. In particular, this is not the last $+$ in row b_{s+1} and thus there is a $+$ at (b_{s+1}, ℓ_s) in D. If (b_s, ℓ_s) is the first $+$ in column ℓ_s, since there are no $+$'s in rows b for $b_s > b > b_{s+1}$ after column ℓ_s by the $\lrcorner$-condition, then row b_{s+1} is the first row above b_s that has a $+$ to the left of column ℓ_s. That is, we have that $b_T = b_{s+1}$ for the $\mathbb{L}$ in column ℓ_s and thus there is a $+$ at (b_{s+1}, ℓ_s). In either case, the $\mathbb{L}$ in column ℓ_s in D always has a $+$ at (b_{s+1}, ℓ_s) and at (b_s, ℓ_s), with 0's in-between. Since in D the $\mathbb{L}$ in column ℓ_{s+1} glues to the left of the $\mathbb{L}$ in column ℓ_s at its bottom-most $+$, there must be 0's in row b_{s+1} in-between columns ℓ_s and ℓ_{s+1}. Therefore, we get the above figure on the right.

For column $j = \ell_{m+1}$, notice that in $\hat{D}$, there must be a row $b < b_m$ with a $+$ to the left of column j since we know at the least there's a $+$ at $(1, a_n)$. Since we know that (b_m, j) is the first $+$ in column j and that $b_T \neq j, a_n$ for the $\mathbb{L}$ in column j in D, we must have that $b_T = b$ and thus there is a $+$ at (b, j) with 0's all above in D. For column ℓ_m, using the same argument as for ℓ_s with row b as b_{s+1}, in the $\mathbb{L}$ in column ℓ_m, there must be a $+$ at (b, ℓ_m) and at (b_m, ℓ_m) with 0's in-between.

Since we know that $\mathbb{L}$'s glue to the left of previous $\mathbb{L}$'s at the first $+$ from the bottom, we get the following in D, regardless if $i + 1$ is a column or a row:
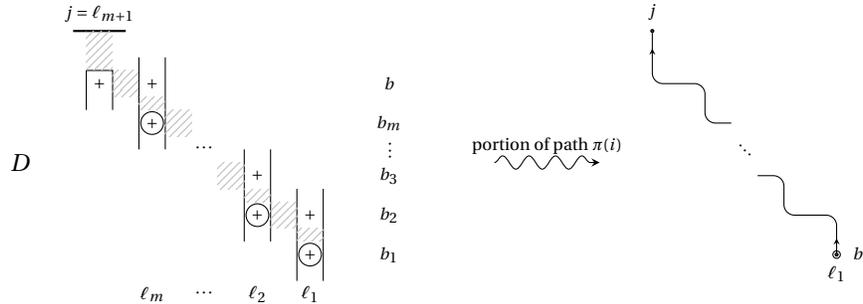

Finally to look at what happens with i in D.

-   If $i + 1 = \ell_1$ and i are both columns, then regardless of how many $+$'s there are in column i in $\hat{D}$, we must have in D a $+$ at (b_1, i) and 0's below in column $i - 1$ since we know how $\mathbb{L}$'s glue.
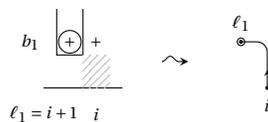
– If $i + 1 = \ell_1$ and i is a row, then we must have $b_1 = i$ (since $\hat{D}$ is co-loopless).
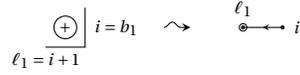
– If $i + 1$ is a row and i is a row, then similarly we must have $b_1 = i$ (since $\hat{D}$ is co-loopless and by the ⌙-condition as there are $+$'s at $(i + 1, \ell_1), (b_1, \ell_1)$).
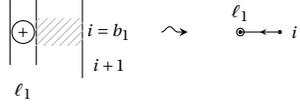
– If $i + 1$ is a row and i is a column, then notice $b_1 < i$ and is a row in $\hat{D}$ with its last $+$ to the left of column i. If i is a column of all 0's in $\hat{D}$, then Step 2(iii) for row b_1 would place the only $+$ in column i in D at (b_1, i). Otherwise, we can use the same argument for i as for ℓ_s, taking ℓ_1 as ℓ_{s+1} (and only looking at rows above i), to show that there must be a $+$ at (b_1, i) and 0's all below in D.
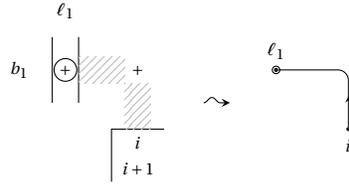

In all the cases, we can connect these path starts at the circled point to the rest of the path, and thus we get that the corresponding path to i in D leads to j, giving us $\pi(i) = j$ as needed for paths in (i).

For paths in (iv), the proof is essentially the same as for (i) except now we need to consider when j is a row. The other difference is we no longer have the fact that $b_T \neq f, a_n$ for columns ℓ_s, which was only used when (b_s, ℓ_s) was the first $+$ in its column in $\hat{D}$. However for paths in (iv), we have that if (b_s, ℓ_s) is the first $+$ in the column, then b_{s+1} is the first row above b_s and below a_n with a $+$ to the left of ℓ_s (since $a_n < j$) and so we still have $b_T = b_{s+1}$ for the ⌙ in column ℓ_s.

When j is a row, we have an additional $+$ at (j, ℓ_{m+1}) in column ℓ_{m+1} above b_m in $\hat{D}$. Now we have $b = j$ for column ℓ_m, and looking at column ℓ_{m+1}, we notice that the $+$ at (j, ℓ_{m+1}) is the last $+$ in row j. We also know that row j is of type (I) since $j > a_n$, and thus in the ⌙ in column ℓ_{m+1} in D, there must be a 0 at (j, ℓ_{m+1}).

Thus we have, where the dotted line refers to where row a_n will be in D:
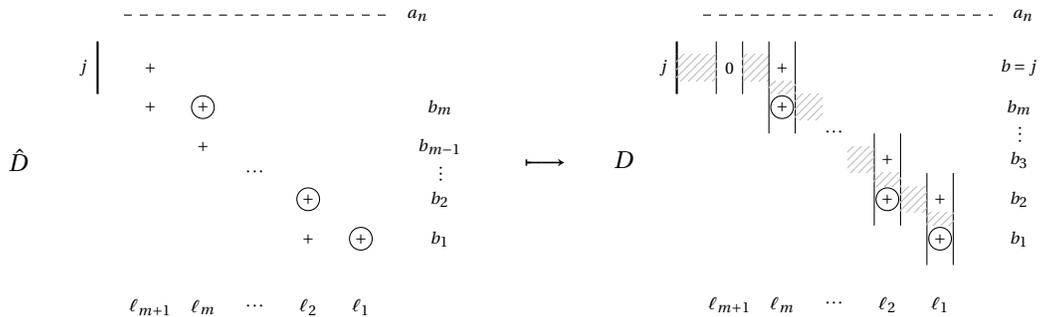
The rest of the analysis is exactly the same for $i+1$ and i, and thus we also get that $\pi(i) = j$ as needed for paths in (iv).

For paths in (ii), first notice we already have the case when the last $+$ in the path is before a_n and j is a row. In this case, all the arguments from paths in (i) stay the same, and we can use the same argument for column ℓ_{m+1} as paths in (iv), since if (j, ℓ_{m+1}) is the last $+$ in row j with $\ell_{m+1} < a_n$, then row j is also of type (I).

When the path contains $+$'s in column a_n, we have (since $j \neq a_n$) either:
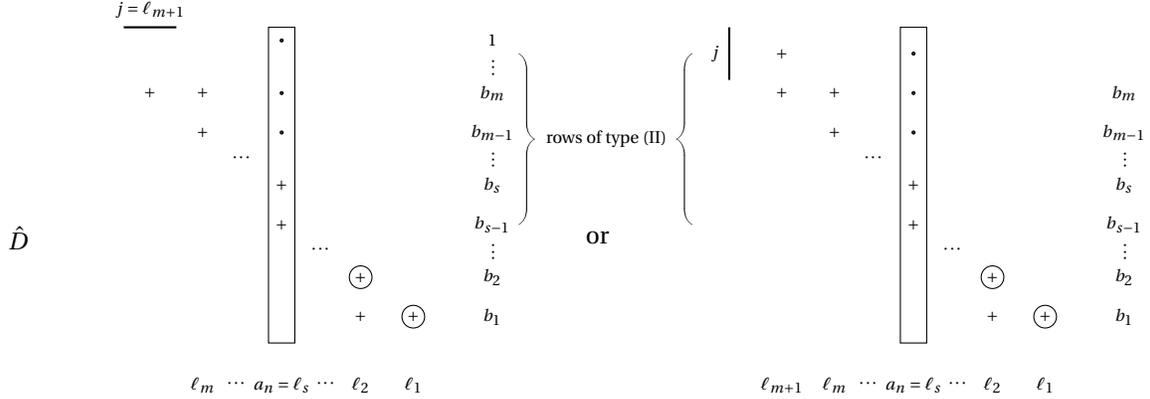

where the dots are indicating $+$'s in column a_n that are not part of the path.

Then notice for columns $\ell_1, \ldots, \ell_{s-1}$, the analysis from paths in (i) stays exactly the same. For columns $\ell_{s+1}, \ldots, \ell_{m+1}$, we note that all rows $b_s, \ldots, b_m$, and j if it is a row, must be of type (II) since by $\lrcorner$-condition, they must have $+$'s in column a_n as there is a $+$ at $(1, a_n)$. In particular, this means that the $\mathbb{L}$'s corresponding to columns $\ell_{s+1}, \ldots, \ell_{m+1}$, which are all $> a_n$, has $+$'s exactly in the same positions as in $\hat{D}$. However now we are done as to get D, column a_n is removed and we know how the $\mathbb{L}$'s are glued, so we have:
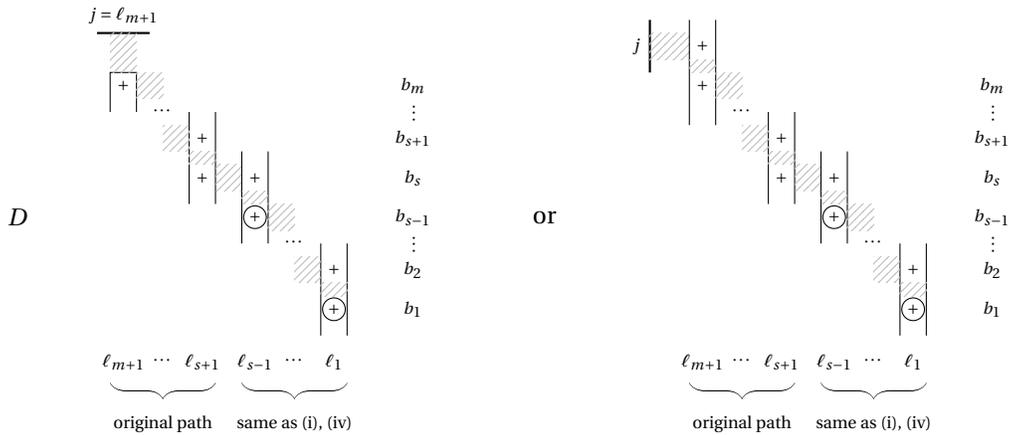
The analysis on $i+1$ and i stays exactly the same as before (since it only depends on $\hat{D}$), even if $i+1 = \ell_1 = a_n$. In this case, the only difference is that the circled $+$ in the analysis for i in D is now the $+$ at (b_1, ℓ_2), and we consider $i+1$ to be a row. Thus we get that $\pi(i) = j$ as needed for paths in (ii).

Lastly for paths in (iii), where the dotted line refers to where row a_n will be in D, we have:

$j = \ell_{m+1}$

$\hat{D}$

rows of type (II)

$$
\begin{array}{l}
b_m \\
b_{m-1} \\
\vdots \\
b_s \\
a_n \\
b_{s-1} \\
\vdots \\
b_2 \\
b_1
\end{array}
$$

$\ell_m \quad \cdots \quad \ell_s \quad \cdots \quad \ell_2 \quad \ell_1$

or

j

$$
\begin{array}{l}
b_m \\
b_{m-1} \\
\vdots \\
b_s \\
a_n \\
b_{s-1} \\
\vdots \\
b_2 \\
b_1
\end{array}
$$

$\ell_{m+1} \quad \ell_m \quad \cdots \quad \ell_s \quad \cdots \quad \ell_2 \quad \ell_1$
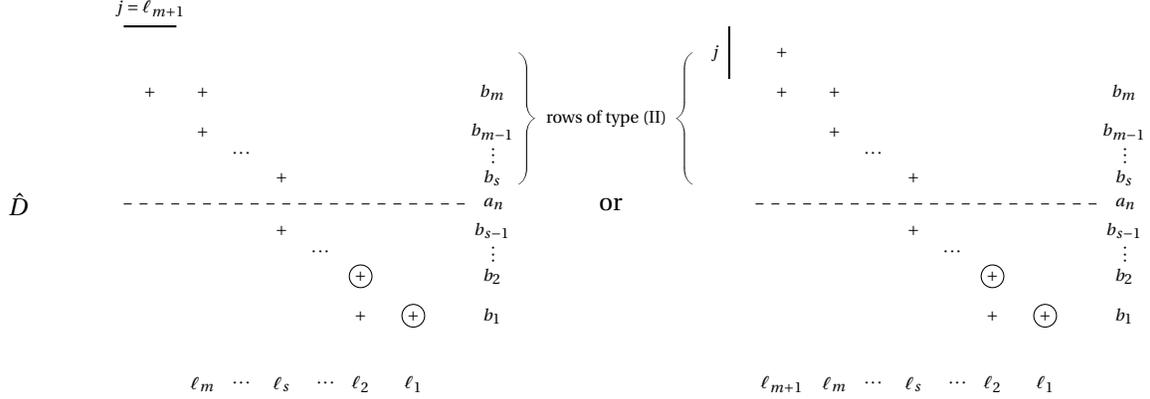
Here we essentially have the same proof as for (ii). Once again for columns $\ell_1, \ldots, \ell_{s-2}$, the analysis from paths in (iv) stays exactly the same. For columns $\ell_{s+1}, \ldots, \ell_{m+1}$, once again the rows $b_s, \ldots, b_m, b_{m+1}$, must be of type (II) since they are all $< a_n$ and have $+$'s to the left of column a_n, and thus the $\mathbb{L}$'s corresponding to these columns have $+$'s in the exact same positions as in $\hat{D}$.

The only difference is we need to look at what happens to column ℓ_s, which row a_n will pass through, and column ℓ_{s-1}. For column ℓ_s, since row a_n will occur in-between rows b_s and b_{s-1}, there will be a $+$ at (a_n, ℓ_s) in the $\mathbb{L}$ corresponding to column ℓ_s. Since row b_s is of type (II) with $\ell_s > a_n$, there is also a $+$ at (b_s, ℓ_s) in the $\mathbb{L}$.

For column ℓ_{s-1}, since the $+$ at (b_{s-1}, ℓ_{s-1}) is not a last $+$ in row b_{s-1}, there is a $+$ in the same position in D. Now, if this $+$ at (b_{s-1}, ℓ_{s-1}) is the first one in column ℓ_{s-1} in $\hat{D}$, then we know that the corresponding $\mathbb{L}$ will have its $b_T = a_n$, in which case there is a $+$ at (a_n, ℓ_{s-1}). Otherwise if there are $+$'s at (b_{s-1}, ℓ_{s-1}) in $\hat{D}$, for $b_{s-1} > b > a_n$ these must be last $+$'s in its row with b of type (I) and thus corresponds to a 0 at (b, ℓ_{s-1}) in D. For $b < a_n$, this means that a_n is in-between the first and last $+$'s in column ℓ_{s-1} and thus there must be a $+$ at (a_n, ℓ_{s-1}). In either case, the $\mathbb{L}$ in column $\ell_s - 1$ always has $+$'s at (a_n, ℓ_{s-1}) and (b_{s-1}, ℓ_{s-1}) with 0's in-between and so we have:

$j = \ell_{m+1}$

D

original path

same as (i), (iv)

$$
\begin{array}{l}
b_m \\
\vdots \\
b_{s+1} \\
b_s \\
a_n \\
b_{s-1} \\
\vdots \\
b_2 \\
b_1
\end{array}
$$

$\ell_{m+1} \quad \cdots \quad \ell_{s+1} \quad \ell_s \quad \ell_{s-1} \quad \cdots \quad \ell_1$

or

a_n

j

$$
\begin{array}{l}
b_m \\
\vdots \\
b_{s+1} \\
b_s \\
a_n \\
b_{s-1} \\
\vdots \\
b_2 \\
b_1
\end{array}
$$

$\ell_{m+1} \quad \cdots \quad \ell_{s+1} \quad \ell_s \quad \ell_{s-1} \quad \cdots \quad \ell_1$
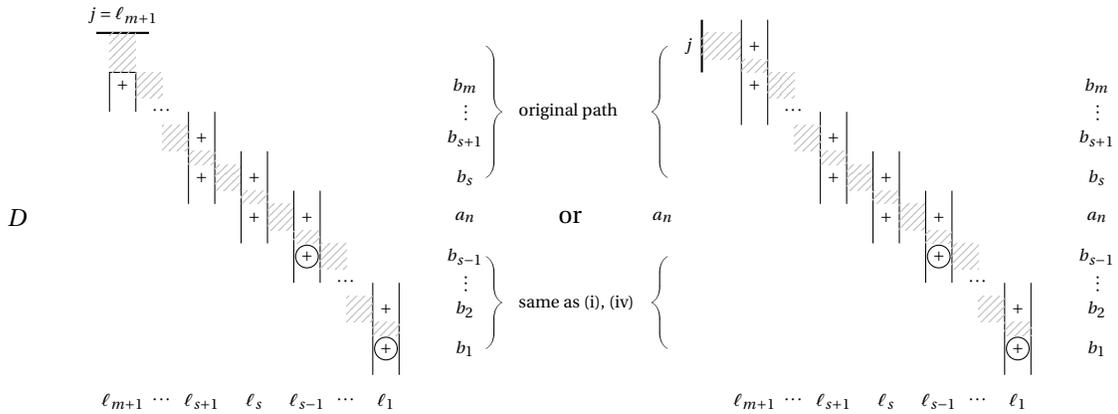
In the special case of $a_n < b_m$ where $j = \ell_{m+1} = \ell_s$, then since we know that the $+$ at (b_m, j) is the first in its column in $\hat{D}$, the $\mathbb{L}$ in column j in D has its $b_T = a_n$ with a $+$ at (a_n, j). Column $\ell_m = \ell_{s-1}$ stays

the same as before with $+$'s at (a_n, ℓ_m) and (b_m, ℓ_m), and 0's in-between, in D:

In the special case of $a_n > b_1$ where $i + 1 = \ell_1$, now the $\mathbb{L}$ in column $\ell_1 = \ell_s$ in D will have a $+$ at (a_n, ℓ_1), since $\ell_1 > a_n$ and thus its $b_B = a_n$. There is also still a $+$ at (b_1, ℓ_1), since $b_1 < a_n$ and thus is of type (II), which aligns with the ℓ_s as in the general case.

The analysis on $i + 1$ and i stays exactly the same as before, with the slight exception of $a_n > b_1$. Since we have $i \geq a_n$ with a $+$ at (a_n, ℓ_1) in D, now a_n plays the role of b_1 and we use the same arguments as before, even if $i = a_n$ in which case we consider i to be a row. In all cases, we get that $\pi(i) = j$ as needed for paths in (iv).

Putting (i)–(iv) together, we are now done since for all paths $\hat{\pi}(i + 1) = j$ we get $\pi(i) = \hat{\pi}(i + 1) = j$! ∎

(**5.3.3**) **T-duality on the level of ⅃-diagrams.**

With the proof that D gives the correct decorated permutation complete, we finally conclude by putting everything together. Theorems 5.1.2, 5.3.1 and 5.3.2, tell us that in fact, the algorithm given in Section 5.1 is telling us what the T-duality map (see Section 4.3.3) looks like on the level of ⅃-diagrams.

Theorem 5.3.3 — *Given a co-loopless ⅃-diagram $\hat{D}$ of type (k, n), the algorithm given in Section 5.1 constructs a loopless ⅃-diagram D of type $(k + 1, n)$ such that*

– *the dimensions of the positroid cells indexed by D and $\hat{D}$ relate via*
$$\dim(S_D) = \dim(S_{\hat{D}}) - 2k + (n - 1),$$
– *and the associated decorated permutations to D and $\hat{D}$ relate via T-duality.*

That is, the algorithm gives the T-duality map from $\hat{\pi} \mapsto \pi$ on the level of ⅃-diagrams.

III

FINAL THOUGHTS

6 | CONCLUSION

6.1. FURTHER DIRECTIONS

In Part I, we started with the tale of the c_2-invariant as an avatar for the Feynman period and saw how combinatorial techniques involving enumerating edge bipartitions through swapping arguments were able to resolve the $p = 2$ case of the long sought after completion conjecture for the c_2-invariant. However, the story is not over as of course what we would really like is a proof of the full conjecture.

Thus, the pivotal next direction to look at is:
Using this idea of swapping edges around a vertex, or vertices, across the edge bipartition, can we generalize the counting arguments seen in Chapter 3 for higher primes p?

For $p > 2$, the combinatorial interpretation of the c_2-invariant becomes much more intricate as now we are enumerating the ways of distributing $p-1$ copies of each edge across $2p-2$ polynomials, where $p-1$ of them come from spanning trees and the other $p-1$ arises from particular spanning 2-forests. Comparing to the $p = 2$ case, recall that we were bipartitioning one copy of each edge into a spanning tree, spanning 2-forest pair. The difficulty in higher p is that now there no longer needs to be a nice pairing up of trees and forests that exactly partition full sets of edges from the graph. Optimistically, Yeats [Yea18a] has been able to first answer this question in the T-case where we only needed to swap around 2-valent vertices. The obstacle, naturally, in the S and R cases is in how to generalize this to include control vertices or perhaps instead how to circumvent the need for control vertices.

Finally, one could also ask about how much farther these enumerative techniques can take us. Can they answer other questions about the c_2-invariant or compute the c_2's for new families of graphs?

In Part II, we looked at the tale of the combinatorial T-duality map as a bridge between triangulations of the hypersimplex and triangulations of the amplituhedron and what this map looked like on ⅃-diagrams. One result, other than the nice structure of the ⅃-diagrams arising from T-duality, was that we could directly see where the dimensional relationship between the positroid cells on either side of the map manifests from.

Continuing with this thread, a natural general question to further investigate is what other properties of positroids or triangulations can be determined through looking at T-duality via ⅃-diagrams? Two specific directions that could be of interest are:

- The T-duality map turns out to be a special case of the $\rho_A(\pi)$ map defined by Benedetti, Chavez and Tamayo [BCT19] when characterizing quotients of uniform positroids. Thus, can we extend our algorithm from Chapter 5 to this more general cyclic shift map?
- In [PSBW21], Parisi, Sherman-Bennett, and Williams, were able to prove Theorem 4.3.7 through looking at the T-duality map via plabic graphs. Thus, are there any interesting connections between the two perspectives on T-duality outside of the canonical bijection between ⅃-diagrams and plabic graphs from [Pos06]? Furthermore, is there any structure to the ⅃-diagrams underlying the triangulations on either side of T-duality?

6.2. Final thoughts

Through the underlying theme of emerging combinatorics in the world of scattering amplitudes, one could see the parallels between the stories of the c_2-invariant and the T-duality map. Both branched off of the need to better understand the scattering amplitude, one through Feynman diagrams in scalar ϕ^4-theory and one through on-shell diagrams in $\mathcal{N} = 4$ SYM. Both then looked at related but simpler, and in some sense nicer, objects with interesting connections to many areas of mathematics, one reducing to the c_2-invariant and the other reducing to positroid cells of the positive Grassmannian. In the end, both then converged to the idea that looking at these objects through a combinatorial perspective can be a powerful tool in answering questions about them, one being the completion conjecture for the c_2-invariant and the other being the relationship between triangulations of the amplituhedron and the hypersimplex. At the heart of it all, they showcased a wonderful tale of beautiful mathematics at the intersection of combinatorics and quantum field theory!

References

Chapter 1. Introduction

[AH+16]   Nima Arkani-Hamed, Jacob Bourjaily, Freddy Cachazo, Alexander Goncharov, Alexander Postnikov, and Jaroslav Trnka. *Grassmannian geometry of scattering amplitudes*. Cambridge University Press, Cambridge, 2016, pp. ix+194. arXiv:1212.5605.

[AHT14]   Nima Arkani-Hamed and Jaroslav Trnka. "The Amplituhedron". *Journal of High Energy Physics* 2014.10 (2014). arXiv:1312.2007.

[Bri+05]   Ruth Britto, Freddy Cachazo, Bo Feng, and Edward Witten. "Direct proof of the tree-level scattering amplitude recursion relation in Yang-Mills theory". *Phys. Rev. Lett.* 94.18 (2005), pp. 181602, 4. arXiv:hep-th/0501052.

[Bro10]   Francis Brown. "On the periods of some Feynman integrals" (2010). arXiv:0910.0114.

[IZ80]   Claude Itzykson and Jean Bernard Zuber. *Quantum field theory*. International Series in Pure and Applied Physics. McGraw-Hill International Book Co., New York, 1980, pp. xxii+705.

[Pos06]   Alexander Postnikov. "Total positivity, Grassmannians, and networks" (2006).

[Yea17]   Karen Yeats. *A combinatorial perspective on quantum field theory*. Vol. 15. SpringerBriefs in Mathematical Physics. Springer, Cham, 2017, pp. ix+120.

[Yea18b]   Karen Yeats. "A special case of completion invariance for the c_2 invariant of a graph". *Canad. J. Math.* 70.6 (2018), pp. 1416–1435. arXiv:1706.08857.

Part I. The c_2-invariant

[Ax64]   James Ax. "Zeroes of polynomials over finite fields". *Amer. J. Math.* 86 (1964), pp. 255–261.

[BB03]   Prakash Belkale and Patrick Brosnan. "Matroids, motives, and a conjecture of Kontsevich". *Duke Math. J.* 116.1 (2003), pp. 147–188. arXiv:math/0012198.

[BEK06]   Spencer Bloch, Hélène Esnault, and Dirk Kreimer. "On motives associated to graph polynomials". *Comm. Math. Phys.* 267.1 (2006), pp. 181–225. arXiv:math/0510011.

[BK95]   D. J. Broadhurst and D. Kreimer. "Knots and numbers in ϕ^4 theory to 7 loops and beyond". *Internat. J. Modern Phys. C* 6.4 (1995), pp. 519–524. arXiv:hep-ph/9504352.

[BK97]   D. J. Broadhurst and D. Kreimer. "Association of multiple zeta values with positive knots via Feynman diagrams up to 9 loops". *Phys. Lett. B* 393.3-4 (1997), pp. 403–412. arXiv:hep-th/9609128.

[Bro09]   Francis Brown. "The massless higher-loop two-point function". *Comm. Math. Phys.* 287.3 (2009), pp. 925–958. arXiv:0804.1660.

[Bro10]   Francis Brown. "On the periods of some Feynman integrals" (2010). arXiv:0910.0114.

[Bro13]   Francis Brown. "Iterated integrals in quantum field theory". *Geometric and topological methods for quantum field theory*. Cambridge Univ. Press, Cambridge, 2013, pp. 188–240.

[BS12]   Francis Brown and Oliver Schnetz. "A K3 in ϕ^4". *Duke Math. J.* 161.10 (2012), pp. 1817–1862. arXiv:1006.4064.

[BS13]   Francis Brown and Oliver Schnetz. "Modular forms in quantum field theory". *Commun. Number Theory Phys.* 7.2 (2013), pp. 293–325. arXiv:1304.5342.

[BS15]   Francis Brown and Oliver Schnetz. "Single-valued multiple polylogarithms and a proof of the zig-zag conjecture". *J. Number Theory* 148 (2015), pp. 478–506. arXiv:1208.1890.

[BSY14]   Francis Brown, Oliver Schnetz, and Karen Yeats. "Properties of c_2 invariants of Feynman graphs". *Adv. Theor. Math. Phys.* 18.2 (2014), pp. 323–362. arXiv:1203.0188.

[BY11]   Francis Brown and Karen Yeats. "Spanning forest polynomials and the transcendental weight of Feynman graphs". *Comm. Math. Phys.* 301.2 (2011), pp. 357–382. arXiv:0910.5429.

[CDY16]   Iain Crump, Matt DeVos, and Karen Yeats. "Period preserving properties of an invariant from the permanent of signed incidence matrices". *Ann. Inst. Henri Poincaré D* 3.4 (2016), pp. 429–454. arXiv:1505.06987.

[Cha82]   Seth Chaiken. "A combinatorial proof of the all minors matrix tree theorem". *SIAM J. Algebraic Discrete Methods* 3.3 (1982), pp. 319–329.

[CY19]   Wesley Chorney and Karen Yeats. "c_2 invariants of recursive families of graphs". *Ann. Inst. Henri Poincaré D* 6.2 (2019), pp. 289–311. arXiv:1701.01208.

[Dor11]   Dzmitry Doryn. "On one example and one counterexample in counting rational points on graph hypersurfaces". *Lett. Math. Phys.* 97.3 (2011), pp. 303–315. arXiv:1006.3533.

[Dor17]   Dmitry Doryn. "The c_2 invariant is invariant". *Adv. Theor. Math. Phys.* 21.8 (2017), pp. 1953–1989. arXiv:1312.7271.

[EY21]   Maria Esipova and Karen Yeats. Work in progress. 2021.

[Hu+18]   Simone Hu, Oliver Schnetz, Jim Shaw, and Karen Yeats. "Further investigations into the graph theory of ϕ^4-periods and the c_2 invariant" (2018). Accepted, to appear in Annales de l'Institut Henri Poincaré D. arXiv:1812.08751.

[Kir47]   G. Kirchhoff. "Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird". *Annalen der Physik* 148.12 (1847), pp. 497–508.

[KZ01]   Maxim Kontsevich and Don Zagier. "Periods". *Mathematics unlimited—2001 and beyond.* Springer, Berlin, 2001, pp. 771–808.

[LMY19]   Mohamed Laradji, Marni Mishna, and Karen Yeats. "Some results on double triangle descendants of K_5" (2019). arXiv:1904.06923.

[Pan19]   Erik Panzer. "Hepp's bound for Feynman graphs and matroids" (2019). arXiv:1908.09820.

[Sage]   The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 8.2).* http://www.sagemath.org.

[Sch10]   Oliver Schnetz. "Quantum periods: a census of ϕ^4-transcendentals". *Commun. Number Theory Phys.* 4.1 (2010), pp. 1–47. arXiv:0801.2856.

[Sch11]   Oliver Schnetz. "Quantum field theory over $\mathbb{F}_q$". *Electron. J. Combin.* 18.1 (2011), Paper 102, 23. arXiv:0909.0905.

[Sch19]   Oliver Schnetz. "Geometries in perturbative quantum field theory" (2019). arXiv:1905.08083.

[SY21]   Oliver Schnetz and Karen Yeats. "c_2 invariants of hourglass chains via quadratic denominator reduction" (2021). arXiv:2102.12383.

[Yea16]   Karen Yeats. "A few c_2 invariants of circulant graphs". *Commun. Number Theory Phys.* 10.1 (2016), pp. 63–86. arXiv:1507.06974.

[Yea17]   Karen Yeats. *A combinatorial perspective on quantum field theory.* Vol. 15. SpringerBriefs in Mathematical Physics. Springer, Cham, 2017, pp. ix+120.

[Yea18a]   Karen Yeats. Private communications. 2018.

[Yea18b]   Karen Yeats. "A special case of completion invariance for the c_2 invariant of a graph". *Canad. J. Math.* 70.6 (2018), pp. 1416–1435. arXiv:1706.08857.

[Yea18c]   Karen Yeats. "A study on prefixes of c_2 invariants" (2018). arXiv:1805.11735.

Part II. Le diagrams

[AH+16]   Nima Arkani-Hamed, Jacob Bourjaily, Freddy Cachazo, Alexander Goncharov, Alexander Postnikov, and Jaroslav Trnka. *Grassmannian geometry of scattering amplitudes*. Cambridge University Press, Cambridge, 2016, pp. ix+194. arXiv:1212.5605.

[AHT14]   Nima Arkani-Hamed and Jaroslav Trnka. "The Amplituhedron". *Journal of High Energy Physics* 2014.10 (2014). arXiv:1312.2007.

[CW07]   Sylvie Corteel and Lauren K. Williams. "Tableaux combinatorics for the asymmetric exclusion process". *Adv. in Appl. Math.* 39.3 (2007), pp. 293–310. arXiv:0602109.

[Ful97]   William Fulton. *Young tableaux*. Vol. 35. London Mathematical Society Student Texts. With applications to representation theory and geometry. Cambridge University Press, Cambridge, 1997, pp. x+260.

[Gel+87]   I. M. Gelfand, R. M. Goresky, R. D. MacPherson, and V. V. Serganova. "Combinatorial geometries, convex polyhedra, and Schubert cells". *Adv. in Math.* 63.3 (1987), pp. 301–316.

[KL72]   S. L. Kleiman and Dan Laksov. "Schubert calculus". *Amer. Math. Monthly* 79 (1972), pp. 1061–1082.

[KW13]   Yuji Kodama and Lauren Williams. "Combinatorics of KP solitons from the real Grassmannian". *Algebras, quivers and representations*. Vol. 8. Abel Symp. Springer, Heidelberg, 2013, pp. 155–193. arXiv:1205.1101.

[KWZ20]   Steven N. Karp, Lauren K. Williams, and Yan X. Zhang. "Decompositions of amplituhedra". *Ann. Inst. Henri Poincaré D* 7.3 (2020). With an appendix by Karp, Williams, Zhang and Hugh Thomas, pp. 303–363. arXiv:1708.09525.

[LPW20]   Tomasz Lukowski, Matteo Parisi, and Lauren K. Williams. "The positive tropical Grassmannian, the hypersimplex, and the m=2 amplituhedron" (2020). arXiv:2002.06164.

[Mnë88]   N. E. Mnëv. "The universality theorems on the classification problem of configuration varieties and convex polytopes varieties". *Topology and geometry—Rohlin Seminar*. Vol. 1346. Lecture Notes in Math. Springer, Berlin, 1988, pp. 527–543.

[Oxl11]   James Oxley. *Matroid theory*. Second. Vol. 21. Oxford Graduate Texts in Mathematics. Oxford University Press, Oxford, 2011, pp. xiv+684.

[Pos06]   Alexander Postnikov. "Total positivity, Grassmannians, and networks" (2006).

[Pos18]   Alexander Postnikov. "Positive Grassmannian and polyhedral subdivisions". *Proceedings of the International Congress of Mathematicians—Rio de Janeiro 2018. Vol. IV. Invited lectures*. World Sci. Publ., Hackensack, NJ, 2018, pp. 3181–3211. arXiv:1806.05307.

[PSBW21]   Matteo Parisi, Melissa Sherman-Bennett, and Lauren Williams. "The m=2 amplituhedron and the hypersimplex: signs, clusters, triangulations, Eulerian numbers" (2021). arXiv:2104.08254.

[PSW09]   Alexander Postnikov, David Speyer, and Lauren Williams. "Matching polytopes, toric geometry, and the totally non-negative Grassmannian". *J. Algebraic Combin.* 30.2 (2009), pp. 173–191. arXiv:0706.2501.

[TW15]   E. Tsukerman and L. Williams. "Bruhat interval polytopes". *Adv. Math.* 285 (2015), pp. 766–810. arXiv:1406.5202.

Part III. Final thoughts

[BCT19]   Carolina Benedetti, Anastasia Chavez, and Daniel Tamayo. "Quotients of uniform positroids" (2019). arXiv:1912.06873.

[Pos06]   Alexander Postnikov. "Total positivity, Grassmannians, and networks" (2006).

[PSBW21]   Matteo Parisi, Melissa Sherman-Bennett, and Lauren Williams. "The m=2 amplituhe-dron and the hypersimplex: signs, clusters, triangulations, Eulerian numbers" (2021). arXiv:2104.08254.

[Yea18a]   Karen Yeats. Private communications. 2018.