

# Bijectivizing the PT–DT Correspondence

by

Cruz Godar

A dissertation accepted and approved in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

in Mathematics

Dissertation Committee:

Benjamin Young, Chair

Patricia Hersh, Core Member

Peter Ralph, Core Member

Christopher Sinclair, Core Member

Cengiz Zopluoglu, Institutional Representative

University of Oregon

Spring 2025

© This work is openly licensed under CC BY-SA 4.0 in 2025 by Cruz Godar.

## ABSTRACT

Cruz Godar

Doctor of Philosophy in Mathematics

Title: Bijectivizing the PT–DT Correspondence

Pandharipande–Thomas theory and Donaldson–Thomas theory (PT and DT) are two branches of enumerative geometry in which particular generating functions arise that count plane-partition-like objects. That these generating functions differ only by a factor of MacMahon’s function was proven recursively by Jenne, Webb, and Young using the double dimer model. We bijectivize two special cases of the correspondence by formulating these generating functions using vertex operators and applying a particular type of local involution known as a *toggle*, first introduced in the form we use by Pak. For the general form of the correspondence, we prove a number of foundational results about the objects in question and make progress toward a bijection.

The toggle involution was used by Pak to prove one of two known bijections between (reverse) plane partitions and hook-length-weighted tableaux. The presentations of these two maps (the Hillman–Grassl and Pak–Sulzgruber algorithms HG and PS) are very different, but work by Garver and Patrias introduced a universal language that can express both HG and PS as pairs of permutations. The question remains whether further algorithms are possible; we answer in the affirmative for those on  $3 \times 3$  tableaux by introducing a novel approach to efficiently filter out invalid algorithms and to prove the validity of those that remain with integer programs.

# Contents

<b>1</b>	<b>Introduction</b>	<b>10</b>
<b>2</b>	<b>Vertex Operators and Single-Dimer Objects</b>	<b>13</b>
2.1	Plane Partitions . . . . .	13
2.2	Toggles . . . . .	18
2.3	One-Leg Objects . . . . .	27
2.4	Two-Leg Objects . . . . .	39
<b>3</b>	<b>Double-Dimer Objects</b>	<b>50</b>
3.1	Three-Leg Objects . . . . .	50
3.2	The Entry Posets of Three-Leg Objects . . . . .	55
3.3	Future Directions . . . . .	62
<b>4</b>	<b>Diagonal RSK and Categorizing Bijections</b>	<b>64</b>
4.1	The RSK and Garver–Patrias Algorithms . . . . .	65
4.2	Properties of GP . . . . .	70
4.3	The Injectivity of GP as an Integer Program . . . . .	75
4.4	Results for $3 \times 3$ Tableaux . . . . .	78
4.5	Conjectures for Larger Tableaux . . . . .	82
<b>A</b>	<b>A Complete List of Bijections <math>\text{GP}_{\mathcal{L}}^{\mathcal{O}}</math> for <math>3 \times 3</math> Tableaux</b>	<b>86</b>

# List of Figures

2.1.1	A 4-hook with pivot $(5, 2)$ in an asymptotic Young diagram $\mathbb{N}^2 \setminus \lambda$ (dark gray), and a 4-hook with pivot $(1, 2)$ in the Young diagram $\lambda$ (light gray). . . . .	15
2.1.2	A plane partition of weight 31, visualized both as a grid of numbers and a stack of 31 blocks. . . . .	15
2.1.3	By placing them as diagonals in a plane partition, we see that $(5, 3, 1, 1) > (3, 2, 1)$ . . . . .	16
2.2.1	Toggling $(3, 2, 1)$ relative to $(5, 3, 1, 1)$ and $(3, 2)$ (middle- and far-left), and toggling $(5, 3, 1, 1)$ relative to $(3, 2, 1)$ and $(4, 2, 1)$ (middle- and far-right). . . . .	19
2.2.2	A cell $x \in \mathbb{N}^2$ and its hook. The edges of the diagram are labeled with the exponents of the $\Gamma$ operators they correspond to. The hook of $x$ intersects the boundary at edges corresponding to operators $\Gamma_+(q^{5/2})$ and $\Gamma_-(q^{3/2})$ , and the hook length of $x$ is $h(x) = 4 = \frac{5}{2} + \frac{3}{2}$ . . . . .	23
2.2.3	A weight-7 plane partition $\pi$ (far left) being mapped bijectively to a weight-7 tableau $\tau(\pi)$ that is weighted by hook length (far right). The center-left and center-right figures are the first two steps in the bijection. . . . .	26
2.3.1	The signs of the $\Gamma$ operators for $\lambda = (4, 2, 1)$ . . . . .	28
2.3.2	Hooks and their boundary edges in $\lambda = (4, 2, 1)$ and the Young diagram asymptotic to it. . . . .	30
2.3.3	A hook inside $\lambda = (4, 4, 3, 1)$ (blue) and a corresponding hook in $\mathbb{N}^2 \setminus \lambda$ intersecting its boundary edges. . . . .	32
2.3.4	A 4-hook in a skew plane partition of shape $(\emptyset, \emptyset, \lambda)$ and the corresponding 4-hook in $\lambda$ (left). The corresponding inner and outer corners in the 4-quotient $\lambda_{4,3}$ (right). . . . .	33

2.3.5	Pivots of corresponding 3-hooks in the Young diagram asymptotic to $\lambda = (3, 3)$ and its quotients. In reading order: $\mathbb{N}^2 \setminus \lambda$ , $\mathbb{N}^2 \setminus \lambda_{3,0}$ , $\mathbb{N}^2 \setminus \lambda_{3,1}$ , $\mathbb{N}^2 \setminus \lambda_{3,2}$ , $\lambda$ , $\mathbb{N}^2$ . . . . .	35
2.3.6	A skew plane partition $\sigma$ of shape $(\emptyset, \emptyset, (2, 1))$ . . . . .	37
2.3.7	Decomposing a one-leg SPP into a hook-length-weighted tableau. At each step, we choose a corner of the asymptotic Young diagram (here, we choose them in lexicographic order) and toggle its diagonal. . . . .	38
2.3.8	Rearranging the entries of the tableau $T(\sigma)$ (left) into a tableau $T(\rho)$ (center) of shape $\lambda$ and a tableau $T(\pi)$ (right) of shape $\mathbb{N}^2$ . . . . .	38
2.3.9	The untoggled RPP $\rho = T^{-1}(T(\rho))$ (left) and plane partition $\pi = T^{-1}(T(\pi))$ . . . . .	38
2.4.1	A two-leg skew plane partition of shape $((2, 2), (3, 1), \emptyset)$ , visualized both as a grid of numbers and a stack of 10 weight-contributing blocks (dark gray) on top of a non-removable ‘‘tray’’ of blocks that contributes weight 1, as measured by the vertex operator expression Equation (2.7). . . . .	40
2.4.2	A one-leg RPP of shape $(\emptyset, \emptyset, (4, 3, 1))$ and weight 19, visualized as 19 boxes removed from an infinite vertical tower. . . . .	41
2.4.3	A two-leg RPP of shape $((3, 1), (2, 2), \emptyset)$ and weight 7, visualized as a tray with 6 boxes removed (the minimal configuration has weight 1 as measured by the vertex operators). . . . .	42
2.4.4	Three steps of the inductive portion of the proof of Proposition 2.4.3. Left: $a = \alpha(3, 2)$ and $b = \alpha(3, 3)$ in a diagram that is $\sigma_3$ with one toggled corner. .	44
2.4.5	An SPP of shape $((2, 2), (3, 1), \emptyset)$ and weight 16. . . . .	47
2.4.6	Iteratively toggling the diagonals of a two-leg SPP. . . . .	47
2.4.7	The limiting diagram after toggling the SPP in Figure 2.4.6. . . . .	48
2.4.8	Commuting the operator $\Gamma_-(q^{1/2})$ past every other $\Gamma_-$ has the effect of toggling every diagonal on the right side of the diagram, with the exception of the topmost and bottommost. . . . .	48
2.4.9	The result of palindromically commuting the $\Gamma$ operators of the same sign. .	48
2.4.10	The two-leg RPP $\rho$ (left; after transposing) and the plane partition $\pi$ (right; after untoggling) corresponding to the SPP $\sigma$ from Figure 2.4.5. . . . .	48

3.1.1	A plane partition with walls shown (left), each rhombus painted with a dimer (center), and the configuration formed from those dimers (right). . . . .	51
3.1.2	Regions I (orange), II (yellow), and III (purple) for $\lambda = (3, 2)$ , $\mu = (2, 1, 1)$ , and $\nu = (4, 2, 1)$ . Taking only regions I and III (left) or II and III (center) produce plane-partition-like objects, while all three together (right) do not. . .	52
3.1.3	The minimal AB configuration with visible dimers for $\lambda = (3, 2)$ , $\mu = (2, 1, 1)$ , and $\nu = (4, 2, 1)$ , as in Figure 3.1.2 (top row). Those dimers alone (middle row). Overlaying those configurations to produce a double-dimer configuration (bottom). Notice that sufficiently far from the origin, the paths follow the shape of the Young diagrams for $\lambda$ , $\mu$ , and $\nu$ , and although they may cross the $120^\circ$ sectors indicated by the red lines when close to the origin, they eventually each remain in only one sector. . . . .	54
3.2.1	Valid values for $A$ (left) and $B$ (center) to create a three-leg RPP $\rho$ of shape $(\lambda, \mu, \nu)$ , where $\lambda = (3, 2)$ , $\mu = (2, 1, 1)$ , and $\nu = (4, 2, 1)$ . The five connected components created from the boxes removed from I, left in II, and removed with multiplicity one from III, colored by region (right). . . . .	56
3.2.2	The poset diagram for the entries $(3, 1)$ (left), $(2, 2)$ (center), and $(1, 4)$ (right) in the three-leg RPP given in Figure 3.2.1, where all other entries are held fixed. Invalid labeling configurations are represented by a missing table entry, and valid configurations are represented by either the label of their connected component, a star if the only boxes in the component are unlabeled, or an empty set if there are no boxes in the component at all. . .	57
3.2.3	Values for $A$ (left) and $B$ (center) to create a three-leg RPP $\rho$ of shape $(\lambda, \mu, \nu)$ , where $\lambda = (2)$ , $\mu = (2, 2)$ , and $\nu = (1, 1)$ . The poset diagram for $(2, 1)$ that fails to have a grading-preserving bijection to a product of intervals (right). . . . .	61

4.1.1	Inserting a 2 into an SSYT. At each step, we draw the insertion arrow after the row under consideration. First, the 2 bumps the 3 in the first row; then that 3 is inserted into the second row, where it bumps the 4. That 4 is inserted into the third row and bumps the 6, which then is inserted into the (empty) fourth row, where it remains. . . . .	66
4.1.2	The SSYTs $P$ and $Q$ (left and right, respectively) that are the output of the RSK algorithm on the biword $b$ as defined in Equation (4.1). . . . .	67
4.1.3	Top: an example of choices for $\mathcal{O}$ and $\mathcal{L}$ that make $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ , and its output on an example tableau $t$ . Bottom: the outputs of applying the RSK algorithm to each word $w_c(t)$ formed from the tableau $t$ , and their shapes, which are placed as diagonals to form the output RPP $\text{GP}_{\mathcal{L}}^{\mathcal{O}}(t)$ . . . . .	68
4.2.1	A Young diagram of shape $\lambda = (7, 6, 4, 2)$ and a rectangle $B$ with $B \subseteq \lambda$ . The bottom-right-most cells in $B$ have their diagonal's content written in them, and the bottom-right-most cells in diagonal $c$ in $\lambda$ have $d = b(c)$ written in them in the notation of Proposition 4.2.6. The diagonal containing the corner box $(i_M, j_M) = (3, 4)$ of $B$ is $d_0 = 1$ . . . . .	73
4.3.1	A pair $(\mathcal{O}, \mathcal{L})$ for a potential $2 \times 3$ algorithm $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ (top). The order and label tableaux produced by restricting $\mathcal{O}$ and $\mathcal{L}$ to the left and right $2 \times 2$ blocks (middle). Reindexing each to contain the numbers 1–4 in the same relative order, which represents the same algorithm (bottom). . . . .	76
4.4.1	Representatives for each of the 8 groups of order-label pairs producing distinct maps $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ . Left: variants of the Pak–Sulzgruber algorithm given by swapping rows 1 and 2, columns 1 and 2, or both in both $\mathcal{O}$ and $\mathcal{L}$ . Right: the same variants, but for the Hillman–Grassl algorithm. In each pair, the order tableau is given on the left and the label tableau on the right. . . . .	79
4.5.1	Variants of row- and column-major order for shape $(4, 3, 2, 1)$ , as defined in Definition 4.5.1. . . . .	84

# List of Tables

4.1	Symmetries of every order-label pair $(\mathcal{O}, \mathcal{L})$ producing one of the eight distinct maps. . . . .	80
4.2	Operations we can apply to the order-label pairs $(\mathcal{O}, \mathcal{L})$ in various groups that preserve the map produced. . . . .	81

# Chapter 1

## Introduction

Plane partitions, along with their many generalizations and their cousins the standard and semistandard Young tableaux, are frequently studied objects in combinatorics (see, for instance, [Bre99]). As is often the case with combinatorial objects of interest, they arise in other areas of mathematics; relevant to our work is a particular use in computing the equivariant Calabi–Yau topological vertex in Pandharipande–Thomas theory and Donaldson–Thomas theory (PT and DT). This generating function is a local contribution to the generating function of DT (resp. PT) invariants for a Calabi–Yau 3-fold with a torus action. One may use Atiyah–Bott localization to reduce the computation to one on the fixed loci of the torus action; the vertex represents the contribution of one fixed point. We refer the interested reader to [Mau+03] for further details. In [PT09], the authors conjecture that two generating functions that count DT and PT objects, which are generalizations of standard and reverse plane partitions, respectively, are equal up to a factor of MacMahon’s function  $M(q)$  (i.e. the generating function for standard plane partitions). The fully general conjecture was proven in [JWY22] using the double-dimer model; however, this last proof is strikingly involved, and no combinatorial proof has been given for the general case or any special case. We give a combinatorial proof for two special cases, with the goal of eventually extending our methods to the fully general case.

The generating function for reverse plane partitions was first derived by Stanley [Sta71], and

later bijectivized by Hillman and Grassl [HG76]. Their map places reverse plane partitions in bijection with hook-length-weighted tableaux of the same shape containing nonnegative integers, and the lack of the plane partition inequalities greatly simplifies the process of working with them. A second bijection was introduced by Pak [Pak01] (and later independently by Sulzgruber [Sul17]), using local operations called *toggles* on the diagonals of reverse plane partitions. These local moves were later independently introduced in [Hop14], and they are the main tool we use.

In Chapter 2, we discuss plane partitions and a pair of vertex operators that allow us to build MacMahon’s generating function  $M(q)$  one diagonal at a time. We examine the toggle operation on diagonals and use it to introduce bijective proofs of the commutation relations of the vertex operators, then give a bijective proof of two special cases of the PT–DT correspondence (so-called *one-leg* and *two-leg* objects).

The fully general (three-leg) case is fundamentally more difficult than the previous two: whereas every object in the one-leg and two-leg correspondences is one modeled by the dimer model, meaning it is in bijection with perfect matchings on a particular variant of the hex graph, one of the types of objects in the three-leg case is a *double-dimer* object instead [Ken03]: it is in correspondence with a vertex pairing on a hex graph in which every vertex is matched with two adjacent ones. In Chapter 3, we thoroughly introduce three-leg objects and prove several structural results about them. The conditions for a three-leg object to be valid traditionally involve a number of complicated global conditions, but these results let us express the validity of certain three-leg objects in terms of only local conditions, the first such description of which we are aware.

The Hillman–Grassl and Pak–Sulzgruber algorithms have substantially different descriptions, but work by Garver and Patrias [GP17] expressed both as variants of a single algorithm, differing only by two permutations. The question of whether other pairs of permutations produce valid algorithms remained open; in Chapter 4, we introduce notation that lets us discuss the Garver–Patrias algorithm in general and prove a number of results about its structure. We then leverage these results to exhaustively but efficiently check all Garver–Patrias algorithms on  $3 \times 3$  tableaux, prove exactly which are valid, and conjecture results

for larger tableaux.

**Acknowledgments:** The authors thank Tatyana Benko and Ava Bamforth for helpful conversations regarding the labeling conditions of 3-leg RPPs, Andy Huchala for assistance in constructing integer programs, and the University of Oregon’s Talapas computing cluster for assistance in large computations.

# Chapter 2

## Vertex Operators and Single-Dimer Objects

### 2.1. Plane Partitions

We begin with straightforward integer partitions, which give rise to Young diagrams and thereby plane partitions. We review several definitions; for further details, we refer the interested reader to [Sta23], [Bre99], or [And84].

**Definition 2.1.1.** A **partition**  $\lambda$  is a sequence of nonnegative integers  $\lambda = (\lambda_1, \lambda_2, \dots)$  such that  $\lambda_i \geq \lambda_{i+1}$  for all  $i \in \mathbb{N}$  and only finitely many  $\lambda_i$  are nonzero<sup>1</sup>. Since every partition ends in zeros, we typically write only the nonzero entries. The **weight** of  $\lambda$  is  $|\lambda| = \sum_{i=1}^{\infty} \lambda_i$ , and the **Young diagram** corresponding to  $\lambda$  is the subset  $\{(i, j) \in \mathbb{N}^2 \mid 1 \leq j \leq \lambda_i\}$ , which we represent as a set of top-left-justified squares whose row lengths weakly decrease. Finally, the **conjugate partition** to  $\lambda$  is the partition  $\lambda'$  given by the column lengths of the Young diagram corresponding to  $\lambda$ .

The following definition is slightly unusual, but it is a special case of a more natural generalization of plane partitions that we introduce later.

---

<sup>1</sup>We use  $\mathbb{N}$  to denote the set  $\{1, 2, \dots\}$  of natural numbers, and  $\mathbb{N}_{\geq 0}$  for  $\{0, 1, 2, \dots\}$ .

**Definition 2.1.2.** Let  $\lambda \subset \mathbb{N}^2$  be a Young diagram. A **Young diagram asymptotic to  $\lambda$**  is the collection of boxes  $\mathbb{N}^2 \setminus \lambda$ .

While this is a slight abuse of notation, since Young diagrams asymptotic to a partition  $\lambda$  are not in fact Young diagrams, it is a convenient definition for objects we will define in Section 2.3. We also recall the usual notion of a hook of a plane partition, which is a right-angled collection of cells extending toward the boundary of a Young diagram.

**Definition 2.1.3.** Let  $\lambda \subset \mathbb{N}^2$  be a Young diagram and let  $(i, j) \in \lambda$ . The **arm** and **leg** of  $(i, j)$  are

$$\begin{aligned}\text{arm}(i, j) &= \{(i, j') \mid j < j' \leq \lambda_i\} \\ \text{leg}(i, j) &= \{(i', j) \mid i < i' \leq \lambda'_j\}.\end{aligned}$$

If  $(i, j) \in \mathbb{N}^2 \setminus \lambda$ , then

$$\begin{aligned}\text{arm}(i, j) &= \{(i, j') \mid \lambda_i < j' < j\} \\ \text{leg}(i, j) &= \{(i', j) \mid \lambda'_j < i' < i\}.\end{aligned}$$

The **hook** of  $(i, j)$  is

$$\text{hook}(i, j) = \text{arm}(i, j) \cup \text{leg}(i, j) \cup \{(i, j)\},$$

and the **hook length** is  $h(i, j) = |\text{hook}(i, j)|$ . We often refer to  **$n$ -hooks** of a diagram, which are hooks with length  $n$ , and the **pivot** of a hook, which is the corner  $(i, j)$  from which it is defined. We will also occasionally have use for the notion of the **content** of a box  $(i, j)$ , which is the quantity  $j - i$ ; since content is constant along diagonals, we often refer to the **content- $c$**  diagonal of a diagram.

With Young diagrams, we can express partitions as both one-dimensional lists of weakly decreasing nonnegative integers and as two-dimensional sets of boxes; our primary object of study generalizes partitions by increasing both of these dimensions by one.

**Definition 2.1.4.** A **plane partition** is a function  $\pi : \mathbb{N}^2 \rightarrow \mathbb{N}_{\geq 0}$  such that  $\pi(i, j) \geq \pi(i+1, j)$  and  $\pi(i, j) \geq \pi(i, j+1)$  for all  $i, j \in \mathbb{N}$  and only finitely many  $\pi(i, j)$  are nonzero. The **weight**

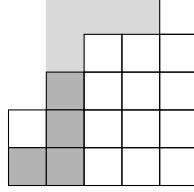


Figure 2.1.1: A 4-hook with pivot  $(5,2)$  in an asymptotic Young diagram  $\mathbb{N}^2 \setminus \lambda$  (dark gray), and a 4-hook with pivot  $(1,2)$  in the Young diagram  $\lambda$  (light gray).

5	4	3	3
4	4	2	
2	1		
2	1		

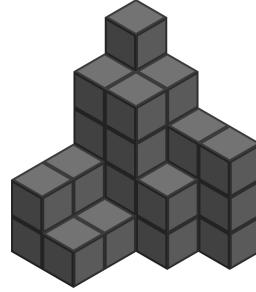


Figure 2.1.2: A plane partition of weight 31, visualized both as a grid of numbers and a stack of 31 blocks.

of  $\pi$  is  $|\pi| = \sum_{i,j \in \mathbb{N}} \pi(i,j)$ . Analogous to a Young diagram, we can associate a plane partition  $\pi$  with the three-dimensional stack of blocks  $\{(i,j,k) \in \mathbb{N}^3 \mid 1 \leq k \leq \pi(i,j)\}$ , as in Figure 2.1.2.

The rows and columns of a plane partition  $\pi$  are themselves interrelated partitions, but the *diagonals* of  $\pi$  turn out to be a more fruitful source of information. We use established techniques and terms in the remainder of this section: the definitions and proofs from here to Section 2.2 are due to [OR05; OR01; Oko03; Oko00; ORV03]. We first define a partial order on partitions that governs when two partitions can sit next to one another in a plane partition.

**Definition 2.1.5.** Let  $\lambda$  and  $\mu$  be partitions. We say  $\lambda$  **interlaces**  $\mu$ , written  $\lambda > \mu$ , if

$$\lambda_1 \geq \mu_1 \geq \lambda_2 \geq \mu_2 \geq \lambda_3 \dots$$

In other words,  $\lambda > \mu$  if and only if  $\lambda$  and  $\mu$  satisfy the plane partition inequalities when written side-by-side as diagonal slices with  $\lambda$  coming first, as in Figure 2.1.3.

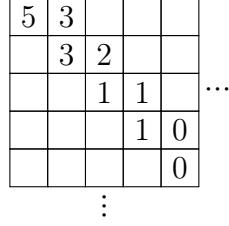


Figure 2.1.3: By placing them as diagonals in a plane partition, we see that  $(5, 3, 1, 1) > (3, 2, 1)$ .

Given a partition  $\mu$ , the set of partitions  $\lambda$  with  $\lambda > \mu$  is a poset product of intervals in  $\mathbb{N}_{\geq 0}$ : each  $\lambda_i$  is bounded below by  $\mu_i$  and above by  $\mu_{i-1}$  (or unbounded for  $i = 1$ ), and critically is independent from every other  $\lambda_i$ . It is largely for this reason that decomposing a plane partition into diagonals is more useful than into rows or columns, and it enables us to express the generating function for plane partitions in terms of operators on the diagonals.

**Definition 2.1.6.** We define a formal  $\mathbb{Q}$ -vector space  $\Lambda$  whose basis consists of vectors  $|\lambda\rangle$  for partitions  $\lambda$ . We also denote corresponding elements of the dual basis by  $\langle\lambda|$ .

On this vector space, we define a weighing operator  $Q(q) : \Lambda \rightarrow \Lambda$ , given by  $Q(q) |\lambda\rangle = q^{|\lambda|} |\lambda\rangle$ . We also define vertex operators  $\Gamma_+$  and  $\Gamma_-$ , each of which accepts a formal variable as a parameter:  $\Gamma_\pm(q) : \Lambda \rightarrow \Lambda$  is given by

$$\begin{aligned}\Gamma_+(q) |\lambda\rangle &= \sum_{\mu > \lambda} q^{|\mu| - |\lambda|} |\mu\rangle \\ \Gamma_-(q) |\lambda\rangle &= \sum_{\mu < \lambda} q^{|\lambda| - |\mu|} |\mu\rangle.\end{aligned}$$

The convention of which operator is denoted  $\Gamma_+$  and which is denoted  $\Gamma_-$  differs between [OR05] and the other cited papers in which these operators appear; our convention matches [OR05]. We think of the subscript + as indicating that  $\Gamma_+$  produces larger partitions, while  $\Gamma_-$  produces smaller ones.

Note that the definition of  $\Gamma_+$  contains an infinite sum, which may fail to converge even in the ring of formal power series; for example,  $\langle \emptyset | \Gamma_-(1)\Gamma_+(1) | \emptyset \rangle$  is not defined. In practice, we avoid these issues by using the weighing operator  $Q$  to produce sums with a formal variable  $q$  that do converge. We refer the interested reader to [Sta11, p. 11–13] for further background.

A product of  $\Gamma$  operators whose arguments are all of the form  $q^i$  for some  $i$  therefore gives a generating function for plane-partition-like objects, since one partition interlacing another is equivalent to the two being able to sit next to one another in a plane partition. The shape of the objects counted is determined by the order of  $\Gamma_+$  operators and  $\Gamma_-$  operators, and its weight is determined by the arguments of the operators; this weight may or may not be equal to the sum of the entries in the object.

These operators are defined and derived in appendix B of [OR05] (and ultimately from [Kac90]) where they are used primarily to compute Schur functions, and they are also used to great effect in [ORV06] to compute the generating functions of various objects. We first demonstrate their use in [ORV06] to express MacMahon's function  $M(q)$ , i.e. the generating function for plane partitions.

Let  $N \in \mathbb{N}$ ; then the generating function for plane partitions whose nonzero entries are contained in the  $N \times N$  square Young diagram is

$$\left\langle \emptyset \left| \left( \prod_{i=1}^N Q \Gamma_-(1) \right) Q \left( \prod_{i=1}^N \Gamma_+(1) Q \right) \right| \emptyset \right\rangle. \quad (2.1)$$

The presence of the weighing operators combined with the lack of variables in the interlacing operators seems to suggest that there is room for improvement in this formula, and indeed the two types of operator have a simple commutation relation that will lead to just that.

**Lemma 2.1.7.** Let  $\lambda$  and  $\mu$  be partitions. Then

$$\begin{aligned} Q(q)\Gamma_+(a) &= \Gamma_+(qa)Q(q) \\ \Gamma_-(a)Q(q) &= Q(q)\Gamma_-(qa). \end{aligned}$$

*Proof.* Fix partitions  $\lambda$  and  $\mu$  with  $\mu > \lambda$ ; then

$$\begin{aligned} \langle \mu | Q(q)\Gamma_+(a) | \lambda \rangle &= q^{|\mu|} a^{|\mu|-|\lambda|} \\ &= (qa)^{|\mu|-|\lambda|} q^{|\lambda|} \\ &= \langle \mu | \Gamma_+(qa)Q(q) | \lambda \rangle, \end{aligned}$$

and similarly for the  $\Gamma_-$  relation.  $\square$

We are now prepared to construct a simpler expression for  $M(q)$ . We commute each  $Q$  outward from the center in Equation (2.1), splitting the middle  $Q$  into  $Q^{1/2}Q^{1/2}$  to preserve symmetry. The  $Q$  operators are annihilated at both the  $|\emptyset\rangle$  and the  $\langle\emptyset|$ , and so Equation (2.1) reduces to

$$\langle \emptyset | \Gamma_- \left( q^{\frac{2N-1}{2}} \right) \cdots \Gamma_- \left( q^{\frac{3}{2}} \right) \Gamma_- \left( q^{\frac{1}{2}} \right) \Gamma_+ \left( q^{\frac{1}{2}} \right) \Gamma_+ \left( q^{\frac{3}{2}} \right) \cdots \Gamma_+ \left( q^{\frac{2N-1}{2}} \right) | \emptyset \rangle.$$

Since any plane partition can have only finitely many nonzero entries, taking the limit as  $N \rightarrow \infty$  produces the following expression for  $M(q)$ :

$$M(q) = \langle \emptyset | \cdots \Gamma_- \left( q^{\frac{3}{2}} \right) \Gamma_- \left( q^{\frac{1}{2}} \right) \Gamma_+ \left( q^{\frac{1}{2}} \right) \Gamma_+ \left( q^{\frac{3}{2}} \right) \cdots | \emptyset \rangle. \quad (2.2)$$

This vertex operator form generalizes to nearly every other object we will discuss, and by understanding local operations on the  $\Gamma$  operators — the content of the next section — we can derive substantially simpler and more useful expressions.

## 2.2. Toggles

Having explored the commutation relations between the  $Q$  and  $\Gamma$  operators, we now turn to how the  $\Gamma$  operators commute with one another. To explain the relationship bijectively, we will require a particular local move called a **toggle**, described independently in [Pak01], [Hop14], and [Bet+14].

**Definition 2.2.1.** Given three partitions  $\lambda > \nu > \mu$ , the toggle of  $\nu$  relative to  $\lambda$  and  $\mu$  is a partition  $T(\nu)$  defined in the following manner:

$$T(\nu)_i = \min \{ \lambda_i, \mu_{i-1} \} + \max \{ \lambda_{i+1}, \mu_i \} - \nu_i,$$

where we take  $\mu_0 = \infty$ . Intuitively, when we write  $\lambda$ ,  $\nu$ , and  $\mu$  side-by-side, each  $\nu_i$  is bounded above by the minimum of the entries immediately above and to the left and below by the maximum of the entries immediately below and to the right; toggling is then just the unique map that is an involution on each entry's poset of possible values.

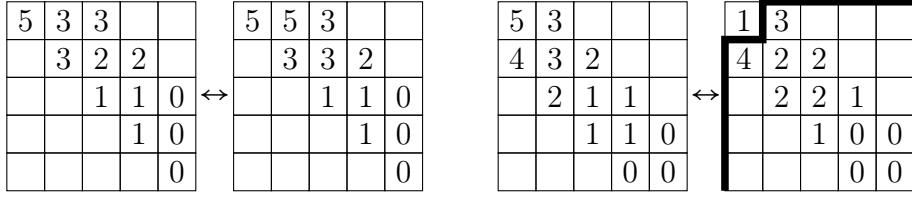


Figure 2.2.1: Toggling  $(3, 2, 1)$  relative to  $(5, 3, 1, 1)$  and  $(3, 2)$  (middle- and far-left), and toggling  $(5, 3, 1, 1)$  relative to  $(3, 2, 1)$  and  $(4, 2, 1)$  (middle- and far-right).

We can also define toggles when the middle partition either interlaces both of the others or is interlaced by them.

**Definition 2.2.2.** If  $\lambda < \nu > \mu$ , then the toggle of  $\nu$  relative to  $\lambda$  and  $\mu$  is a map that produces a pair  $(T(\nu), n)$ , where  $\lambda > T(\nu) < \mu$  and  $n \in \mathbb{N}_{\geq 0}$ . Similarly to Definition 2.2.1,  $T(\nu)$  is given by

$$T(\nu)_i = \min \{\lambda_i, \mu_i\} + \max \{\lambda_{i+1}, \mu_{i+1}\} - \nu_{i-1}$$

for  $i \geq 2$ . We handle  $\nu_1$  separately: we say that the toggle **pops off** the value  $n = \nu_1 - \max \{\lambda_1, \mu_1\}$ . Similarly, if  $\nu$  is a partition with  $\lambda > \nu < \mu$ , and  $n \in \mathbb{N}_{\geq 0}$ , then the toggle of  $\nu$  relative to  $\lambda$  and  $\mu$  sends  $(\nu, n)$  to a partition  $T(\nu, n)$  with  $\lambda < T(\nu, n) > \mu$ , defined by

$$T(\nu, n)_i = \min \{\lambda_i, \mu_i\} + \max \{\lambda_{i+1}, \mu_{i+1}\} - \nu_i$$

for  $i \geq 2$  and  $T(\nu, n)_1 = n + \min \{\lambda_1, \mu_1\}$ .

**Example 2.2.3.** Let  $\lambda = (5, 3, 1, 1)$ ,  $\mu = (3, 2)$ , and  $\nu = (3, 2, 1)$ , so that  $\lambda > \nu > \mu$ . Then the toggle of  $\nu$  relative to  $\lambda$  and  $\mu$  is  $T(\nu) = (5, 3, 1)$ , as shown in the left half of Figure 2.2.1.

On the other hand, with  $\eta = (4, 2, 1)$ ,  $\eta < \lambda > \nu$ , and we can toggle  $\lambda$  relative to  $\eta$  and  $\nu$  to produce the partition  $(2, 2)$  and the popped-off value 1. This is shown in the right half of Figure 2.2.1 — we write the value of 1 where it was popped off and separate it from the remaining diagram by a bold border. As we will soon see, this slightly strange notation will enable us to perform multiple subsequent toggles.

Toggling is a useful local move in many areas of combinatorics surrounding plane partitions and similar objects (for example, they are used to give an alternate definition for the RSK

algorithm in [Hop14]), but we need them to fully explain how the  $\Gamma$  operators commute with one another. The commutation relation  $\Gamma_-(b)\Gamma_+(a) = \frac{1}{1-ab}\Gamma_+(a)\Gamma_-(b)$  is given in [OR05, Appendix B.2], and it turns out to be an algebraic equivalent to toggling, with the particular type of toggle dependent on the signs of the  $\Gamma$  operators.

**Proposition 2.2.4.** Let  $\lambda$  and  $\mu$  be partitions. Then there is a bijection between partitions  $\nu$  with  $\mu < \nu > \lambda$  and pairs  $(\nu', n)$  of partitions  $\nu'$  with  $\mu > \nu' < \lambda$  and nonnegative integers  $n$ , given by toggling  $\nu$  with respect to  $\lambda$  and  $\mu$ . Moreover, the bijection preserves weight in the following manner:

$$\begin{aligned} |\nu| - |\lambda| &= |\lambda| - |T(\nu)| + n \text{ and} \\ |\nu| - |\mu| &= |\mu| - |T(\nu)| + n, \end{aligned} \tag{2.3}$$

where  $n$  is the entry popped off in the toggle.

*Proof.* Given such a  $\nu$ , we first verify that the toggled partition  $T(\nu)$  is interlaced by both  $\lambda$  and  $\mu$ . Let  $i \geq 2$  and consider  $\nu_i$ . Since  $\mu < \nu > \lambda$ , we have the following inequalities:

$$\begin{aligned} \nu_{i-1} &\geq \lambda_{i-1} \\ \vee\! &\quad \vee\! \\ \mu_{i-1} &\geq \nu_i \geq \lambda_i \\ \vee\! &\quad \vee\! \\ \mu_i &\geq \nu_{i+1} \end{aligned}$$

Equivalently,  $\min(\lambda_{i-1}, \mu_{i-1}) \geq \nu_i \geq \max(\lambda_i, \mu_i)$ . By negating the inequality and performing some simple algebraic moves, we find that

$$\begin{aligned} -\min(\lambda_{i-1}, \mu_{i-1}) &\leq -\nu_i \leq -\max(\lambda_i, \mu_i) \\ \Rightarrow 0 &\leq \min(\lambda_{i-1}, \mu_{i-1}) - \nu_i \leq \min(\lambda_{i-1}, \mu_{i-1}) - \max(\lambda_i, \mu_i) \\ \Rightarrow \max(\lambda_i, \mu_i) &\leq \min(\lambda_{i-1}, \mu_{i-1}) + \max(\lambda_i, \mu_i) - \nu_i \leq \min(\lambda_{i-1}, \mu_{i-1}) \\ \Rightarrow \max(\lambda_i, \mu_i) &\leq T(\nu)_{i-1} \leq \min(\lambda_{i-1}, \mu_{i-1}). \end{aligned}$$

Putting this back into grid form, we have that  $\mu > T(\nu) < \lambda$ , as required:

$$\begin{aligned} T(\nu)_{i-2} &\geq \lambda_{i-1} \\ \vee\! &\quad \vee\! \\ \mu_{i-1} &\geq T(\nu)_{i-1} \geq \lambda_i \\ \vee\! &\quad \vee\! \\ \mu_i &\geq T(\nu)_i \end{aligned}$$

We now show that toggling is weight-preserving in the manner specified in Equation (2.3). The crucial observation is that  $\min(\lambda_i, \mu_i) + \max(\lambda_i, \mu_i) = \lambda_i + \mu_i$ . Therefore, the weight of  $T(\nu)$  is

$$\begin{aligned} |T(\nu)| &= \sum_{i \geq 2} \min(\lambda_{i-1}, \mu_{i-1}) + \max(\lambda_i, \mu_i) - \nu_i \\ &= -\min(\lambda_1, \mu_1) + \sum_{i \geq 1} (\lambda_i + \mu_i) - \sum_{i \geq 2} \nu_i \\ &= -\min(\lambda_1, \mu_1) + |\lambda| + |\mu| - (|\nu| - \nu_1) \\ &= |\lambda| + |\mu| - |\nu| + n, \end{aligned}$$

where  $n$  is the entry popped off in the process of toggling. This proves both of the claims regarding weight.  $\square$

We later became aware that this proposition was stated in [Bet+14, Equation 3.1], although they do not use the term *toggle*. A similar commutation relation holds for  $\Gamma$  operators of the same sign and is also given in [OR05, Appendix B.2]:  $\Gamma_+(a)\Gamma_+(b) = \Gamma_+(b)\Gamma_+(a)$ , and identically for  $\Gamma_-$ . Again, we give a bijective proof with toggles.

**Proposition 2.2.5.** Let  $\lambda$  and  $\mu$  be partitions. Then there is a bijection between partitions  $\nu$  with  $\mu > \nu > \lambda$  and partitions  $\nu'$  with  $\mu > \nu' > \lambda$ , given by toggling  $\nu$  with respect to  $\lambda$  and  $\mu$ , and it is weight-preserving in the following manner:

$$\begin{aligned} |\nu| - |\lambda| &= |\mu| - |T(\nu)| \\ |\mu| - |\nu| &= |T(\nu)| - |\lambda| \end{aligned}$$

*Proof.* Given such a  $\nu$ , the interlacing fact is immediate: the toggling operation necessarily preserves both directions of interlacing. For the weight preservation, the toggled partition  $T(\nu)$  is defined by

$$T(\nu)_i = \min(\lambda_i, \mu_{i-1}) + \max(\lambda_{i+1}, \mu_i) - \nu_i,$$

where we take  $\mu_0 = \infty$ , so

$$|T(\nu)| = \sum_i (\lambda_i + \mu_i - \nu_i) = |\lambda| + |\mu| - |\nu|,$$

and so

$$\begin{aligned} b^{|\mu|-|T(\nu)|} a^{|T(\nu)|-|\lambda|} &= b^{|\mu|-|\lambda|-|\mu|+|\nu|} a^{|\lambda|+|\mu|-|\nu|-|\lambda|} \\ &= b^{|\nu|-|\lambda|} a^{|\mu|-|\nu|}, \end{aligned}$$

as required.  $\square$

To apply these propositions to produce explicit bijections on objects counted by generating functions expressed as products of vertex operators, we require one more technical result.

**Lemma 2.2.6.** Let  $z_1(q)$  be a generating function given by

$$z_1(q) = \left\langle \emptyset \left| \prod_{n \in \mathbb{Z}} \Gamma_{s_n}(q^{p_n}) \right| \emptyset \right\rangle,$$

where each  $s_n = \pm 1$ , and we write  $\Gamma_{\pm 1}$  to mean  $\Gamma_{\pm}$ , respectively. Let  $m \in \mathbb{Z}$ , and let  $z_2(q)$  be equal to  $z_1(q)$ , except with the order of  $\Gamma_{s_m}(q^{p_m})$  and  $\Gamma_{s_{m+1}}(q^{p_{m+1}})$  swapped in the product. Let  $S_1$  and  $S_2$  be the sets of objects counted by  $z_1$  and  $z_2$ , with weight marked by  $q$ , as in Definition 2.1.6. Since  $\Gamma_{s_m}(q^{p_m})$  and  $\Gamma_{s_{m+1}}(q^{p_{m+1}})$  are adjacent in  $z_1$ , the objects in  $S_1$  have some diagonal that both of these  $\Gamma$  operators count; call it  $d_m$ . Then there is a weight-preserving bijection  $f : S_1 \rightarrow S_2$ , and it is given by toggling this diagonal  $d_m$  with respect to the two diagonals adjacent to it.

*Proof.* We show the case where  $s_m = -1$  and  $s_{m+1} = 1$ ; the other cases are exactly analogous.

Write  $z_1(q)$  as

$$z_1(q) = \sum_{\mu < \nu > \lambda} \left\langle \emptyset \left| \prod_{n \leq m} \Gamma_{s_n}(q^{p_n}) \right| \nu \right\rangle \left\langle \nu \left| \prod_{n \geq m+1} \Gamma_{s_n}(q^{p_n}) \right| \emptyset \right\rangle.$$

In this presentation,  $\nu$  is the diagonal counted by both  $\Gamma_{s_m}(q^{p_m})$  and  $\Gamma_{s_{m+1}}(q^{p_{m+1}})$ . Applying Proposition 2.2.4, toggling that diagonal gives a bijection between the partitions  $\nu$  counted in the sum to pairs  $(\nu', k)$  of partitions  $\nu'$  satisfying  $\mu > \nu' < \lambda$  and integers  $k \geq 0$  that satisfy Equation (2.3). The generating function that counts these new objects is then

$$\frac{1}{1 - q^{p_m + p_{m+1}}} \sum_{\mu > \nu' < \lambda} \left\langle \emptyset \left| \left( \prod_{n < m} \Gamma_{s_n}(q^{p_n}) \right) \Gamma_{s_{m+1}}(q^{p_{m+1}}) \right| \nu' \right\rangle \left\langle \nu' \left| \Gamma_{s_m}(q^{p_m}) \left( \prod_{n > m+1} \Gamma_{s_n}(q^{p_n}) \right) \right| \emptyset \right\rangle,$$

which is exactly  $z_2(q)$ .  $\square$

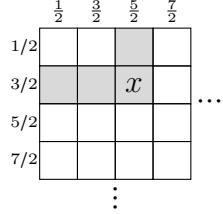


Figure 2.2.2: A cell  $x \in \mathbb{N}^2$  and its hook. The edges of the diagram are labeled with the exponents of the  $\Gamma$  operators they correspond to. The hook of  $x$  intersects the boundary at edges corresponding to operators  $\Gamma_+(q^{5/2})$  and  $\Gamma_-(q^{3/2})$ , and the hook length of  $x$  is  $h(x) = 4 = \frac{5}{2} + \frac{3}{2}$ .

This lemma bijectivizes the commutation of the  $\Gamma$  operators: given a vertex operator expression for a generating function  $f(q)$ , the commutation of two same-sign operators to form a new generating function  $g(q)$  is a weight-preserving bijection of the objects counted by  $f$  to those counted by  $g$ , given by toggling the diagonal corresponding to the commuted operators. Similarly, commuting a  $\Gamma_+$  to the left past a  $\Gamma_-$  is a weight-preserving bijection given by toggling, and it also produces a nonnegative integer based on the arguments of the  $\Gamma$  operators. These bijections form the bedrock of many of our subsequent results: if the generating function for a plane-partition-like object has a vertex operator expansion and an identity can be proven algebraically by performing successive commutations, then it can be bijectivized by interpreting each successive commutation as a toggle of a particular diagonal. Our first application of this method is to compute a bijectivization of the product expansion of MacMahon's function.

**Theorem 2.2.7.** There is a weight-preserving bijection  $\tau$  between plane partitions and tableaux of shape  $\mathbb{N}^2$  (i.e. functions  $\mathbb{N}^2 \rightarrow \mathbb{N}_{\geq 0}$  with no restrictions on outputs) that are weighted by hook length.

*Proof.* Our bijection is functionally identical to the Pak–Sulzgruber algorithm [Pak01; Sul17], except described on standard plane partitions instead of reverse ones. What is new is the method by which we derive it, which is generalizable to the objects we discuss in future sections. We follow the methods in [ORV06, page 11] used to prove that the vertex operator

expansion

$$M(q) = \langle \emptyset | \cdots \Gamma_-(q^{5/2}) \Gamma_-(q^{3/2}) \Gamma_-(q^{1/2}) \Gamma_+(q^{1/2}) \Gamma_+(q^{3/2}) \Gamma_+(q^{5/2}) \cdots | \emptyset \rangle, \quad (2.4)$$

can be iteratively commuted into the expression

$$M(q) = \prod_{\square \in \mathbb{N}^2} \frac{1}{1 - q^{h(\square)}}.$$

Consider the first commutation in Equation (2.4). Define

$$f_0 = \langle \emptyset | \cdots \Gamma_-(q^{5/2}) \Gamma_-(q^{3/2}) \Gamma_-(q^{1/2}) \Gamma_+(q^{1/2}) \Gamma_+(q^{3/2}) \Gamma_+(q^{5/2}) \cdots | \emptyset \rangle = M(q)$$

and

$$f_1 = \langle \emptyset | \cdots \Gamma_-(q^{5/2}) \Gamma_-(q^{3/2}) \Gamma_+(q^{1/2}) \Gamma_-(q^{1/2}) \Gamma_+(q^{3/2}) \Gamma_+(q^{5/2}) \cdots | \emptyset \rangle,$$

where the middle two operators have been commuted. Denote the sets of objects counted by these two generating functions by  $P_0$  and  $P_1$ , respectively. While  $P_0$  is simply the set of plane partitions, weighted as usual,  $P_1$  is less well-behaved. Its elements are of the form of the far-right object in Figure 2.2.1: each is a placement of nonnegative integers into  $\mathbb{N}^2 \setminus \{(1, 1)\}$  that weakly decrease along rows and columns, and the weight of such an object is given by  $f_1$ . By Lemma 2.2.6, the map  $\tau_0 : P_0 \rightarrow P_1 \times \mathbb{N}_{\geq 0}$  given by toggling the main diagonal is a bijection.

Define  $f_n$ ,  $P_n$ , and  $\tau_n : P_n \rightarrow P_{n+1} \times \mathbb{N}_{\geq 0}$  for  $n \geq 1$  similarly; we may enumerate them in any order compatible with the order in which corners appear when toggling. We will define our bijection  $\tau$  from the set of plane partitions to the set of tableaux of shape  $\mathbb{N}^2$  effectively by composing every  $\tau_k$ , but the notation makes this slightly cumbersome. Let  $(s_n) \subset \mathbb{N}^2$  be the sequence of cells such that  $s_n$  is popped off by  $\tau_n$ . The sequence depends on our ordering of the  $\tau_n$ ; one possible ordering is by off-diagonals, i.e.

$$(s_n)_{n \in \mathbb{N}} = ((1, 1), (1, 2), (2, 1), (1, 3), (2, 2), (3, 1), (1, 4), \dots),$$

since each subsequent off-diagonal consists entirely of corners after every cell in the previous off-diagonal has been popped off. More generally, we may choose any ordering so that for any  $k \in \mathbb{N}$ , the tableau

$$t : \{s_n \mid n \in \{1, 2, \dots, k\}\} \rightarrow \mathbb{N}$$

given by  $t(s_n) = n$  is a standard Young tableau.

Let  $B$  be the set of tableaux of shape  $\mathbb{N}^2$  (i.e. functions  $\mathbb{N}^2 \rightarrow \mathbb{N}_{\geq 0}$ ) and define functions  $\tau'_n : P_n \times B \rightarrow P_{n+1} \times B$ , where  $\tau'_n$  applies  $\tau_n$  to its first argument and leaves its second argument unchanged, except for setting the value in cell  $s_n$  to the value popped off by  $\tau_n$ . For a plane partition  $\pi$  and the zero tableau 0, the composition  $(\cdots \circ \tau'_2 \circ \tau'_1 \circ \tau'_0)(\pi, 0)$  then produces a tuple  $(\emptyset, \beta)$  for a tableau  $\beta$  of shape  $\mathbb{N}^2$ ; we define the map  $\tau : P_0 \rightarrow B$  by setting  $\tau(\pi) = \beta$ . This map  $\tau$  is well-defined since every plane partition has only finitely many nonzero entries — given a plane partition, we may stop toggling diagonals once there are no longer any nonzero entries left. Moreover, since each  $\tau_i$  is a bijection,  $\tau$  is also a bijection.

It remains to show that  $\tau$  is weight-preserving when its output tableaux are weighted by hook length. Specifically, if  $\tau'_n(\pi, \beta) = (\pi', \beta')$  and the number popped off in the toggle is  $a$ , then we wish to show that  $|\pi| = |\pi'| + a \cdot h(s_n)$ , where the weights of  $\pi$  and  $\pi'$  are as measured by  $f_n$  and  $f_{n+1}$ , respectively. Now  $f_n$  and  $f_{n+1}$  differ only by a single commutation that moves some  $\Gamma_-(q^i)$  to the right past some  $\Gamma_+(q^j)$ , producing a factor of  $\frac{1}{1-q^{i+j}}$  that corresponds to the number popped off in the toggle. Our claim of weight preservation then reduces to the claim that  $h(s_n) = i + j$ .

Set  $s_n = (k, l)$ . In the original square diagram,  $\text{hook}(s_n)$  intersects the boundary of  $\mathbb{N}^2$  at a horizontal edge corresponding to  $\Gamma_+(q^{l-\frac{1}{2}})$  and a vertical one corresponding to  $\Gamma_-(q^{k-\frac{1}{2}})$ , as in Figure 2.2.2. When we toggle a diagonal and commute a  $\Gamma_-$  to the right past a  $\Gamma_+$ , the corresponding edges swap places in the diagram — in effect, the horizontal edge corresponding to  $\Gamma_+$  moves down, and the vertical edge corresponding to  $\Gamma_-$  moves right. Therefore, no matter the order in which we toggle diagonals before reaching  $s_n$ , the horizontal edge corresponding to  $\Gamma_+(q^{l-\frac{1}{2}})$  is always above  $s_n$  and the vertical one corresponding to  $\Gamma_-(q^{k-\frac{1}{2}})$  is always to its left. When we finally toggle the diagonal containing  $s_n$  after  $n - 1$  previous toggles, those two edges are bordering  $s_n$ , and so the values of  $i$  and  $j$  from the previous paragraph are in fact  $k - \frac{1}{2}$  and  $l - \frac{1}{2}$ . Since  $h(s_n) = k + l - 1 = i + j$ , our claim is proven.  $\square$

In short, this map  $\tau$  sends a plane partition to a tableau weighted by hook length, given

<table border="1"> <tr><td>3</td><td>1</td><td>0</td></tr> <tr><td>2</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td style="text-align: center;">⋮</td><td></td><td></td></tr> </table> ...	3	1	0	2	1	0	0	0	0	⋮			<table border="1"> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td style="background-color: black; color: white;">2</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td style="text-align: center;">⋮</td><td></td><td></td></tr> </table> ...	1	1	0	2	0	0	0	0	0	⋮			<table border="1"> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td style="background-color: black; color: white;">2</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td style="text-align: center;">⋮</td><td></td><td></td></tr> </table> ...	1	1	0	2	0	0	0	0	0	⋮			<table border="1"> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td style="background-color: black; color: white;">2</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td style="text-align: center;">⋮</td><td></td><td></td></tr> </table> ...	1	1	0	2	0	0	0	0	0	⋮		
3	1	0																																																	
2	1	0																																																	
0	0	0																																																	
⋮																																																			
1	1	0																																																	
2	0	0																																																	
0	0	0																																																	
⋮																																																			
1	1	0																																																	
2	0	0																																																	
0	0	0																																																	
⋮																																																			
1	1	0																																																	
2	0	0																																																	
0	0	0																																																	
⋮																																																			

Figure 2.2.3: A weight-7 plane partition  $\pi$  (far left) being mapped bijectively to a weight-7 tableau  $\tau(\pi)$  that is weighted by hook length (far right). The center-left and center-right figures are the first two steps in the bijection.

by toggling diagonals until the diagram is empty (i.e. the  $\Gamma$  operators have been completely commuted) and recording the popped numbers in the locations from which they were removed. The map is functionally identical to that described in [Pak01; Sul17], but the vertex operator description provides an alternate lens through which to view the bijection and a clear explanation of why it is independent of the order in which we toggle diagonals (specifically, the commutators of  $\Gamma$  operators that are produced can be freely factored out of the entire expression).

**Example 2.2.8.** Let  $\pi$  be the plane partition given in Figure 2.2.3. In the expression for  $M(q)$  given in Equation (2.4),  $\pi$  is represented by a term of  $q^7$ . Commuting the  $\Gamma_-(q^{1/2})$  with the  $\Gamma_+(q^{1/2})$  results in

$$M(q) = \frac{1}{1-q} \langle \emptyset | \cdots \Gamma_-(q^{5/2}) \Gamma_-(q^{3/2}) \Gamma_+(q^{1/2}) \Gamma_-(q^{1/2}) \Gamma_+(q^{3/2}) \Gamma_+(q^{5/2}) \cdots | \emptyset \rangle,$$

and the term of  $q^7$  now represents the object second from left in Figure 2.2.3: a  $1 \times 1$  tableau with the entry 1, and an object in  $P_1$ . We draw the two superimposed with a bold dividing border to emphasize how the process preserves the overall shape of  $\pi$ . The  $1 \times 1$  tableau is weighted by hook length (i.e. it has weight 1), and the object on the right has weight 6 as counted by  $f_1$  (in the language of Definition 2.1.6), so the combined weight of 7 is preserved. Note that this weight of 6 is *not* the sum of the entries; in general, only the starting plane partition has weight equal to the sum of its entries.

We now have two choices of corner to commute — if we choose to commute  $\Gamma_-(q^{3/2})$  with  $\Gamma_+(q^{1/2})$ , the resulting pair of objects is second from right in Figure 2.2.3. The tableau now has weight 5, since the box containing 2 has hook length 2, while the object in  $P_2$  has

weight 2 as counted by  $f_2$ . After additionally toggling the diagonal containing the 1, there are no more nonzero entries in the plane-partition-like object, and so all future toggles place a zero into the tableau. The result is the final tableau  $\tau(\pi)$  on the far right, whose weight (accounting for hook length) is correctly equal to 7.

## 2.3. One-Leg Objects

The techniques of the previous section apply to far more than just plane partitions: any object whose generating function is expressible as a product of these  $\Gamma$  operators is a candidate for a decomposition given by iterative toggling. We begin with two definitions of plane-partition-like objects — the standard notion of reverse plane partitions, along with a less standard notion of a skew plane partition (see [ORV06, Section 3.4]) — before connecting them with a result that we bijectivize.

**Definition 2.3.1.** Let  $\lambda \subset \mathbb{N}^2$  be a Young diagram.

- 1 A **reverse plane partition**, or RPP, of shape  $(\emptyset, \emptyset, \lambda)$  is a function  $\rho : \lambda \rightarrow \mathbb{N}_{\geq 0}$  such that  $\rho(i, j) \leq \rho(i + 1, j)$  and  $\rho(i, j) \leq \rho(i, j + 1)$  for all choices  $(i, j)$  where those quantities are defined. We write the generating function for RPPs of shape  $(\emptyset, \emptyset, \lambda)$  as  $W_{(\emptyset, \emptyset, \lambda)}(q)$ , where  $q$  marks the weight (i.e. the sum of all entries), following the notation of [PT09].
- 2 A **skew plane partition**, or SPP, of shape  $(\emptyset, \emptyset, \lambda)$  is a function  $\sigma : \mathbb{N}^2 \setminus \lambda \rightarrow \mathbb{N}_{\geq 0}$  such that  $\sigma(i, j) \geq \sigma(i + 1, j)$  and  $\sigma(i, j) \geq \sigma(i, j + 1)$  for all  $(i, j) \in \mathbb{N}^2 \setminus \lambda$  and only finitely many  $\sigma(i, j)$  are nonzero. We write the generating function for SPPs of shape  $(\emptyset, \emptyset, \lambda)$  as  $V_{(\emptyset, \emptyset, \lambda)}(q)$ , where  $q$  once again marks the weight.

We call these **one-leg** objects, again following [PT09], since only one of the three entries is nonempty in the shape term  $(\emptyset, \emptyset, \lambda)$ . The notation suggests further generalizations are possible, and this is indeed the case; we discuss the case where at most two of the entries

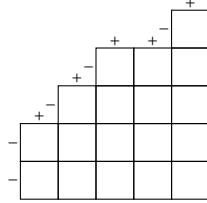


Figure 2.3.1: The signs of the  $\Gamma$  operators for  $\lambda = (4, 2, 1)$ .

of the shape term are nonempty in Section 2.4, and we approach the fully general case in Chapter 3.

For these one-leg objects, the PT–DT correspondence states that

$$V_{(\emptyset, \emptyset, \lambda)}(q) = M(q)W_{(\emptyset, \emptyset, \lambda)}(q), \quad (2.5)$$

or that combinatorially, every SPP of weight  $n$  and shape  $(\emptyset, \emptyset, \lambda)$  corresponds to an RPP of shape  $(\emptyset, \emptyset, \lambda)$  and a plane partition whose weights sum to  $n$ . This equation was stated in [PT09]; throughout this section, we supply details that were omitted and bijectivize the equation.

We begin by focusing on skew plane partitions. To express  $V_{(\emptyset, \emptyset, \lambda)}(q)$  in terms of the  $\Gamma$  operators, the middle section of the sequence of  $\Gamma$ s contains a mix of  $\Gamma_+$  and  $\Gamma_-$  according to the sequence of horizontal and vertical edges in the diagram. With  $\lambda = (4, 2, 1)$  for instance, the sequence of signs of the  $\Gamma$  operators is

$$(\dots, -, -, +, -, +, -, +, +, -, +, \dots)$$

when read from bottom-left to top-right, determined from the order of edges in Figure 2.3.1. The generating function  $V_{(\emptyset, \emptyset, \lambda)}$  then has the following expression. To make this sequence more legible, the  $Q$  operator corresponding to the middle diagonal of the diagram is bolded.

$$\langle \emptyset | \dots Q \Gamma_-(1) Q \Gamma_+(1) Q \Gamma_-(1) Q \Gamma_+(1) \mathbf{Q} \Gamma_-(1) Q \Gamma_+(1) Q \Gamma_-(1) Q \Gamma_+(1) Q \dots | \emptyset \rangle,$$

After commuting every  $Q$  outward, this results in

$$\langle \emptyset | \dots \Gamma_-\left(q^{\frac{7}{2}}\right) \Gamma_+\left(q^{-\frac{5}{2}}\right) \Gamma_-\left(q^{\frac{3}{2}}\right) \Gamma_+\left(q^{-\frac{1}{2}}\right) \Gamma_-\left(q^{-\frac{1}{2}}\right) \Gamma_+\left(q^{\frac{3}{2}}\right) \Gamma_+\left(q^{\frac{5}{2}}\right) \Gamma_-\left(q^{-\frac{7}{2}}\right) \Gamma_+\left(q^{\frac{9}{2}}\right) \dots | \emptyset \rangle.$$

The pattern of increasing odd-numerator fractions of 2 persists, but now the “out of place”  $\Gamma$  operators have negative-exponent arguments. This construction will prove quite useful, and it will be helpful to formalize it.

**Definition 2.3.2.** Let  $\lambda \subset \mathbb{N}^2$  be a Young diagram and label the edges of  $\mathbb{N}^2 \setminus \lambda$  from bottom-left to top-right with consecutive integers so that the two edges adjacent to the content-0 diagonal are labeled  $-1$  and  $0$ . The **edge sign sequence** of  $\lambda$ , denoted  $e_\lambda$ , is the doubly infinite sequence whose entries are  $\pm 1$ , where  $e_\lambda(n) = 1$  if the edge labeled  $n$  is horizontal and  $-1$  if it is vertical. We also define the **edge power sequence** of  $\lambda$  as  $p_\lambda(n)$ , where

$$p_\lambda(n) = \begin{cases} |n + \frac{1}{2}|, & e_\lambda(n) = \text{sign}(n + \frac{1}{2}) \\ -|n + \frac{1}{2}|, & e_\lambda(n) \neq \text{sign}(n + \frac{1}{2}) \end{cases}.$$

The edge sign sequence is identical to the edge sequence defined in [BCY10, Definition 7.3.1], except with opposite signs (the authors also use the opposite convention of  $\Gamma_+$  and  $\Gamma_-$ ). It is nearly identical to the notion of encoded shape in [Bet+14, Figure 3]. It also bears a resemblance to the word associated with steep domino tilings in [BCC14, Proposition 1], as well as the sign sequence in [Bou+15, Definition 2.1]; both also encode binary data related to square-shaped objects into signs.

Given a Young diagram  $\lambda$ , its edge sign sequence  $e(n) = e_\lambda(n)$ , and its edge power sequence  $p(n) = p_\lambda(n)$ , the generating function  $V_{(\emptyset, \emptyset, \lambda)}(q)$  is

$$V_{(\emptyset, \emptyset, \lambda)}(q) = \left\langle \emptyset \left| \prod_{n \in \mathbb{Z}} \Gamma_{e(n)}(q^{p(n)}) \right| \emptyset \right\rangle, \quad (2.6)$$

where  $\Gamma_1$  and  $\Gamma_{-1}$  are written to mean  $\Gamma_+$  and  $\Gamma_-$ , respectively. The exponents serve a greater purpose than merely defining the shape — they have a more direct interpretation in terms of hook lengths in the tableau. We first prove a technical lemma regarding the exponent sign sequence, and then a more substantial result.

**Lemma 2.3.3.** Let  $\lambda \subset \mathbb{N}^2$  be a Young diagram with edge power sequence  $p_\lambda(n)$ , let  $b$  be a corner of  $\lambda$  (i.e. a box  $(i, j)$  with  $(i + 1, j) \notin \lambda$  and  $(i, j + 1) \notin \lambda$ ), and suppose that the bottom and right edges of  $b$  have labels  $p_\lambda(k - 1)$  and  $p_\lambda(k)$  for some  $k \in \mathbb{Z}$ . Let  $\mu$  be the

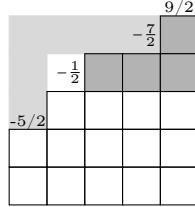


Figure 2.3.2: Hooks and their boundary edges in  $\lambda = (4, 2, 1)$  and the Young diagram asymptotic to it.

Young diagram given by removing  $b$  from  $\lambda$ , so that the left and top edges of  $b$  have labels  $p_\mu(k-1)$  and  $p_\mu(k)$ . Then  $p_\mu(k-1) = p_\lambda(k) + 1$ .

*Proof.* All edge power sequences are the same in absolute value, so  $|p_\mu(k-1)| = |p_\lambda(k-1)|$  and  $|p_\mu(k)| = |p_\lambda(k)|$ . Since the two edge sign sequences  $e_\lambda$  and  $e_\mu$  differ only in that  $e_\mu(k-1) = -e_\lambda(k-1)$  and  $e_\mu(k) = -e_\lambda(k)$ , we must have that  $p_\mu(k-1) = -p_\lambda(k-1)$  and  $p_\mu(k) = -p_\lambda(k)$ .

To finish the proof, we relate  $p_\lambda(k-1)$  to  $p_\lambda(k)$ . We know  $e_\lambda(k-1) = 1$  and  $e_\lambda(k) = -1$  since  $b$  is a corner, and what remains is a straightforward computation with three cases. If  $k = 0$ , then  $p_\lambda(k-1) = p_\lambda(k) = -\frac{1}{2}$ . If  $k > 0$ , then  $p_\lambda(k-1) = k - 1 + \frac{1}{2}$  and  $p_\lambda(k) = -(k + \frac{1}{2})$ . Finally, if  $k < 0$ , then  $p_\lambda(k-1) = -(k - 1 + \frac{1}{2})$  and  $p_\lambda(k) = k + \frac{1}{2}$ . In every case,  $p_\lambda(k-1) + p_\lambda(k) = -1$ . Therefore,

$$\begin{aligned} p_\mu(k-1) &= -p_\lambda(k-1) \\ &= -(-1 - p_\lambda(k)) \\ &= p_\lambda(k) + 1, \end{aligned}$$

as required.  $\square$

**Theorem 2.3.4.** Let  $\lambda \subset \mathbb{N}^2$  be a Young diagram with edge power sequence  $p_\lambda(n)$ , let  $(i, j) \in \mathbb{N}^2 \setminus \lambda$ , and let  $k$  and  $l$  be the labels of the vertical edge in row  $i$  and the horizontal edge in column  $j$  of the boundary of  $\lambda$ , respectively. Then the hook length  $h(i, j)$  satisfies the formula

$$p_\lambda(k) + p_\lambda(l) = h(i, j)$$

If  $(i, j) \in \lambda$  and  $k$  and  $l$  are as before, then the same result holds, but with an additional

sign:

$$p_\lambda(k) + p_\lambda(l) = -h(i, j).$$

In Figure 2.3.2, for example, the hook with pivot  $(2, 5)$  has boundary edges with labels  $-\frac{1}{2}$  and  $\frac{9}{2}$  and hook length  $-\frac{1}{2} + \frac{9}{2} = 4$ , and the hook with pivot  $(1, 1)$  has boundary edges with labels  $-\frac{5}{2}$  and  $-\frac{7}{2}$  and hook length  $-\left(-\frac{5}{2} - \frac{7}{2}\right) = 6$ .

*Proof.* We prove the result by induction on the size of  $\lambda$  and write  $h_\lambda$  for the hook length function since the induction will require us to consider diagrams of different shape. When  $\lambda = \emptyset$ , there are no boxes inside of  $\lambda$ , so we need only show the result for boxes outside. Given  $(i, j) \in \mathbb{N}^2$ , its hook length is  $h_\lambda(i, j) = i + j - 1$ , and its hook meets the boundary of  $\lambda$  — i.e. the boundary of  $\mathbb{N}^2$  — at a vertical edge with label  $k = -i$  and a horizontal edge with label  $l = j - 1$  (recall that the edges bordering the content-0 diagonal have labels  $-1$  and  $0$ ). Both  $p_\lambda(k)$  and  $p_\lambda(l)$  are then positive, and in particular,

$$\begin{aligned} p_\lambda(k) + p_\lambda(l) &= \left| -i + \frac{1}{2} \right| + \left| j - 1 + \frac{1}{2} \right| \\ &= -\left( -i + \frac{1}{2} \right) + \left( j - \frac{1}{2} \right) \\ &= i + j - 1 \\ &= h_\lambda(i, j). \end{aligned}$$

Now suppose the proposition holds for Young diagrams with at most  $n - 1$  boxes, let  $\lambda$  be one with  $n$  boxes, and let  $(i, j) \in \mathbb{N}^2 \setminus \lambda$ . If  $\lambda_i = \lambda'_j = 0$  (i.e. the hook meets the boundary of  $\mathbb{N}^2$  itself), then

$$p_\lambda(k) + p_\lambda(l) = i + j - 1 = h_\lambda(i, j)$$

by identical logic to the base case. Otherwise, suppose without loss of generality that  $\lambda_i \neq 0$ . Then the left leg of the hook of  $(i, j)$  meets the boundary of  $\lambda$  at the box  $b = (i, \lambda_i)$ . If  $\lambda_{i+1} < \lambda_i$ , then  $b$  is a corner, and we may remove it from the Young diagram to produce a diagram  $\mu$  with  $n - 1$  boxes. By the induction hypothesis,

$$p_\mu(k - 1) + p_\mu(l) = h_\mu(i, j).$$

Now  $h_\mu(i, j) = h_\lambda(i, j) + 1$ , and by Lemma 2.3.3,  $p_\mu(k - 1) = p_\lambda(k) + 1$ . Moreover, the edge

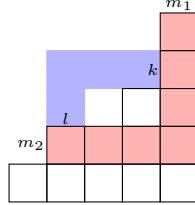


Figure 2.3.3: A hook inside  $\lambda = (4, 4, 3, 1)$  (blue) and a corresponding hook in  $\mathbb{N}^2 \setminus \lambda$  intersecting its boundary edges.

labeled  $l$  in  $\lambda$  is unchanged in  $\mu$  and still labeled  $l$ , so  $p_\mu(l) = p_\lambda(l)$ . In total,

$$p_\lambda(k) + 1 + p_\lambda(l) = h_\lambda(i, j) + 1,$$

proving the result.

If  $b = (i, \lambda_i)$  is not a corner, the argument is much simpler: let  $b'$  be the corner in the same column in  $b$ , which is then necessarily strictly below  $b$ , and let  $\mu$  be the Young diagram given by removing it. Then the induction hypothesis guarantees that

$$p_\mu(k) + p_\mu(l) = h_\mu(i, j),$$

but  $p_\mu(k) = p_\lambda(k)$ ,  $p_\mu(l) = p_\lambda(l)$ , and  $h_\mu(i, j) = h_\lambda(i, j)$ .

It remains to show that the formula holds for a box  $(i, j) \in \lambda$ . The labels  $k$  and  $l$  are now on the vertical edge of  $(i, \lambda_i + 1)$  and the horizontal edge of  $(\lambda'_j + 1, j)$ , respectively. Both of those boxes, as well as  $(\lambda'_j + 1, \lambda_i + 1)$ , lie outside  $\lambda$ , so the first part of the proposition applies to them. Suppose the horizontal and vertical boundary edges of the hook corresponding to the box  $(\lambda'_j + 1, \lambda_i + 1)$  are labeled  $m_1$  and  $m_2$ , respectively, as in Figure 2.3.3. Then

$$\begin{aligned} h_\lambda(i, \lambda_i + 1) &= p_\lambda(k) + p_\lambda(m_1) \\ h_\lambda(\lambda'_j + 1, j) &= p_\lambda(m_2) + p_\lambda(l) \\ h_\lambda(\lambda'_j + 1, \lambda_i + 1) &= p_\lambda(m_2) + p_\lambda(m_1). \end{aligned}$$

On the other hand,

$$\begin{aligned} h_\lambda(\lambda'_j + 1, \lambda_i + 1) - h_\lambda(i, \lambda_i + 1) - h_\lambda(\lambda'_j + 1, j) &= (\lambda'_j + 1 - i - 1) + (\lambda_i + 1 - j - 1) + 1 \\ &= (\lambda'_j - i) + (\lambda_i - j) + 1 \\ &= h_\lambda(i, j). \end{aligned}$$

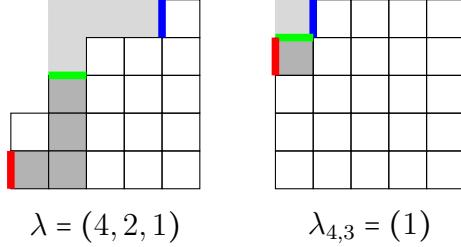


Figure 2.3.4: A 4-hook in a skew plane partition of shape  $(\emptyset, \emptyset, \lambda)$  and the corresponding 4-hook in  $\lambda$  (left). The corresponding inner and outer corners in the 4-quotient  $\lambda_{4,3}$  (right).

Replacing every hook length expression with a sum of entries in the edge power sequence results in

$$\begin{aligned} p_\lambda(m_1) + p_\lambda(m_2) - (p_\lambda(k) + p_\lambda(m_1) + p_\lambda(l) + p_\lambda(m_2)) &= h_\lambda(i, j) \\ \Rightarrow h_\lambda(i, j) &= -p_\lambda(k) - p_\lambda(l), \end{aligned}$$

as required.

This shows the result for every box in  $\lambda$ , proving the proposition.  $\square$

In the Young diagram asymptotic to the empty partition (i.e.  $\mathbb{N}^2$ ), there are exactly  $n$  distinct  $n$ -hooks: their pivots are the boxes along the off-diagonal from  $(n, 1)$  to  $(1, n)$ . In the more general case of a Young diagram asymptotic to a partition  $\lambda$ , the number of  $n$ -hooks depends also on the number of (down-right)  $n$ -hooks in  $\lambda$ . The following proposition uses the standard notion of  $n$ -quotients of a partition; for a thorough reference, see e.g. [Def+22]. Here, we give a brief treatment of  $n$ -quotients that integrates well with our notion of edge sign sequence.

**Definition 2.3.5.** Let  $\lambda$  be a partition. Given the edge sign sequence  $e_\lambda(m)$  for  $m \in \mathbb{Z}$ , the subsequence  $e_\lambda(nm + i)$  consisting of every  $n$ th edge sign from  $e_\lambda(m)$  corresponds to a distinct partition  $\lambda_{n,i}$  for each  $i \in \{0, \dots, n-1\}$ , called the  **$n$ -quotients** of  $\lambda$ . For example, the object on the right in Figure 2.3.4 is the 4-quotient that selects the red, green, and blue edges.

**Proposition 2.3.6.** Let  $\lambda$  be a partition and let  $n \in \mathbb{N}$ . If there are  $k$  distinct  $n$ -hooks in  $\lambda$ , then there are  $n+k$  distinct  $n$ -hooks in the Young diagram asymptotic to  $\lambda$ .

*Proof.* Every  $n$ -hook in  $\lambda$  corresponds to a  $+$  in  $e_\lambda$  followed by a  $-$  exactly  $n$  terms later, and every  $n$ -hook in  $\mathbb{N}^2 \setminus \lambda$  corresponds to a  $+$  edge followed  $n$  terms later by a  $-$ . For example, let  $\lambda = (4, 2, 1)$ . Then the box  $(1, 2) \in \lambda$  has hook length 4 and corresponds to the green and blue edges in Figure 2.3.4, and the box  $(1, 5)$  also has hook length 4 and corresponds to the green and red edges. There is then a one-to-one correspondence between corners of  $\lambda_{n,i}$  for all  $i$  and  $n$ -hooks in  $\lambda$ , and similarly between corners of  $\mathbb{N}^2 \setminus \lambda_{n,i}$  and  $n$ -hooks in  $\mathbb{N}^2 \setminus \lambda$ .

On the other hand, every Young diagram asymptotic to any partition  $\mu$  contains exactly one more corner than  $\mu$ , and so fixing  $i$ , there is a one-to-one correspondence between corners of  $\lambda_{n,i}$  and all the corners of  $\mathbb{N}^2 \setminus \lambda_{n,i}$  except for one. Since the two types of corners alternate from bottom-left to top-right, we can make this correspondence explicit by associating each corner of  $\lambda_{n,i}$  with the corner of  $\mathbb{N}^2 \setminus \lambda_{n,i}$  immediately preceding it in the edge sign sequence. For example, the light and dark gray corners in  $\lambda_{4,3}$  in Figure 2.3.4 are in correspondence.

In total, each of the  $k$  distinct  $n$ -hooks in  $\lambda$  corresponds to a corner in  $\lambda_{n,i}$  for a unique choice of  $i$ , which corresponds to a corner in  $\mathbb{N}^2 \setminus \lambda_{n,i}$  and therefore an  $n$ -hook in  $\mathbb{N}^2 \setminus \lambda$ . However, there is exactly one unmatched corner in  $\mathbb{N}^2 \setminus \lambda_{n,i}$  for each  $i$ , each of which contributes another  $n$ -hook in  $\mathbb{N}^2 \setminus \lambda$ , resulting in a total of  $n + k$  distinct  $n$  hooks in  $\mathbb{N}^2 \setminus \lambda$ .  $\square$

We aim to bijectivize the one-leg PT–DT correspondence Equation (2.5) on the level of hooks; that is, to decompose every object involved into a hook-length-weighted tableau and place the resulting entries in bijection with one another. To that end, we define a map associating the  $n + k$  distinct  $n$ -hooks of the asymptotic Young diagram  $\mathbb{N}^2 \setminus \lambda$  to the  $n$  distinct  $n$ -hooks of the asymptotic Young diagram  $\mathbb{N}^2$  and the  $k$  distinct  $n$ -hooks of the Young diagram  $\lambda$ .

**Definition 2.3.7.** For a box  $b \in \mathbb{N}^2 \setminus \lambda$  with hook length  $h(b) = n$ , let  $c$  be the unique corner in a quotient  $\mathbb{N}^2 \setminus \lambda_{n,i}$  that corresponds to  $b$ , and define  $\varphi_\lambda(b)$  in the following manner. If  $c$  is the upper-right-most corner in  $\mathbb{N}^2 \setminus \lambda_{n,i}$ , then  $\varphi_\lambda(b)$  is the unique box in  $\mathbb{N}^2$  with hook length  $n$  that corresponds to the corner in the  $i$ th  $n$ -quotient of  $\mathbb{N}^2$  (i.e. the  $i$ th box in the  $n$ th off-diagonal of  $\mathbb{N}^2$  when reading from bottom-left to top-right). Otherwise,  $\varphi_\lambda(b)$  is the unique box in  $\lambda$  corresponding to the corner in  $\lambda_{n,i}$  immediately following  $c$  in the alternating sequence of corners inside and outside of  $\lambda_{n,i}$  when reading from bottom-left to top-right.

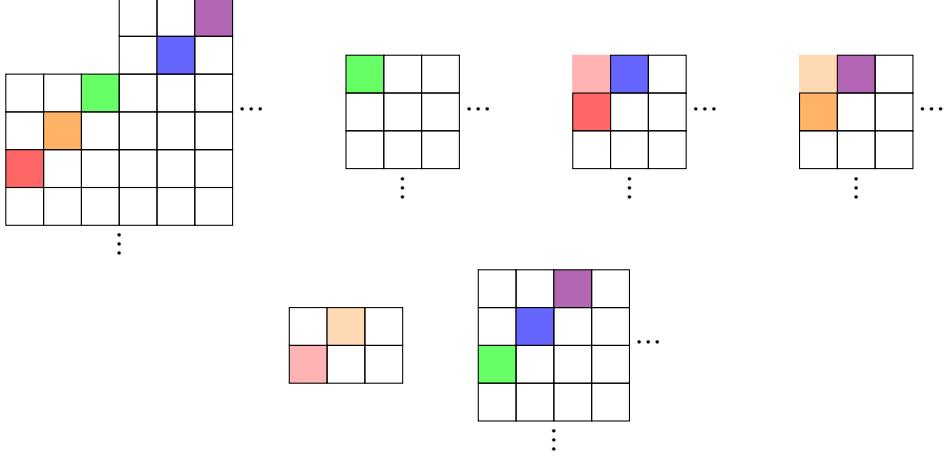


Figure 2.3.5: Pivots of corresponding 3-hooks in the Young diagram asymptotic to  $\lambda = (3, 3)$  and its quotients. In reading order:  $\mathbb{N}^2 \setminus \lambda$ ,  $\mathbb{N}^2 \setminus \lambda_{3,0}$ ,  $\mathbb{N}^2 \setminus \lambda_{3,1}$ ,  $\mathbb{N}^2 \setminus \lambda_{3,2}$ ,  $\lambda$ ,  $\mathbb{N}^2$ .

This bijection is, perhaps unsurprisingly, best explained by example.

**Example 2.3.8.** Let  $\lambda = (3, 3)$ . The Young diagram asymptotic to  $\lambda$  has 5 distinct 3-hooks; their pivots are shaded in Figure 2.3.5.

The three 3-quotients of  $\lambda$  are  $\lambda_{3,0} = \emptyset$ ,  $\lambda_{3,1} = (1)$ , and  $\lambda_{3,2} = (1)$ . Each colored box in  $\mathbb{N}^2 \setminus \lambda$  corresponds to a unique corner in  $\mathbb{N}^2 \setminus \lambda_{3,i}$  for some  $i$ ; for example, the box  $(4, 2)$  is colored orange, and the two edges where its hook intersects the boundary of the diagram are present and adjacent in  $\lambda_{3,2}$  (and therefore no other 3-quotient). It therefore corresponds to the orange box  $(2, 1)$  in  $\mathbb{N}^2 \setminus \lambda_{3,2}$ . Similarly, the box  $(1, 6) \in \mathbb{N}^2 \setminus \lambda$  is colored purple and corresponds to the purple box  $(1, 2) \in \mathbb{N}^2 \setminus \lambda_{3,2}$ .

These two corners  $(2, 1)$  and  $(1, 2)$  in  $\mathbb{N}^2 \setminus \lambda_{3,2}$  necessarily have exactly one corner of  $\lambda_{3,2}$  (i.e.  $(1, 1)$ ) occurring between them when reading the edge sequence from bottom-left to top right. We therefore associate  $(2, 1)$  with  $(1, 1)$  and leave  $(1, 2)$  temporarily unassociated.

Repeating this calculation for the remaining quotients  $\lambda_{3,0}$  and  $\lambda_{3,1}$  associates the red box  $(2, 1) \in \mathbb{N}^2 \setminus \lambda_{3,1}$  with  $(1, 1) \in \lambda_{3,1}$ , and leaves the green, blue, and purple boxes in the quotients unassociated. The purple box corresponds to a corner in the quotient  $\lambda_{3,2}$ ; in the asymptotic Young diagram  $\mathbb{N}^2 = \mathbb{N}^2 \setminus \emptyset$ , the unique box with hook length 3 that corresponds

to a corner in the quotient  $\mathbb{N}^2 \setminus \emptyset_{3,2}$  is  $(1, 3)$ , so we define  $\varphi_\lambda(1, 6) = (1, 3)$ , and similarly for the green and blue boxes. By the same logic, the red and orange corners in  $\lambda_{1,3}$  and  $\lambda_{2,3}$  correspond to the boxes  $(2, 1)$  and  $(1, 2)$  in  $\lambda$ . In total, the map  $\varphi_\lambda$  has the following effect on hook-length-3 boxes in  $\mathbb{N}^2 \setminus \lambda$ :

$$\begin{aligned}\varphi_\lambda(5, 1) &= (2, 1) \in \lambda & \varphi_\lambda(4, 2) &= (1, 2) \in \lambda \\ \varphi_\lambda(3, 3) &= (3, 1) \in \mathbb{N}^2 & \varphi_\lambda(2, 5) &= (2, 2) \in \mathbb{N}^2 & \varphi_\lambda(1, 6) &= (1, 3) \in \mathbb{N}^2.\end{aligned}$$

At last, we are prepared to bijectivize the one-leg PT–DT correspondence Equation (2.5). The proof amounts to a generalization of Theorem 2.2.7 to SPPs that is equivalent to the Pak–Sulzgruber algorithm [Pak01; Sul17], combined with an application of the  $\varphi_\lambda$  map.

**Theorem 2.3.9.** Let  $\lambda \subset \mathbb{N}^2$  be a Young diagram. Then there is a weight-preserving bijection between SPPs  $\sigma$  of shape  $(\emptyset, \emptyset, \lambda)$  and pairs  $(\rho, \pi)$  of RPPs  $\rho$  of shape  $(\emptyset, \emptyset, \lambda)$  and plane partitions  $\pi$ , where  $|\sigma| = |\rho| + |\pi|$ .

*Proof.* Recall that  $V_{(\emptyset, \emptyset, \lambda)}(q)$  can be expressed in terms of vertex operators as

$$V_{(\emptyset, \emptyset, \lambda)}(q) = \left\langle \emptyset \left| \prod_{n \in \mathbb{Z}} \Gamma_{e(n)}(q^{p(n)}) \right| \emptyset \right\rangle,$$

where  $e(n) = e_\lambda(n)$  and  $p(n) = p_\lambda(n)$  are the edge sign and edge power sequences, respectively. Analogous to the proof of Theorem 2.2.7, we follow the algebraic proof of commuting every  $\Gamma_-$  to the right and every  $\Gamma_+$  to the left, while leaving the order of same-sign operators unchanged (i.e. only commuting operators of opposite sign). By Lemma 2.2.6, we may associate a bijection to each commutation we perform, where each one toggles a diagonal in the diagram corresponding to a corner. Moreover, each produces a factor of  $(1 - q^{h(\square)})^{-1}$  by Theorem 2.3.4 and the same logic as in the proof of Theorem 2.2.7. Since any given SPP contains only finitely many nonzero entries, the composition of these bijections is well-defined. Algebraically, we have expressed

$$V_{(\emptyset, \emptyset, \lambda)}(q) = \prod_{\square \in \mathbb{N}^2 \setminus \lambda} \frac{1}{1 - q^{h(\square)}},$$

and combinatorially, there is a bijection between SPPs and hook-length-weighted tableaux of the same shape, given by toggling diagonals until the SPP is empty. We denote it  $\tau$ , since it specializes to the bijection  $\tau$  given in Theorem 2.2.7 when  $\lambda = \emptyset$ .

		3	0	
	4	2	0	...
5	3	2	0	
0	0	0	0	
				⋮

Figure 2.3.6: A skew plane partition  $\sigma$  of shape  $(\emptyset, \emptyset, (2, 1))$ .

We may now complete the proof of the theorem. Given an SPP  $\sigma$  of shape  $(\emptyset, \emptyset, \lambda)$ , we apply the bijection  $\tau$  to create a hook-length-weighted tableau  $\tau(\sigma)$ . We then apply the map  $\varphi_\lambda$  from Definition 2.3.7 that associates the cells of  $\tau(\sigma)$  with the cells of two tableaux  $R$  and  $P$  of shapes  $\lambda$  and  $\mathbb{N}^2$ , respectively, and preserves hook length.

What remains is to convert these two hook-length-weighted tableaux back into an RPP of shape  $(\emptyset, \emptyset, \lambda)$  and a plane partition. For the latter, we apply  $\tau^{-1}$  to produce a plane partition  $\pi = \tau^{-1}(P)$ . For the former, slightly more care must be taken, since the hooks in  $R$  extend down and right rather than up and left. However, we may rotate  $R$  by  $180^\circ$  to express it as an SPP while preserving cells' hook lengths; it is exactly this latter presentation of the bijection that is the Pak–Sulzgruber algorithm [Pak01; Sul17]. We rotate  $R$ , apply  $\tau^{-1}$  to this new SPP, and rotate back, producing an RPP  $\rho$  of shape  $(\emptyset, \emptyset, \lambda)$  that satisfies  $|\sigma| = |\rho| + |\pi|$ .

As in the proof of Theorem 2.3.4, one useful property of this bijectivization is that it cleanly demonstrates the independence of the algorithm from the order in which we toggle diagonals — each  $\Gamma$  commutation produces a commutator that factors directly out of the vertex operator product and does not affect any other operators. This independence is also proven directly in [Pak01].  $\square$

**Example 2.3.10.** Let  $\lambda = (2, 1)$  and let  $\sigma$  be the SPP of shape  $(\emptyset, \emptyset, \lambda)$  in Figure 2.3.6. To convert  $\sigma$  into its corresponding tableau  $\tau(\sigma)$ , we iteratively toggle diagonals beginning with a corner; one consistent way to accomplish this is to choose those corners in lexicographic order. With this choice of order, we produce the sequence of diagrams in Figure 2.3.7. For compactness, we omit the ellipses on the right and bottom of the diagrams, and since we record each entry in the tableau in exactly the location we remove it from the plane partition,

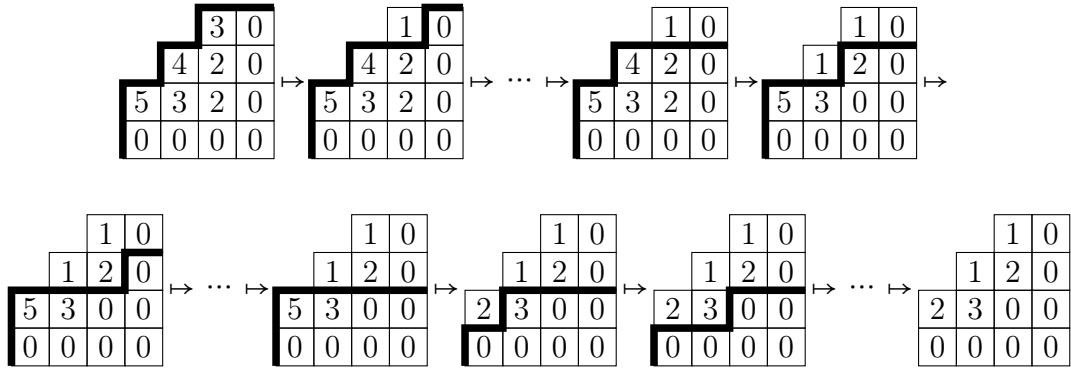


Figure 2.3.7: Decomposing a one-leg SPP into a hook-length-weighted tableau. At each step, we choose a corner of the asymptotic Young diagram (here, we choose them in lexicographic order) and toggle its diagonal.

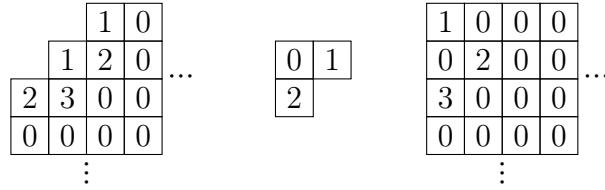


Figure 2.3.8: Rearranging the entries of the tableau  $T(\sigma)$  (left) into a tableau  $T(\rho)$  (center) of shape  $\lambda$  and a tableau  $T(\pi)$  (right) of shape  $\mathbb{N}^2$ .

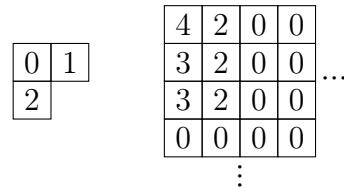


Figure 2.3.9: The untoggled RPP  $\rho = T^{-1}(T(\rho))$  (left) and plane partition  $\pi = T^{-1}(T(\pi))$ .

we superimpose the two objects, following the convention of [Pak01] by separating them with a thicker border.

We now apply the map  $\varphi_\lambda$  to  $\tau(\sigma)$ , producing the tableaux  $R$  and  $P$  in Figure 2.3.8. Finally, we apply  $\tau^{-1}$  to each of these two tableaux (taking care to rotate  $R$  by  $180^\circ$  before untoggling so that the usual plane partition inequalities hold), producing a final RPP  $\rho$  and plane partition  $\pi$  in Figure 2.3.9. Checking the weights, we have  $|\rho| + |\pi| = 3 + 16 = 19 = |\sigma|$ , as expected.

What is new in this section is neither the bijection  $\tau$  nor the corresponding algebraic proof using vertex operators, but the connection between the two of them, along with the use of  $n$ -quotients to bijectivize the one-leg PT–DT correspondence. While the details are more difficult in future sections, the fundamental approach remains similar.

## 2.4. Two-Leg Objects

Both the generating function identity and the bijection of Theorem 2.3.9 hold in a different and in fact more general setting. We begin by generalizing the definitions of skew and reverse plane partition before stating and proving the new result.

**Definition 2.4.1.** Let  $\lambda$  and  $\mu$  be partitions. A **skew plane partition** (SPP) with **shape**  $(\lambda, \mu, \emptyset)$  is a plane partition  $\sigma$  for which  $\sigma(i, j) \geq \max\{\lambda_j, \mu_i\}$  for all  $i, j \in \mathbb{N}$ , and only finitely many  $\sigma(i, j)$  satisfy  $\sigma(i, j) > \max\{\lambda_j, \mu_i\}$ . The generating function  $V_{(\lambda, \mu, \emptyset)}(q)$  is given by

$$V_{(\lambda, \mu, \emptyset)}(q) = \left\langle \lambda \left| \prod_{n=-\infty}^0 \Gamma_- (q^{(-n+1)/2}) \prod_{n=0}^{\infty} \Gamma_+ (q^{(n+1)/2}) \right| \mu \right\rangle, \quad (2.7)$$

where  $q$  marks the weight [ORV06, Equation 3.19]. If  $V_0(\lambda, \mu, \emptyset)$  denotes the minimal exponent of  $q$  in  $V_{(\lambda, \mu, \emptyset)}(q)$ , i.e. the weight of the minimal configuration, then the weight  $|\sigma|$  of a general SPP  $\sigma$  with shape  $(\lambda, \mu, \emptyset)$  is given by

$$|\sigma| = V_0(\lambda, \mu, \emptyset) + \sum_{i,j \in \mathbb{N}} (\sigma(i, j) - \max\{\lambda_j, \mu_i\}).$$

We call these objects **two-leg** SPPs in the terminology of [PT09].

5	4	3	3	3
5	3	2	1	1
3	2	1	0	0
2	2	0	0	0
2	2	0	0	0
⋮				

⋯

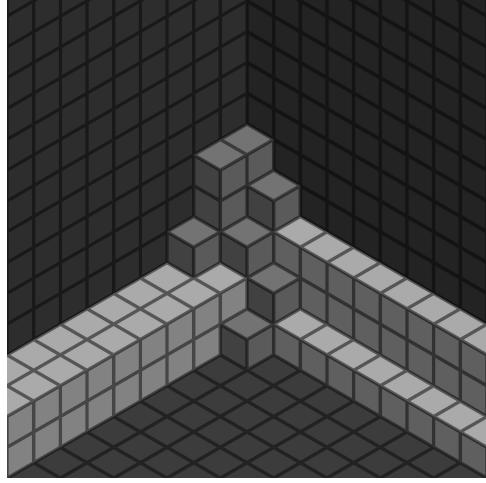


Figure 2.4.1: A two-leg skew plane partition of shape  $((2, 2), (3, 1), \emptyset)$ , visualized both as a grid of numbers and a stack of 10 weight-contributing blocks (dark gray) on top of a non-removable “tray” of blocks that contributes weight 1, as measured by the vertex operator expression Equation (2.7).

An example of an SPP of weight 11 and shape  $(\lambda, \mu, \emptyset)$  for  $\lambda = (2, 2)$  and  $\mu = (3, 1)$  is given in Figure 2.4.1. We remark that our convention differs slightly from [ORV06]: their definition uses the conjugate partition  $\lambda'$  instead of  $\lambda$  itself, in order to match their convention for the fully general three-leg case. In [ORV06, Equation 3.20], the authors also give an expression for three-leg (and thereby two-leg) objects in terms of Schur functions.

The notion of a two-leg RPP is somewhat more complicated. Observe that the definition of a one-leg RPP (Definition 2.3.1) can be directly translated into the language of an SPP without rotating 180° by treating the entries as counting the number of blocks *removed* from the top of a tower that descends downward infinitely (rather than added to an empty floor), as in Figure 2.4.2. This interpretation generalizes more easily — just as the one leg extends infinitely downward, a two-leg RPP will have legs extending horizontally backward from which blocks are removed, effectively placing a cap on the maximum value of each entry.

**Definition 2.4.2.** Let  $\lambda$  and  $\mu$  be partitions. A **reverse plane partition with shape  $(\lambda, \mu, \emptyset)$**  is a function  $\rho : (\mathbb{Z} \times \mathbb{N} \cup \mathbb{N} \times \mathbb{Z}) \rightarrow \mathbb{N}_{\geq 0}$  such that for all  $i$  and  $j$  for which  $\rho(i, j)$  is defined,

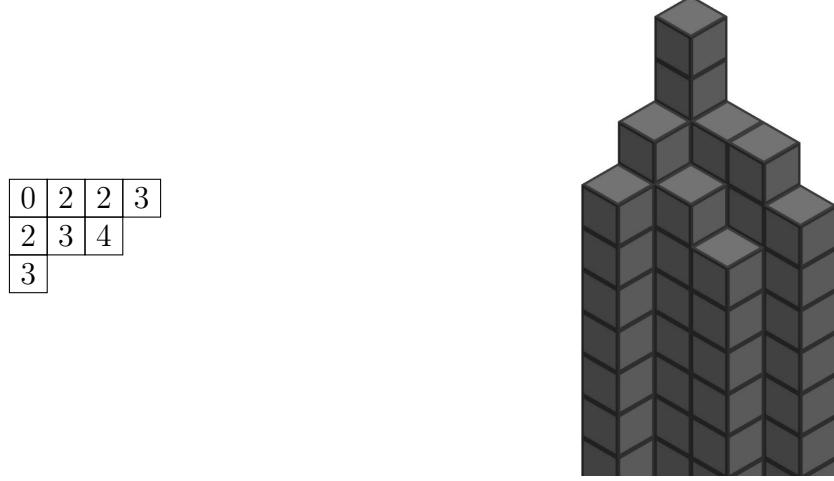


Figure 2.4.2: A one-leg RPP of shape  $(\emptyset, \emptyset, (4, 3, 1))$  and weight 19, visualized as 19 boxes removed from an infinite vertical tower.

1.  $\rho(i, j) \geq \max\{\rho(i + 1, j), \rho(i, j + 1)\}$  (the usual plane partition inequalities).
2.  $\rho(i, j) \leq \min\{\lambda_j, \mu_i\}$ , and the inequality is strict for only finitely many pairs  $(i, j)$ .

Similarly to two-leg SPPs, the generating function  $W_{(\lambda, \mu, \emptyset)}(q)$  for RPPs with shape  $(\lambda, \mu, \emptyset)$  is given by

$$W_{(\lambda, \mu, \emptyset)}(q) = \left\langle \mu \left| \prod_{n=-\infty}^0 \Gamma_+(q^{(-n+1)/2}) \prod_{n=0}^\infty \Gamma_-(q^{(n+1)/2}) \right| \lambda \right\rangle, \quad (2.8)$$

and if  $W_0(\lambda, \mu, \emptyset)$  is the minimal exponent of  $q$  in this generating function, then the weight of a general RPP  $\rho$  with shape  $(\lambda, \mu, \emptyset)$  is given by

$$|\rho| = W_0(\lambda, \mu, \emptyset) + \sum (\min\{\lambda_j, \mu_i\} - \rho(i, j)),$$

where the sum ranges over the entire diagram.

A two-leg RPP has the appearance of a tray pressed up against an infinitely tall building, from which we remove blocks near the corner — Figure 2.4.3 gives an example.

Both two-leg objects are in fact generalizations of their one-leg counterparts: for example,  $V_{(\lambda, \emptyset, \emptyset)}(q) = q^k V_{(\emptyset, \emptyset, \lambda)}(q)$  for some  $k$  by a  $120^\circ$  rotation of the 3-dimensional box diagrams,

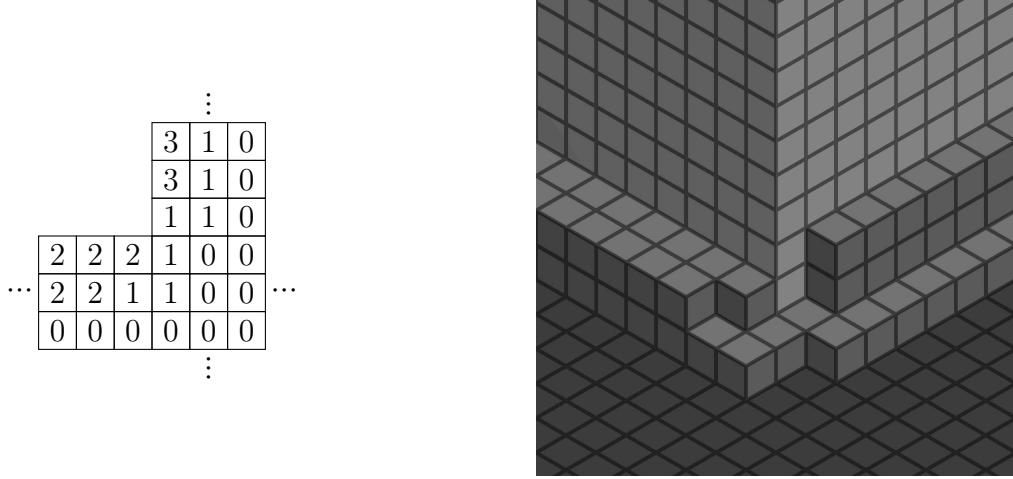


Figure 2.4.3: A two-leg RPP of shape  $((3,1), (2,2), \emptyset)$  and weight 7, visualized as a tray with 6 boxes removed (the minimal configuration has weight 1 as measured by the vertex operators).

and similarly for  $W_{(\lambda, \emptyset, \emptyset)}(q)$ . The factor of  $q^k$  arises since the weights of the minimal configurations may be different. Specifically,  $V_0(\lambda, \mu, \emptyset)$  is not zero in general; as an example,  $V_0((1), \emptyset, \emptyset) = -\frac{1}{2}$ .

The vertex operator expressions for two-leg SPPs and RPPs hint at a relationship between the two:  $W_{(\lambda, \mu, \emptyset)}(q)$  is obtained from  $V_{(\lambda, \mu, \emptyset)}(q)$  by commuting every  $\Gamma_-$  past every  $\Gamma_+$ , then commuting the operators of the same sign so that their order is reversed, and finally swapping the positions of  $\lambda$  and  $\mu$ . In the proofs of Theorem 2.2.7 and Theorem 2.3.9, the first step alone of these three was sufficient, since regardless of the order of the operators relative to others of the same sign, the  $\langle \emptyset |$  and  $| \emptyset \rangle$  bounding the operators ensured that the entire factor counted only the empty partition. As always, commuting a  $\Gamma_+$  with a  $\Gamma_-$  has the effect of toggling the diagonal containing the relevant edges and therefore changing the shape of the objects counted by the generating function. By Proposition 2.2.5 and Lemma 2.2.6, commuting two  $\Gamma$  operators of the same sign is still a toggling operation, but no longer one that changes the shape. Moreover, without the vertical leg, we also no longer need to handle the bookkeeping of  $n$ -quotients and various hook lengths.

Before bijectivizing the two-leg version of the PT–DT correspondence, we require an ad-

ditional technical result. If we limit the size of two-leg SPPs or RPPs to some finite box, effectively limiting  $V_{(\lambda,\mu,\emptyset)}(q)$  and  $W_{(\lambda,\mu,\emptyset)}(q)$  to finitely many  $\Gamma$  operators, then constructing a bijection between the two collections of objects is possible directly from the previous paragraph with little additional work. However, we encounter an issue attempting to define such a bijection for all two-leg SPPs and RPPs: while performing an infinite number of commutations of  $\Gamma$  operators does not necessarily present a problem, performing an infinite number of toggles on a plane-partition-like object *does*. To ensure our map is well-defined, we must be able to guarantee that a limiting behavior exists and can be determined with a finite amount of computation; this is the content of the following proposition.

**Proposition 2.4.3.** Let  $\lambda, \mu \subset \mathbb{N}^2$  be two Young diagrams and let  $\sigma$  be an SPP of shape  $(\lambda, \mu, \emptyset)$ . Given  $n \in \mathbb{N}$ , let  $\sigma_n$  be the result of toggling diagonals of  $\sigma$  until exactly the cells in  $[1, n]^2$  have been popped off. Then:

1. There is an  $N \in \mathbb{N}$  such that all future toggles of  $\sigma_N$  pop off zeros.
2. Fix  $n \geq N$  and let  $\alpha(n, i)$  be the partition given by the diagonal of  $\sigma_n$  beginning at  $(i, n+1)$ . Then for all  $i \in \{1, 2, \dots, n+1\}$ ,  $\alpha(n+1, i) = \alpha(n, i)$ . Similarly, if  $\beta(n, i)$  is the partition given by the diagonal of  $\sigma_n$  beginning at  $(n+1, i)$ , then for all  $i \in \{1, 2, \dots, n+1\}$ ,  $\beta(n+1, i) = \beta(n, i)$ .
3. Let  $\gamma = \alpha(n, n+1) = \beta(n, n+1)$  be the partition on the main diagonal of  $\sigma_n$ ; then  $\gamma = \alpha(n, n) = \beta(n, n)$ .

*Proof.* 1. Toggling every diagonal of  $\sigma$  can produce only finitely many nonzero numbers that are popped off; otherwise,  $\sigma$  would have infinite weight. Therefore, an  $N \in \mathbb{N}$  exists that satisfies the first condition of the proposition. We may also choose  $N$  large enough so that  $\sigma(i, j) = \max\{\lambda_j, \mu_i\}$  (i.e. the minimum possible value) whenever  $(i, j) \in \mathbb{N}^2 \setminus [1, N]^2$ , and also large enough so that  $N \geq \max\{\lambda'_1, \mu'_1\}$ . Now if  $n \geq N$  and  $(i, j) \in \mathbb{N}^2 \setminus [1, n]^2$  satisfies  $\sigma_n(i, j) \neq \sigma(i, j)$ , then  $|j - i|$  must be at most  $n$  — that is,  $(i, j)$  must lie in the diagonals that were toggled to produce  $\sigma_n$ . Therefore, for all

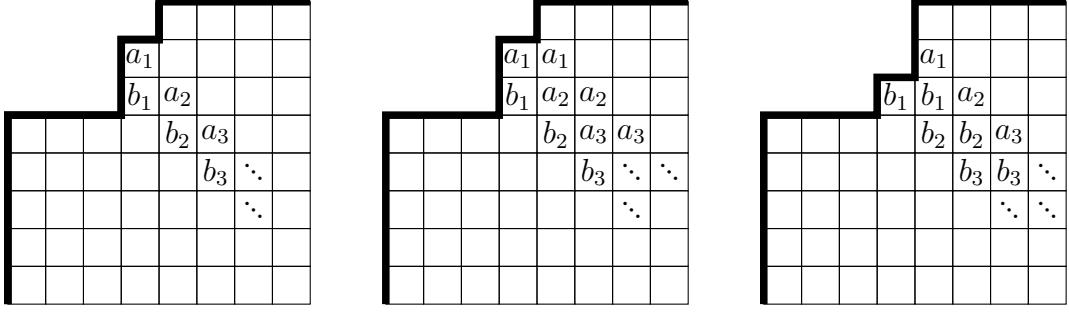


Figure 2.4.4: Three steps of the inductive portion of the proof of Proposition 2.4.3. Left:  $a = \alpha(3, 2)$  and  $b = \alpha(3, 3)$  in a diagram that is  $\sigma_3$  with one toggled corner.

$n \geq N$  and all  $(i, j)$  with  $|j - i| \geq n$ ,  $\sigma_n(i, j) = \max\{\lambda_j, \mu_i\}$ . In particular,  $\sigma_n(i, j) = \mu_i$  whenever  $n \geq N$  and  $j - i \geq N$ .

2. For fixed  $n \geq N$ , we prove the second claim of the proposition by induction on  $i$ . The base case is shown by the previous paragraph:  $\alpha(n, 1) = \alpha(n + 1, 1) = \mu$ . For the induction step, suppose  $i \geq 2$  and  $\alpha(n, i - 1) = \alpha(n + 1, i - 1)$ . For ease of notation, set  $a = \alpha(n, i - 1)$  and  $b = \alpha(n, i)$ ; if we begin with  $\sigma_n$  and toggle the diagonals beginning with cells  $(k, n + 1)$  for all  $k \in \{1, \dots, i - 2\}$ , the result is then the leftmost diagram of Figure 2.4.4. Since the diagonal immediately above  $a$  will not change from any of the remaining toggles that produce  $\sigma_{n+1}$ , it is equal to  $\alpha(n + 1, i - 1)$ ; therefore, the induction hypothesis guarantees that it is in fact equal to  $a$ , as in the middle diagram of Figure 2.4.4. When we toggle the next diagonal down, the plane partition inequalities guarantee that  $a_k \geq b_k$  for all  $k \in \mathbb{N}$ . Therefore, each  $a_k$  toggles to

$$\min\{a_{k-1}, b_{k-1}\} + \max\{a_k, b_k\} - a_k = b_{k-1} + a_k - a_k = b_{k-1},$$

resulting in the rightmost diagram of Figure 2.4.4 and proving the claim. An exactly symmetric argument shows the result for the  $\beta$  partitions.

3. The weight of  $\sigma_n$  is given by the generating function

$$\left\langle \lambda \left| \prod_{k=-\infty}^{-n} \Gamma_-(q^{(-k+1)/2}) \prod_{k=0}^{n-1} \Gamma_+(q^{(k+1)/2}) \prod_{k=-n+1}^0 \Gamma_-(q^{(-k+1)/2}) \prod_{k=n}^{\infty} \Gamma_+(q^{(k+1)/2}) \right| \mu \right\rangle,$$

so the contribution to the weight from the main diagonal  $\gamma$  is given by the middle two  $\Gamma$  operators; that is,

$$\frac{n}{2} (|\alpha(n, n)| - |\gamma|) + \frac{n}{2} (|\beta(n, n)| - |\gamma|).$$

By the previous part of the proposition,

$$\alpha(n+1, n+1) = \alpha(n, n+1) = \gamma$$

$$\beta(n+1, n+1) = \alpha(n, n+1) = \gamma,$$

so if

$$\gamma' = \alpha(n+1, n+2) = \beta(n+1, n+2)$$

is the partition on the main diagonal of  $\sigma_{n+1}$ , then its contribution to the weight of  $\sigma_{n+1}$  is

$$\begin{aligned} & \frac{n+1}{2} (|\alpha(n+1, n+1)| - |\gamma'|) + \frac{n+1}{2} (|\beta(n+1, n+1)| - |\gamma'|) \\ &= \frac{n+1}{2} (|\gamma| - |\gamma'|) + \frac{n+1}{2} (|\gamma| - |\gamma'|) \\ &= (n+1) (|\gamma| - |\gamma'|). \end{aligned}$$

However, the weights of  $\sigma_n$  and  $\sigma_{n+1}$  are equal since no nonzero numbers are popped off in the toggles producing  $\sigma_{n+1}$ . Comparing the two generating functions, the only difference in weight is exactly the contribution from  $\gamma'$ , so  $|\gamma| = |\gamma'|$ . Since  $\gamma > \gamma'$ , the only way for this to occur is if  $\gamma = \gamma'$ .

□

**Theorem 2.4.4.** Let  $\lambda, \mu \in \mathbb{N}^2$  be two Young diagrams. Then there is a weight-preserving bijection between SPPs  $\sigma$  of shape  $(\lambda, \mu, \emptyset)$  and pairs  $(\rho, \pi)$  of RPPs  $\rho$  of shape  $(\lambda, \mu, \emptyset)$  and plane partitions  $\pi$ , where  $|\sigma| = |\rho| + |\pi|$ .

*Proof.* As in the proofs of Theorem 2.2.7 and Theorem 2.3.9, our bijection follows the algebraic proof. Two-leg RPPs and SPPs do not occur frequently in the literature, but the existence of an algebraic proof that  $V_{(\lambda, \mu, \emptyset)}(q) = W_{(\lambda, \mu, \emptyset)}(q)M(q)$  was mentioned in [PT09];

we provide such a proof in full detail here. We begin by commuting the  $\Gamma$  operators of opposite sign in the expression

$$V_{(\lambda, \mu, \emptyset)}(q) = \left\langle \mu \left| \prod_{n=-\infty}^0 \Gamma_- (q^{(-n+1)/2}) \prod_{n=0}^\infty \Gamma_+ (q^{(n+1)/2}) \right| \lambda \right\rangle,$$

producing

$$V_{(\lambda, \mu, \emptyset)}(q) = M(q) \left\langle \mu \left| \prod_{n=0}^\infty \Gamma_+ (q^{(n+1)/2}) \prod_{n=-\infty}^0 \Gamma_- (q^{(-n+1)/2}) \right| \lambda \right\rangle. \quad (2.9)$$

Since the only difference between this vertex operator product and Equation (2.2) is that it contains  $\langle \mu |$  and  $|\lambda \rangle$  instead of  $\langle \emptyset |$  and  $|\emptyset \rangle$ , these commutations produce a factor of  $M(q)$ .

We now “palindromically” commute the operators of the same sign (i.e reverse their order). This produces no factors — it only reweights the objects that are counted to be closer to our usual definitions. The result is

$$V_{(\lambda, \mu, \emptyset)}(q) = M(q) W_{(\mu, \lambda, \emptyset)}(q).$$

However,  $W_{(\mu, \lambda, \emptyset)}(q) = W_{(\lambda, \mu, \emptyset)}(q)$ ; the transpose is a weight-preserving bijection between the sets counted by these two generating functions.

Exactly as in the proof of Theorem 2.2.7, we define a map  $\tau$  on SPPs of shape  $(\lambda, \mu, \emptyset)$  by iteratively toggling diagonals. In that proof, we could justify that the map was well-defined since every plane partition contains only finitely many nonzero entries, but here we must be more careful. Let  $\sigma$  be an SPP of shape  $(\lambda, \mu, \emptyset)$ ; with the  $N \in \mathbb{N}$  guaranteed by Proposition 2.4.3, we may perform toggles until only every cell in  $[1, N]^2$  is popped off, then use just those values to create an  $N \times N$  tableau that we can untoggle into a plane partition  $\pi$  using Theorem 2.2.7.

The remaining toggled object ( $\sigma_N$  in the parlance of Proposition 2.4.3) is constant on diagonals with sufficiently small content, and by that proposition, it continues to have that property as we toggle more and more diagonals. Therefore, we do not need to perform any further toggles to determine the limiting object, an RPP-like remnant infinitely far from the origin. We then perform the toggles corresponding to the same-sign commutations. We first commute  $\Gamma_- (q^{1/2})$  to the left past every other  $\Gamma_-$  operator, then  $\Gamma_- (q^{3/2})$  to the left

6	5	3	3	3	3
5	3	3	1	1	1
3	3	2	0	0	0
2	2	0	0	0	0
2	2	0	0	0	0
2	2	0	0	0	0

⋮

Figure 2.4.5: An SPP of shape  $((2, 2), (3, 1), \emptyset)$  and weight 16.

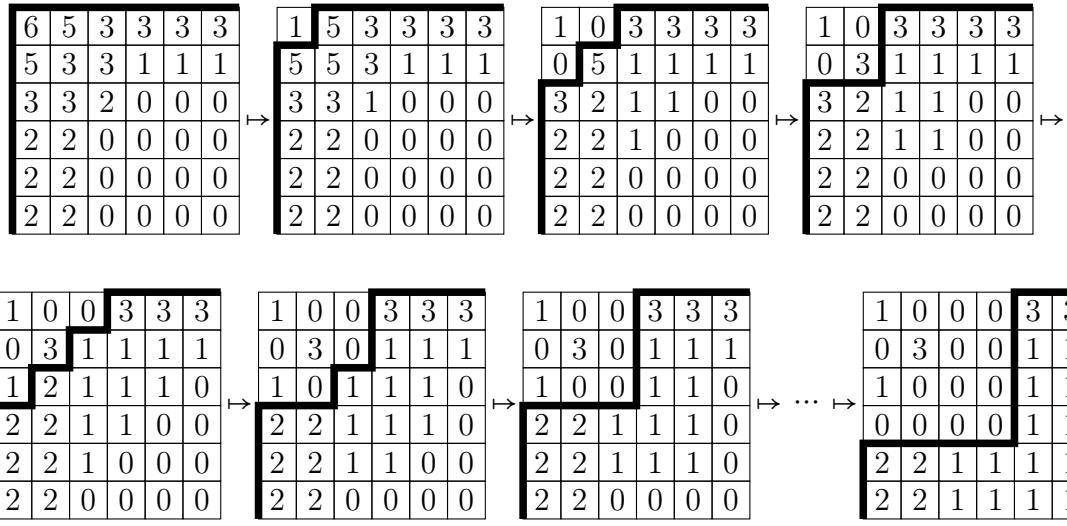


Figure 2.4.6: Iteratively toggling the diagonals of a two-leg SPP.

past every other  $\Gamma_-$  except for  $\Gamma_-(q^{1/2})$ , and so on. Each of these commutations nominally involves an infinite number of toggles, but since the RPP-like object is eventually constant on diagonals, we may determine the output after only a finite number of toggles. Similarly, we need only commute a finite number of the  $\Gamma_-(q^{k/2})$  to the left in total, since the final RPP must also eventually be constant on diagonals far enough away from the origin in order to have finite weight. The result is a two-leg RPP of shape  $(\mu, \lambda, \emptyset)$ ; exactly as in the algebraic proof, we then transpose the diagram to produce an RPP  $\rho$  of shape  $(\lambda, \mu, \emptyset)$ , as required.  $\square$

**Example 2.4.5.** Let  $\lambda = (2, 2)$  and  $\mu = (3, 1)$ , and let  $\sigma$  be the SPP of shape  $(\lambda, \mu, \emptyset)$  and weight 16 given in Figure 2.4.5. To associate  $\sigma$  with a pair  $(\rho, \pi)$  of an RPP  $\rho$  of shape  $(\lambda, \mu, \emptyset)$  and a plane partition  $\pi$ , we begin as in the one-leg case by iteratively toggling the diagonals of  $\pi$  that begin with corners. In Figure 2.4.6, we perform the toggles to produce

1	0	0	0		3
0	3	0	0	...	1 1
1	0	0	0		1 1
0	0	0	0		1 1
				⋮	⋮
2	2	1	1	...	1 1
2	1	1	1		1 1

Figure 2.4.7: The limiting diagram after toggling the SPP in Figure 2.4.6.

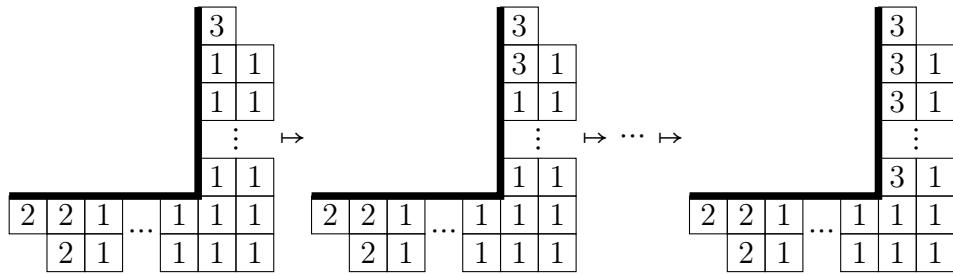


Figure 2.4.8: Commuting the operator  $\Gamma_- (q^{1/2})$  past every other  $\Gamma_-$  has the effect of toggling every diagonal on the right side of the diagram, with the exception of the topmost and bottommost.

3						
3						
3						
1						
⋮						
3						
2	2	2	...	2	1	1
2	2			2	1	1

Figure 2.4.9: The result of palindromically commuting the  $\Gamma$  operators of the same sign.

2	2			
2	2			
2	2			
3	3	3	1	1
1	1	1	1	1

⋮

4	3	0	0
3	1	0	0
1	1	0	0
0	0	0	0

⋮

Figure 2.4.10: The two-leg RPP  $\rho$  (left; after transposing) and the plane partition  $\pi$  (right; after untoggling) corresponding to the SPP  $\sigma$  from Figure 2.4.5.

squares instead of in lexicographic order — the limiting behavior is easier to observe with this approach.

In this example, the value of  $N$  guaranteed by Proposition 2.4.3 is  $N = 3$ , as demonstrated by the final two objects in the sequence in Figure 2.4.6. All future toggles pop off zeros, and diagonals with sufficiently small content are constantly  $(1, 1)$ . The limiting diagram resulting from performing all of the toggles is then given by Figure 2.4.7. We draw the bounding partitions  $\lambda$  and  $\mu$  as diagonals in accordance with the sequence of vertex operators — the diagram begins at a diagonal of  $(2, 2)$  and ends at a diagonal of  $(3, 1)$ . We can easily check at this halfway point that the weights are correct: the tableau in the top-left is weighted by hook length, so it has weight 13. On the other hand, the RPP-like object left over from the toggling is counted by the generating function Equation (2.8), meaning its weight is  $\frac{1}{2} \cdot 2 + \frac{1}{2} + \frac{3}{2} = 3$ , as expected.

To complete the bijection, we palindromically toggle the remaining diagonals of the RPP-like object, as described in the proof of Theorem 2.4.4. Beginning with the right side of the vertex operator expression, we commute the  $\Gamma_-(q^{1/2})$  to the left past every other  $\Gamma_-$ , but no  $\Gamma_+$ . This corresponds to toggling every diagonal on the right of the diagram from top to bottom, except the topmost and bottommost, as shown in Figure 2.4.8. We next commute the  $\Gamma_-(q^{3/2})$  left past every  $\Gamma_-$  except the  $\Gamma_-(q^{1/2})$ , in effect toggling the diagonals from top to bottom but ignoring the second-to-bottom diagonal as well as the bottom one. This has no effect on our example, and in fact the rest of the  $\Gamma_-$  commutations proceed without changing the diagram further. Commuting the  $\Gamma_+$  is an analogous task, and the final diagram is given in Figure 2.4.9.

Comparing this object to the minimal configuration, exactly two boxes have been removed, and the weight of the base RPP of shape  $(\lambda, \mu, \emptyset)$  is 1, so this RPP is still weight 3. All that remains is to transpose the RPP and to convert the tableau produced in the first step back into a plane partition via Theorem 2.2.7, and we have successfully mapped the SPP  $\sigma$  to an RPP  $\rho$  of the same shape and a plane partition  $\pi$  (Figure 2.4.10).

# Chapter 3

## Double-Dimer Objects

### 3.1. Three-Leg Objects

The fully general case of the PT–DT correspondence involves objects that are substantively more complicated than the two special cases we have bijectivized. The notation for and definitions of one- and two-leg SPPs suggest a three-leg object counted by a generating function  $V_{(\lambda,\mu,\nu)}(q)$ , and this is indeed the case. Its definition is exactly what we might expect: we place blocks into a cubical room, all three edges of which are dented in.

**Definition 3.1.1.** Let  $\lambda$ ,  $\mu$ , and  $\nu$  be partitions. A **skew plane partition with shape**  $(\lambda, \mu, \nu)$  is an SPP of shape  $(\emptyset, \emptyset, \nu)$  for which  $\alpha(i, j) \geq \max\{\lambda_i, \mu_j\}$  for all  $(i, j) \in \mathbb{N}^2 \setminus \nu$ , and only finitely many  $\alpha(i, j)$  satisfy  $\alpha(i, j) > \max\{\lambda_i, \mu_j\}$ . The generating function  $V_{(\lambda,\mu,\nu)}(q)$  is given by

$$V_{(\lambda,\mu,\nu)}(q) = \left\langle \mu \left| \prod_{n \in \mathbb{Z}} \Gamma_{e(n)}(q^{p(n)}) \right| \lambda \right\rangle \quad (3.1)$$

for  $e(n) = e_\nu(n)$  and  $p(n) = p_\nu(n)$ , where  $q$  marks the weight. We once again denote the minimal exponent of  $q$  in  $V_{(\lambda,\mu,\nu)}(q)$  by  $V_0(\lambda, \mu, \nu)$ , and the weight  $|\alpha|$  of an SPP  $\alpha$  with shape  $(\lambda, \mu, \nu)$  is given by

$$|\alpha| = V_0(\lambda, \mu, \nu) + \sum_{(i,j) \in \mathbb{N}^2 \setminus \nu} (\alpha(i, j) - \max\{\lambda_i, \mu_j\}).$$

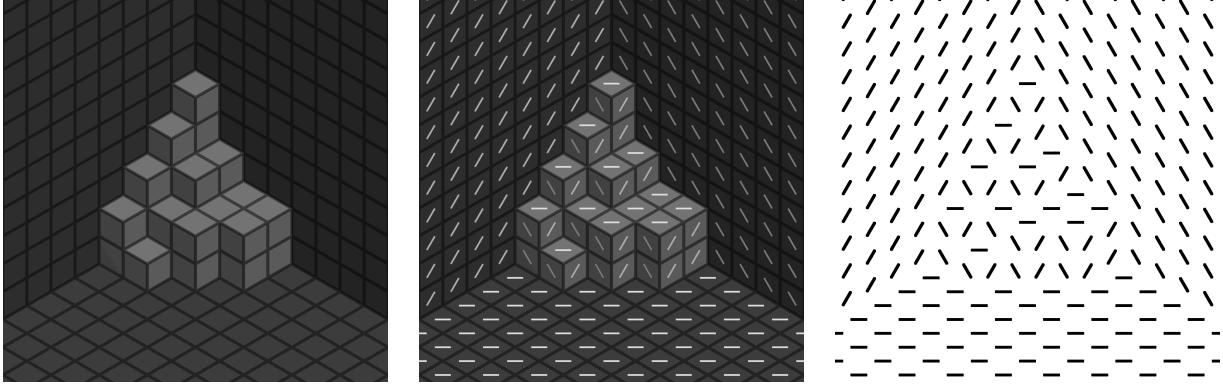


Figure 3.1.1: A plane partition with walls shown (left), each rhombus painted with a dimer (center), and the configuration formed from those dimers (right).

We may combine the bijections from Theorem 2.3.9 and Theorem 2.4.4 to produce a partial result: as in the one-leg case, starting with an SPP of shape  $(\lambda, \mu, \nu)$  and iteratively toggling diagonals produces exactly the data of a plane partition and an RPP of shape  $(\emptyset, \emptyset, \nu)$ , and leaves behind an object similar to a two-leg RPP, but with unusual weighting: the order of the operators in Equation (3.1) has been reversed. Unfortunately, the corresponding generalization of RPPs is not nearly so simple as three-leg SPPs, nor does it look or act similarly to the product of this bijection. As an indication of why there is nuance to this definition, notice that in a two-leg RPP, as in Figure 2.4.3, the middle area in the minimal configuration is the intersection of the two horizontal legs. If a three-leg RPP were simply three legs and their intersection, then setting the vertical leg to the empty partition should result in two-leg RPPs having empty intersections in their centers rather than the two-fold intersections that they do have.

To understand the definition of a three-leg RPP, we first need to develop an alternate perspective for all of our current objects of study. Plane partitions are famously in correspondence with perfect matchings (or **dimer configurations**) on a hexagon lattice in  $\mathbb{R}^2$  by drawing a line in the middle of each rhombus face when the 3-dimensional block expression of a plane partition is viewed from an isometric perspective, as in Figure 3.1.1.

Since each vertex is matched with exactly one other, we call these **single-dimer** objects. In contrast, the three-leg generalization of RPPs are **double-dimer** objects, meaning every

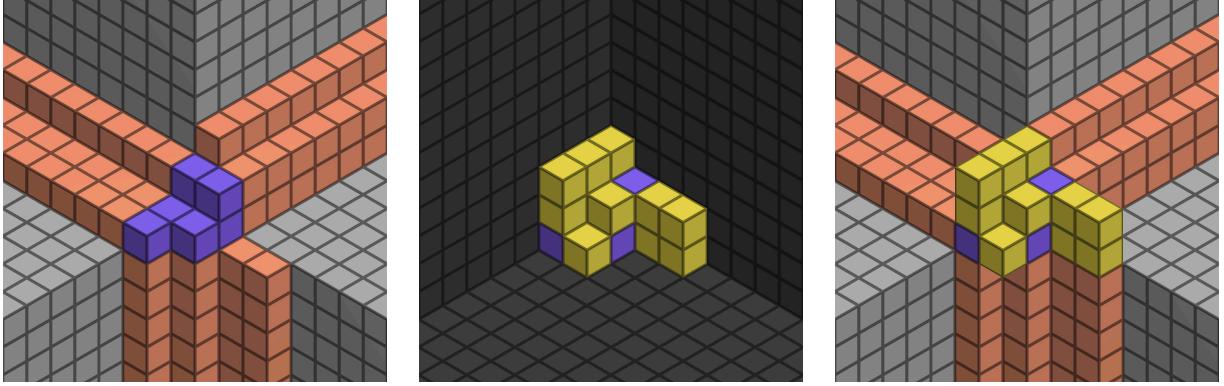


Figure 3.1.2: Regions I (orange), II (yellow), and III (purple) for  $\lambda = (3, 2)$ ,  $\mu = (2, 1, 1)$ , and  $\nu = (4, 2, 1)$ . Taking only regions I and III (left) or II and III (center) produce plane-partition-like objects, while all three together (right) do not.

vertex is matched to exactly two others. We refer the interested reader to [Ken03] for further details on dimer configurations. While a definition in terms of double-dimer objects is certainly more natural, a description in terms of boxes is much more conducive to our existing results. Such a description exists, but it is nuanced and technical; we first require more notation.

**Definition 3.1.2.** Fix partitions  $\lambda$ ,  $\mu$ , and  $\nu$  and define  $\text{Cyl}_1, \text{Cyl}_2, \text{Cyl}_3 \subset \mathbb{Z}^3$  by

$$\begin{aligned}\text{Cyl}_1 &= \{(x, y, z) \in \mathbb{Z}^3 \mid y \geq 1 \text{ and } 1 \leq z \leq \lambda_y\} \\ \text{Cyl}_2 &= \{(x, y, z) \in \mathbb{Z}^3 \mid x \geq 1 \text{ and } 1 \leq z \leq \mu_x\} \\ \text{Cyl}_3 &= \{(x, y, z) \in \mathbb{Z}^3 \mid y \geq 1 \text{ and } 1 \leq x \leq \nu_y\}.\end{aligned}$$

These cylindrical regions are simply Cartesian products of the Young diagrams for  $\lambda$ ,  $\mu$ , and  $\nu$  with  $\mathbb{Z}$ . We now define the following intersections of these regions: first,  $I_i = \text{Cyl}_i \cap (\mathbb{Z}_{\leq 0})^3$  and  $I = I_1 \cup I_2 \cup I_3$  are simply the negative components of each cylinder. Next,  $II_{\bar{1}} = \text{Cyl}_2 \cap \text{Cyl}_3$ ,  $II_{\bar{2}} = \text{Cyl}_1 \cap \text{Cyl}_3$ ,  $II_{\bar{3}} = \text{Cyl}_1 \cap \text{Cyl}_2$ , and  $II = II_{\bar{1}} \cup II_{\bar{2}} \cup II_{\bar{3}}$  are the twofold intersections. Finally,  $III = \text{Cyl}_1 \cap \text{Cyl}_2 \cap \text{Cyl}_3$  is the threefold intersection.

For  $\lambda = (3, 2)$ ,  $\mu = (2, 1, 1)$ , and  $\nu = (4, 2, 1)$ , regions I, II, and III are shown in Figure 3.1.2.

In an RPP of shape  $(\emptyset, \emptyset, \nu)$ , only region I is present, and in one of shape  $(\lambda, \mu, \emptyset)$ , only I and II are present. However, introducing region III changes things drastically. Just as the two-way intersection boxes in region II are counted with multiplicity one in two-leg RPPs, three-way intersection boxes in III will be counted with multiplicity *two*.

**Definition 3.1.3.** Fix partitions  $\lambda$ ,  $\mu$ , and  $\nu$  and the corresponding regions I, II, and III. A **reverse plane partition of shape**  $(\lambda, \mu, \nu)$  is a pair  $\rho = (A, B)$  of collections of boxes  $A$  and  $B$ , with  $A \subseteq I \cup III$  and  $B \subseteq II \cup III$ , such that

1. All but finitely many boxes in  $I \cup III$  are in  $A$ .
2. If  $(i, j, k) \in A$  and  $(i-1, j, k) \in I \cup III$ , then  $(i-1, j, k) \in A$ , and similarly for  $(i, j-1, k)$ ,  $(i, j, k-1)$  (i.e.  $A$  satisfies the plane partition inequalities). Similarly, if  $(i, j, k) \in B$  and  $(i-1, j, k) \in II \cup III$ , then  $(i-1, j, k) \in B$ , and identically for  $(i, j-1, k)$  and  $(i, j, k-1)$ .
3. Each connected component (via adjacent boxes) of

$$(I \setminus A) \cup (II \cap B) \cup (III \cap (A \Delta B))$$

intersects only one of  $I_1 \cup II_{\bar{1}}$ ,  $I_2 \cup II_{\bar{2}}$ , and  $I_3 \cup II_{\bar{3}}$  (here,  $A \Delta B = (A \cup B) \setminus (A \cap B)$  is the symmetric difference of  $A$  and  $B$ ).

We define the weight of  $\rho$  to be

$$|\rho| = V_0(\lambda, \mu, \nu) + |(I \cup III) \setminus A| + |(II \cup III) \setminus B|,$$

i.e. the total number of boxes removed, counted with multiplicity. Finally, we define

$$W_{(\lambda, \mu, \nu)}(q) = \sum_{\rho} q^{|\rho|}$$

to be the generating function for three-leg RPPs.

These three-leg RPPs are so-called PT objects, introduced in [PT09]. However, it is considerably more convenient from a combinatorial perspective to think of the third condition in terms of labels: boxes in  $I_1 \cup II_{\bar{1}}$  that are removed from  $A$  or left in  $B$  are labeled 1,

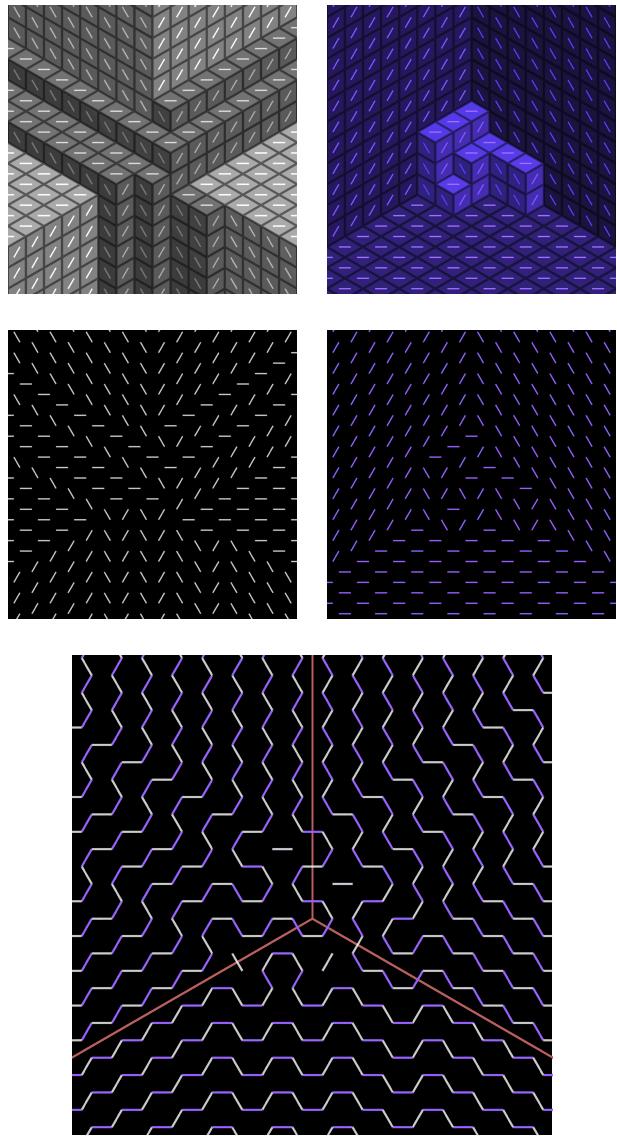


Figure 3.1.3: The minimal AB configuration with visible dimers for  $\lambda = (3, 2)$ ,  $\mu = (2, 1, 1)$ , and  $\nu = (4, 2, 1)$ , as in Figure 3.1.2 (top row). Those dimers alone (middle row). Overlaying those configurations to produce a double-dimer configuration (bottom). Notice that sufficiently far from the origin, the paths follow the shape of the Young diagrams for  $\lambda$ ,  $\mu$ , and  $\nu$ , and although they may cross the  $120^\circ$  sectors indicated by the red lines when close to the origin, they eventually each remain in only one sector.

and similarly for labels 2 and 3. Boxes present in III with multiplicity one are chameleons, assuming any label present in their component or remaining unlabeled if none exists. An RPP  $\rho = (A, B)$  is then valid if and only if there is no conflict in labels, i.e. two differing labels in a single connected component. These so-called **labeled  $AB$  configurations** were first introduced in [JWY22], although they were primarily studied in terms of their dimer configurations, and so their definition is slightly more complicated in terms of labeling. In contrast to the seemingly arcane labeling rules of the  $AB$  configurations, the corresponding double-dimer objects produced by superimposing the individual single-dimer configurations for  $A$  and  $B$  have a simple construction: they are exactly the double-dimer configurations such that sufficiently far from the origin, the unbounded paths have shapes defined by the three legs, and each of those paths remains in only one of the three  $120^\circ$  sectors with degree measures in the intervals  $(-30^\circ, 90^\circ)$ ,  $(90^\circ, 210^\circ)$ , and  $(210^\circ, 330^\circ)$ , as in Figure 3.1.3 ([JWY22, Section 4]).

The PT–DT correspondence in its full generality —  $V_{(\lambda, \mu, \nu)}(q) = W_{(\lambda, \mu, \nu)}(q)M(q)$  — holds for these three-leg objects, and it was proven algebraically in [JWY22]. However, the challenges to developing an analogue of our combinatorial proof are immediate and substantial. Three-leg RPPs do not naturally admit a vertex operator expression — the labeling rules mean that the operators to translate between diagonal slices no longer depend only on local conditions, and the doubled boxes make it unclear what objects those vertex operators would even be acting upon. Nevertheless, we can begin to understand the structure of three-leg RPPs by examining the posets of their individual entries; these posets are the focus of the following section.

## 3.2. The Entry Posets of Three-Leg Objects

The toggle operation described in Section 2.2 is no more than an involution on a large rectangular graded poset: if an entry of a diagonal is bounded above by  $M$  and below by  $m$ , then its toggle map  $x \mapsto M + m - x$  is the unique involution on the graded poset  $[m, M] \cap \mathbb{N}_{\geq 0}$  of possible values. The poset of an entire diagonal is then just a product of these intervals,

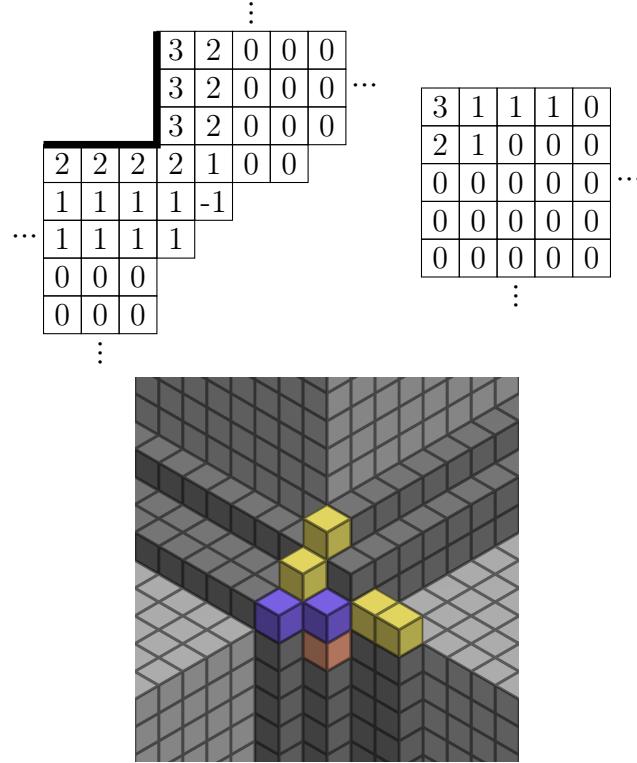


Figure 3.2.1: Valid values for  $A$  (left) and  $B$  (center) to create a three-leg RPP  $\rho$  of shape  $(\lambda, \mu, \nu)$ , where  $\lambda = (3, 2)$ ,  $\mu = (2, 1, 1)$ , and  $\nu = (4, 2, 1)$ . The five connected components created from the boxes removed from I, left in II, and removed with multiplicity one from III, colored by region (right).

possibly with one unbounded interval; toggling it performs the involution on each factor and extracts the amount the unbounded component is above the minimum if it is present. This framing is substantially easier to generalize to objects like three-leg RPPs, since we no longer need to find the exact map from scratch — understanding the entries' posets is enough.

**Example 3.2.1.** Let  $\lambda = (3, 2)$ ,  $\mu = (2, 1, 1)$ , and  $\nu = (4, 2, 1)$ , and let  $\rho = (A, B)$  be a three-leg SPP, where  $A$  and  $B$  are given in Figure 3.2.1.

We first verify that this is a valid three-leg RPP. There is only one box removed from I:  $(2, 2, 0)$ . The boxes removed from II are  $(1, 3, 2)$ ,  $(1, 4, 2)$ ,  $(1, 4, 1)$ ,  $(2, 2, 2)$ ,  $(3, 1, 3)$ , and  $(3, 2, 1)$ ; and the boxes removed from III are  $(1, 2, 2)$  (from both  $A$  and  $B$ ),  $(2, 2, 1)$  (from only  $A$ ), and  $(3, 1, 1)$  (from only  $B$ ). For the relevant connected components, we need to consider the boxes *not* removed from II, which are  $(1, 1, 3)$ ,  $(1, 3, 1)$ ,  $(1, 4, 1)$ , and  $(2, 1, 2)$ . These

$b$		
(3, 1)      (2, 2)      (1, 4)		
2 1 0  2 0  -1 -2 -3  ⋮	1 0  1 0  -1 -2 -3  ⋮	1 0  1 0  -1 -2 -3  ⋮
2 1 0  2 0  -1 -2 -3  ⋮	1 0  1 0  -1 -2 -3  ⋮	1 0  1 0  -1 -2 -3  ⋮

Figure 3.2.2: The poset diagram for the entries  $(3, 1)$  (left),  $(2, 2)$  (center), and  $(1, 4)$  (right) in the three-leg RPP given in Figure 3.2.1, where all other entries are held fixed. Invalid labeling configurations are represented by a missing table entry, and valid configurations are represented by either the label of their connected component, a star if the only boxes in the component are unlabeled, or an empty set if there are no boxes in the component at all.

four boxes, along with the one removed from I and the two removed with multiplicity one from III, are superimposed in Figure 3.2.1, using the same colors for regions as Figure 3.1.2. While this figure makes the resulting five connected components clear, it must be interpreted carefully. The dark gray boxes are those in  $A$ , while the central purple box and the orange box below it are not: they are the boxes in regions III and I, respectively, that were removed from  $A$ . Similarly, only the boxes left in  $B$  are drawn, superimposed on the diagram.

Recall that only region I and II boxes contribute labels, whereas region III boxes take on the labels of their connected components. The labels are therefore:

- $\{(1, 1, 3)\}$ : 2,
- $\{(1, 3, 1), (1, 4, 1)\}$ : 1,
- $\{(2, 1, 2)\}$ : 2,
- $\{(2, 2, 0), (2, 2, 1)\}$ : 3,
- $\{(3, 1, 1)\}$ : unlabeled.

Each component has at most one distinct label, and so the configuration is valid.

Let us now examine several entries' posets; specifically, corners of  $\nu$ . Holding all other entries fixed, what are the valid values of  $A$  and  $B$  at  $(3, 1)$ ? Denote them  $a$  and  $b$ ; in theory, the maximum value of  $a$  is 1, and the minimum value is unbounded, whereas the possible values for  $b$  are only 2, 1, and 0. In practice, however, the labeling condition means not all pairs of these are guaranteed to be valid. The core issue is that whenever  $a \leq -1$ , there are boxes removed from I that contribute the label of 3, while whenever  $b \geq 2$ , there are boxes in region II intersecting with  $B$  that contribute a label other than 3 — in this case, 2. Both cannot be true at once, and so the entry poset is as given in Figure 3.2.2.

The entry posets are graded by diagonal, with the top-left entry having minimal weight, and each of the three in Figure 3.2.2 decomposes into two distinct rectangles: the top one, in which the boxes have labels other than 3, and a weakly narrower one below it that is unbounded below, in which the boxes exclusively have label 3. to a product of intervals, one of which is unbounded. In fact, every boundary corner in any three-leg RPP has an entry poset of this form; this is the content of Theorem 3.2.5, but we first require a small amount of notation and a number of intermediate results.

**Definition 3.2.2.** Let  $\lambda$ ,  $\mu$ , and  $\nu$  be partitions, and let  $\rho = (A, B)$  be a three-leg RPP with shape  $(\lambda, \mu, \nu)$ . Fix a corner  $(i, j)$  of  $\nu$ . We write  $A_a$  and  $B_b$  to mean  $A$  and  $B$ , but with  $A(i, j) = a$  and  $B(i, j) = b$ , where  $a, b \in \mathbb{Z}$ . Note that  $(A_a, B_b)$  need not be a valid three-leg RPP.

For fixed  $\rho = (A, B)$ , we also define the tower of labelable boxes  $T_{i,j}$  on  $(i, j)$  for any  $(i, j) \in \mathbb{Z}^2$  by

$$T_{i,j} = ((i, j) \times \mathbb{Z}) \cap ((I \setminus A) \cup (II \cap B) \cup (III \cap (A \Delta B))),$$

i.e. the boxes that contribute to the connected components in  $\rho$ .

**Proposition 3.2.3.** Let  $\lambda$ ,  $\mu$ , and  $\nu$  be partitions, and let  $\rho = (A, B)$  be an RPP with shape  $(\lambda, \mu, \nu)$ . Let  $(i, j)$  be a corner of  $\nu$ , and let  $\nu' = \nu \setminus \{(i, j)\}$  and  $A' = A \setminus ((i, j) \times \mathbb{Z})$ . If  $P' \subset \mathbb{N}_{\geq 0}$  is the poset of values  $b$  for which  $(A', B_b)$  is a valid RPP of shape  $(\lambda, \mu, \nu')$ , then  $P'$  is an interval (i.e. of the form  $[a, b] \cap \mathbb{N}_{\geq 0}$ ).

*Proof.* Let  $T_{i,j}$  be the tower of labelable boxes on  $(i,j)$  as previously defined. Since  $(i,j) \in \nu$ , all boxes  $(i,j,k) \in T_{i,j}$  that are in region I or region II have label 3. Similarly, if  $T'_{i,j}(b)$  is the tower of labelable boxes on  $(i,j)$  in the RPP  $(A',B_b)$ , then it consists exclusively of label-3 boxes in region II: regions I and III both no longer intersect the cylinder  $(i,j) \times \mathbb{Z}$  and so do not intersect  $T'_{i,j}(b)$ . Lowering the value of  $b$  then always results in fewer labeled boxes, which can never introduce a labeling conflict. Therefore, if  $b_0 \in P'$ , then so is every  $b < b_0$  that satisfies the plane partition inequalities, and so  $P'$  is an interval.  $\square$

Our next result is slightly more technical: it concerns the labeling status of region III boxes in particular cases.

**Proposition 3.2.4.** Let  $\lambda, \mu$ , and  $\nu$  be partitions, and let  $\rho = (A,B)$  be an RPP with shape  $(\lambda, \mu, \nu)$ . Suppose  $C$  is a connected component of

$$(I \setminus A) \cup (II \cap B) \cup (III \cap (A \triangle B)),$$

i.e. the labelable boxes of  $\rho$ . If  $C$  contains a multiplicity-one region III box in  $A$ , then it contains *only* such boxes.

*Proof.* Suppose  $(i,j,k) \in A$  with multiplicity one. Then  $(i,j,k) \notin B$ , and so  $A(i,j) \geq k$  and  $B(i,j) < k$ . If there were an adjacent multiplicity-one region III box in  $B$ , then it would have to be in one of the directions in which  $A$  is not guaranteed to have a box; specifically, one of  $(i+1,j,k)$ ,  $(i,j+1,k)$ , or  $(i,j,k+1)$ . However, those are exactly the directions in which  $B$  is guaranteed also not to have a box, and so multiplicity-one region III boxes in  $A$  and  $B$  cannot border one another.

If  $(i,j,k) \in A$  is in region III, then  $(i-1,j,k), (i,j-1,k), (i,j,k-1) \in A$ , meaning multiplicity-one region III boxes in  $A$  cannot border region I boxes not in  $A$ . Similarly, for  $(i,j,k) \in A$  to be a multiplicity-one region III box,  $(i,j,k) \notin B$ , so  $(i+1,j,k), (i,j+1,k), (i,j,k+1) \notin B$ , guaranteeing that  $(i,j,k)$  does not border region II boxes present in  $B$  either. In total, multiplicity-one region III boxes in  $A$  are not connected to either labeled boxes or to multiplicity-one region III boxes in  $B$ .  $\square$

With these results, we are prepared to state and prove the classification of corner entry posets.

**Theorem 3.2.5.** Let  $\lambda$ ,  $\mu$ , and  $\nu$  be partitions, and let  $\rho = (A, B)$  be a three-leg RPP with shape  $(\lambda, \mu, \nu)$ . Let  $(i, j) \in \mathbb{N}^2$  be a corner of  $\nu$ , and let  $P \subset \mathbb{Z} \times \mathbb{N}_{\geq 0}$  be the graded poset of pairs  $(a, b)$  for which  $(A_a, B_b)$  is a valid RPP of shape  $(\lambda, \mu, \nu)$ . Moreover, let  $\nu'$  be equal to  $\nu$ , but with  $(i, j)$  removed, and let  $A'$  be equal to  $A$ , but with all boxes  $(i, j) \times \mathbb{Z}$  removed. Finally, let  $P' \subset \mathbb{N}_{\geq 0}$  be the poset of values  $b$  for which  $(A', B_b)$  is a valid three-leg RPP of shape  $(\lambda, \mu, \nu')$ . Then  $P'$  is an interval, and  $P$  is equal to

$$([b_{\min}, a_{\max}] \times [b_{\min}, b_{\max}] \cup (-\infty, b_{\min}) \times [b_{\min}, b_{\text{mid}}]) \cap \mathbb{Z}^2,$$

where  $P' = [b_{\min}, b_{\text{mid}}] \cap \mathbb{N}_{\geq 0}$ ,  $a_{\max} = \min \{A(i-1, j), A(i, j-1)\}$  is the maximum possible value of  $a$ , and  $b_{\max}$  is the largest value of  $b$  so that  $(A_{a_{\max}}, B_b)$  is a valid RPP of shape  $(\lambda, \mu, \nu)$ .

*Proof.* By Proposition 3.2.3,  $P'$  is indeed an interval. Let  $b \in P'$  and let  $(A', B_b)$  be one of the valid RPPs of shape  $(\lambda, \mu, \nu')$ ; then all boxes in  $T_{i,j}$  are in region II and have label 3; what enforces the maximum on  $b$  is possibly the plane partition inequalities, but also possibly a labeling conflict. When we consider RPPs  $(A_a, B_b)$  of shape  $(\lambda, \mu, \nu)$ , however, the situation is strikingly similar: we claim that when  $a < b_{\min}$ , the boxes in  $T_{i,j}$  have label 3. If  $b_{\min} = 0$ , then  $a < 0$  implies at least one label-3 region I box has been removed from  $A$ . On the other hand, if  $b_{\min} > 0$ , then one of  $B(i+1, j)$  and  $B(i, j+1)$  must be nonzero. Without loss of generality, suppose it is the former; since  $(i, j)$  is a corner,  $(i+1, j)$  is outside  $\nu$ , meaning  $(i+1, j, b_{\min}-1)$  is a labelable box in region II that is present in  $B$ . Since  $a < b_{\min}$ ,  $(i, j, b_{\min}-1)$  is a multiplicity-one region III box that then inherits the label of 3 and propagates it to the rest of  $T_{i,j}$ , proving the claim.

For fixed  $a < b_{\min}$ , there is a one-to-one correspondence between RPPs  $(A_a, B_b)$  and RPPs  $(A', B_b)$ : in the former, the boxes in  $B$  are label-3 multiplicity-one region III boxes, and in the latter, they are label-3 region II boxes, but the labeling conditions for both are identical. Aside from those labeling conditions,  $A$  and  $B$  are completely independent, and so we have completely determined the structure of  $P$  when  $a < b_{\min}$ : specifically,

$$P \cap ((-\infty, b_{\min}) \times \mathbb{N}_{\geq 0}) = ((-\infty, b_{\min}) \cap \mathbb{Z}) \times P'.$$

Next, we examine the portion of  $P$  when  $a \geq b_{\min}$ . By Proposition 3.2.4, the boxes in  $T_{i,j}$  are

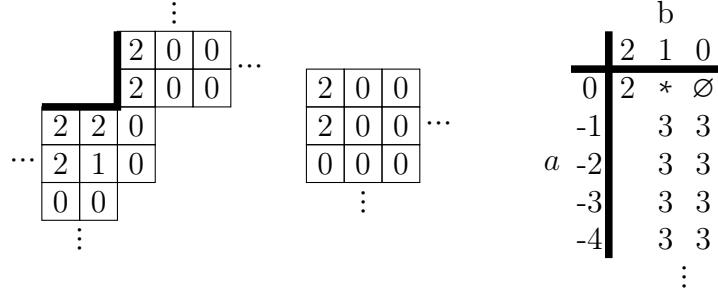


Figure 3.2.3: Values for  $A$  (left) and  $B$  (center) to create a three-leg RPP  $\rho$  of shape  $(\lambda, \mu, \nu)$ , where  $\lambda = (2)$ ,  $\mu = (2, 2)$ , and  $\nu = (1, 1)$ . The poset diagram for  $(2, 1)$  that fails to have a grading-preserving bijection to a product of intervals (right).

guaranteed to be unlabeled, and therefore not cause a labeling conflict, so long as there is at least one multiplicity-one region III box in  $A$ ; in other words, when  $a > b$ . When  $a = b$ ,  $T_{i,j}$  is empty and therefore does not cause a labeling conflict either. This shows that  $P$  contains all pairs of the form  $(a, b)$  for  $b_{\min} \leq b \leq a \leq a_{\max}$ , but we may also interchange  $a$  and  $b$  so long as the plane partition inequalities are satisfied — the set  $T_{i,j}$  remains the same, and so it is still guaranteed to be unlabeled, even when it only contains multiplicity-one region III boxes in  $B$ . Therefore,  $P$  contains the entire rectangle

$$([b_{\min}, a_{\max}] \times [b_{\min}, \min \{a_{\max}, B(i-1, j), B(i, j-1)\}]) \cap \mathbb{Z}^2.$$

While  $a$  cannot be any larger than  $a_{\max}$ ,  $b$  possibly could be — building with unlabeled boxes causes fewer labeling conflicts than building with labeled ones. However, whether the multiplicity-one region III boxes in  $B$  when  $b \geq a$  inherit the label of 3 is the only effect that the value of  $a$  can have on the maximum value of  $b$ . Therefore,  $b_{\max}$  is always the maximum value of  $b$  when  $a \geq b_{\min}$ , and so we have shown that  $P$  contains the rectangle

$$([b_{\min}, a_{\max}] \times [b_{\min}, b_{\max}]) \cap \mathbb{Z}^2,$$

proving the theorem.  $\square$

In each of the posets in Figure 3.2.2, the height of the upper rectangle is equal to the width of the lower one, meaning each has a grading-preserving bijection to a product of intervals.

This would make a potential toggling map simple to define by choosing the unique involution on the finite factor and extracting the infinite component — in other words, balancing the  $A$  and  $B$  components and treating them separately. Unfortunately, not every entry poset is of this form. With  $\lambda = (2)$ ,  $\mu = (2, 2)$ ,  $\nu = (1, 1)$ , and  $A$  and  $B$  given in Figure 3.2.3, the poset diagram of the corner  $(2, 1)$  of  $\nu$  is as shown in that figure. Replacing every entry of 2 in  $A$  and  $B$  with  $n$ , and the entry of 1 in  $A$  with  $m$  for  $0 \leq m < n$ , we can produce a poset diagram with bottom width  $m + 1$  and top width  $n$ , meaning the poset diagrams can be arbitrarily poorly-behaved.

### 3.3. Future Directions

The results of Section 3.1 indicate that defining a local, toggle-like move on three-leg RPPs is likely complicated. Nevertheless, the classification of poset diagrams is the first local description of the labeling condition for three-leg RPPs of which the authors are aware, and such a description points in the direction of a local map on diagonals that preserves weight while simplifying the structure of an RPP.

**Conjecture 3.3.1.** Let  $\lambda$ ,  $\mu$ , and  $\nu$  be partitions, and let  $\rho = (A, B)$  be a three-leg RPP of shape  $(\lambda, \mu, \nu)$ . If  $(i, j)$  is **any** entry, then the poset  $P$  of pairs  $(a, b)$  such that  $(A_a, B_b)$  is a valid RPP decomposes into

$$P = ([b_{\min}, a_{\max}] \times [b_{\min}, b_{\max}] \cup [a_{\min}, b_{\min}] \times [b_{\min}, b_{\text{mid}}]) \cap \mathbb{Z}^2,$$

where  $a_{\min}$  is the minimum possible value of  $a$ , possibly  $-\infty$ ,  $a_{\max}$  is the maximum,  $b_{\max}$  is the absolute maximum value of  $b$ , and  $b_{\text{mid}}$  is the maximum possible value when  $a < b_{\min}$ . Moreover, each entry poset depends only on the values of the four neighboring entries in  $A$  and  $B$ .

With this conjecture, it would become theoretically possible to define vertex operators  $\Delta_+$  and  $\Delta_-$  that operate on diagonal slices of  $AB$  configurations, using the entry poset categorization to generalize the notion of interlacing. It is not immediately clear how these would

be arranged into a sequence to produce the generating function  $W_{(\lambda,\mu,\nu)}(q)$ , other than that commuting two of opposite sign should correspond to a bijective operation on RPPs that removes the corresponding corner from  $\nu$  and possibly pops off a number. Iteratively performing this toggle-like operation would eventually remove all of  $\nu$  and produce exactly a factor of  $W_{(\emptyset,\emptyset,\nu)}(q)$ , leaving behind an object resembling a two-leg RPP — ideally, exactly the same as the objects produced by the bijection mentioned at the beginning of Section 3.1.

# Chapter 4

## Diagonal RSK and Categorizing Bijections

As previously stated, the bijection  $\tau$  defined in Theorem 2.3.9 is equivalent to the map defined in [Pak01], except defined on skew plane partitions instead of reverse ones; however, any fixed SPP  $\sigma$  of shape  $(\emptyset, \emptyset, \lambda)$  contains only finitely many nonzero entries, and so we may truncate it to the boxes in  $[1, N]^2$  for some  $N \in \mathbb{N}$  without losing data. By rotating it  $180^\circ$ , we may express  $\sigma$  as an RPP. This alternate perspective agrees exactly with the description of the algorithm in [Pak01]: toggles pop off corners on the bottom-right of the diagram instead of the top-left, and the definition of a toggle itself is rotated  $180^\circ$  compared to Definition 2.2.1.

Pak's algorithm is one of two known weight-preserving bijections between RPPs of shape  $(\emptyset, \emptyset, \lambda)$  and hook-length-weighted tableaux of shape  $\lambda$  — the other is an algorithm introduced in 1975 by Hillman and Grassl [HG76]. This second map, which we will denote as HG, operates on an RPP of shape  $(\emptyset, \emptyset, \lambda)$  by iteratively decrementing the entries that lie along paths of boxes moving weakly up and right. Each strip corresponds to a hook in the diagram, and for each strip in the RPP that we decrement, we increment the entry of the tableau at the pivot of the corresponding hook. The pivots that are incremented move in a reverse column-major order — bottom to top, left to right — and the inverse map  $\text{HG}^{-1}$

operates by undoing this process, ordering the pivots top to bottom, right to left.

In 2017, Sulzgruber introduced a similar algorithm [Sul17]. It also iteratively decrements paths of boxes corresponding to hooks, but the pivots that are incremented move bottom to top, right to left. The map is equivalent to Pak’s, and so we denote it PS. Sulzgruber’s description of the inverse map  $\text{PS}^{-1}$  therefore orders the pivots top to bottom, left to right.

Also in 2017, Garver and Patrias introduced a new description of  $\text{PS}^{-1}$  using the RSK algorithm [GP17]. Interestingly, they also present an equivalent description of  $\text{HG}^{-1}$  given by slightly changing two parameters in the algorithm. This is the only unified description of HG and PS of which we are aware, and it raises the question of whether there are more bijections given by varying the parameters in other ways. In Section 4.1, we state the general Garver–Patrias algorithm and its specializations that produce  $\text{PS}^{-1}$  and  $\text{HG}^{-1}$ , and in Section 4.2, we prove a number of properties about the algorithm in general. In Section 4.3, we describe the requirements for a variant of Garver–Patrias to be a well-defined bijection and the manner in which we encoded these requirements into an integer programming problem. In Section 4.4, we discuss the results of an exhaustive search through algorithms on  $3 \times 3$  tableaux and conjecture a result for larger tableaux.

## 4.1. The RSK and Garver–Patrias Algorithms

We begin by briefly reviewing the Robinson–Schensted–Knuth (RSK) algorithm [BR38; Sch61a; Knu70] and some of its properties. RSK is a very well-studied algorithm — see, for instance, [Sta23, page 316] — and there are many equivalent formulations of it. Here, we treat RSK as an algorithm on words of positive integers that produces a semistandard Young tableau (SSYT), which is identical to an RPP, except that columns must increase strictly instead of weakly.

**Definition 4.1.1.** Let  $P$  be an SSYT and let  $k \in \mathbb{N}$ . The **row insertion** of  $k$  into  $P$ , denoted  $P \leftarrow k$ , is an SSYT defined in the following manner. If  $k$  is at least as large as every entry in the first row of  $P$ , then  $P \leftarrow k$  is given by adding  $k$  to the end of the first row of

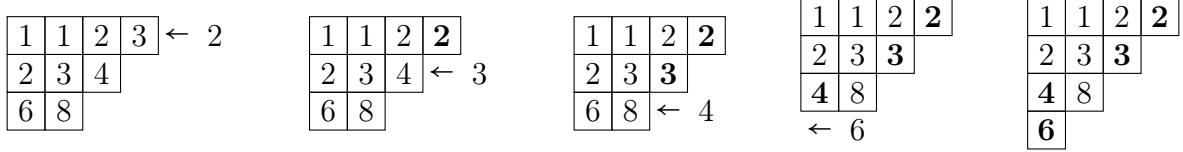


Figure 4.1.1: Inserting a 2 into an SSYT. At each step, we draw the insertion arrow after the row under consideration. First, the 2 bumps the 3 in the first row; then that 3 is inserted into the second row, where it bumps the 4. That 4 is inserted into the third row and bumps the 6, which then is inserted into the (empty) fourth row, where it remains.

$P$ . Otherwise, let  $j$  be the smallest integer so that  $P(1, j) > k$ , let  $r_1$  be the first row of  $P$ , and let  $P'$  be the SSYT given by all of the remaining rows of  $P$ . Finally, let  $r'_1$  be the same as  $r_1$ , but with the  $j$ th entry replaced by  $k$ ; then  $P \leftarrow k$  is defined recursively as an SSYT whose first row is  $r'_1$  and whose subsequent rows are the SSYT  $P' \leftarrow P(1, j)$ .

In practice, row insertion is often thought of as a process that “bumps” entries from one row down to the next. We illustrate this with an example in Figure 4.1.1, where the entries that have been modified are bolded at each step.

**Algorithm 4.1.2.** Let  $B$  be the set of sequences of ordered pairs

$$b = ((i_1, j_1), (i_2, j_2), \dots, (i_n, j_n)),$$

where  $i_k, j_k \in \mathbb{N}$  such that  $b$  is in lexicographic order:  $i_k \leq i_{k+1}$  for all  $1 \leq k < n$ , and whenever  $i_k = i_{k+1}$ ,  $j_k \leq j_{k+1}$ . We typically represent  $b$  as a *biword*

$$b = \begin{pmatrix} i_1 & i_2 & \cdots & i_n \\ j_1 & j_2 & \cdots & j_n \end{pmatrix},$$

The RSK algorithm is a bijection between  $B$  and pairs of SSYTs of identical shape. Given such a sequence  $b$ ,  $\text{RSK}(b) = (P, Q)$ , where

$$P = \emptyset \leftarrow j_1 \leftarrow j_2 \leftarrow \cdots \leftarrow j_n,$$

and  $\emptyset$  denotes the empty SSYT. During the  $k$ th row insertion, let  $(r_k, c_k)$  be the location of the final box modified (i.e. where the insertion process terminates). We construct  $Q$  so that  $Q(r_k, c_k) = i_k$ . Amazingly,  $Q$  is an SSYT, and the process is invertible [Knu70].

1	1	2	3
2	2	3	
3	3		

1	1	1	2
2	2	2	
3	3		

Figure 4.1.2: The SSYTs  $P$  and  $Q$  (left and right, respectively) that are the output of the RSK algorithm on the biword  $b$  as defined in Equation (4.1).

**Example 4.1.3.** Let  $b$  be the biword

$$b = \begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 \\ 3 & 3 & 3 & 2 & 2 & 2 & 3 & 1 & 1 \end{pmatrix}. \quad (4.1)$$

We compute  $\text{RSK}(b) = (P, Q)$  by first constructing  $P$  by successively row inserting the entire bottom row of  $b$ , left to right, producing the SSYTs in Figure 4.1.2. Each row insertion eventually places an entry at the end of a row and doesn't bump another entry; each of these locations is where we place the corresponding top-row entry of  $b$ .

For our purposes, we need a substantially simpler presentation of the RSK algorithm: given a word  $w = j_1 j_2 \cdots j_n$  for  $j_k \in \mathbb{N}$ , we may simply define  $P(w) = \emptyset \leftarrow j_1 \leftarrow j_2 \leftarrow \cdots \leftarrow j_n$  as the SSYT formed by successively row inserting the entries of  $w$ , without regard for the  $Q$  tableau. Specifically, the Garver–Patrias algorithm depends only on the *shape* of  $P(w)$ .

The Garver–Patrias algorithm is defined in [GP17], along with a variant; we describe it here in more generality by introducing the concept of order and label tableaux, which generalizes both the original algorithm and its variant.

**Algorithm 4.1.4.** Fix a Young diagram  $\lambda$  and define the **order tableau** and **label tableau**  $\mathcal{O}$  and  $\mathcal{L}$  to be tableaux of shape  $\lambda$  that contain distinct positive integer entries (typically,  $1, 2, \dots, |\lambda|$ ). The Garver–Patrias algorithm  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is a weight-preserving map from hook-length-weighted tableaux of shape  $\lambda$  to RPPs of shape  $\lambda$ . Some choices of  $\mathcal{O}$  and  $\mathcal{L}$  may fail to make  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  well-defined; we indicate when this occurs subsequently.

Let  $t$  be a tableau of shape  $\lambda$  and let  $(i_c, j_c)$  be the bottom-right-most box in the diagonal of content  $c$ . Let  $r_c$  be the rectangle of boxes

$$r_c = \{1, 2, \dots, i_c\} \times \{1, 2, \dots, j_c\},$$

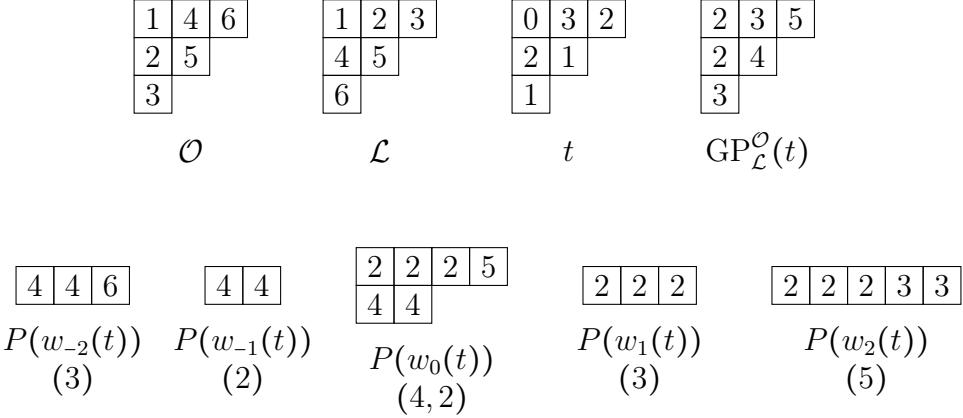


Figure 4.1.3: Top: an example of choices for  $\mathcal{O}$  and  $\mathcal{L}$  that make  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ , and its output on an example tableau  $t$ . Bottom: the outputs of applying the RSK algorithm to each word  $w_c(t)$  formed from the tableau  $t$ , and their shapes, which are placed as diagonals to form the output RPP  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}(t)$ .

and let  $o_c = \mathcal{O} \cap r_c$  and  $l_c = \mathcal{L} \cap r_c$  be the  $i_c \times j_c$  tableaux given by the entries of  $\mathcal{O}$  and  $\mathcal{L}$  in  $r_c$ . Now  $o_c$  contains  $m = i_c j_c$  distinct positive integers; list the cells in  $r_c$  as  $(a_1, b_1), (a_2, b_2), \dots, (a_m, b_m)$ , so that

$$o_c(a_1, b_1) < o_c(a_2, b_2) < \dots < o_c(a_m, b_m).$$

Now form a word  $w_c(t)$  by taking the entries of  $l_c$  in this order, with multiplicities given by the entries of  $t$  in  $r_c$ :

$$w_c(t) = l_c(a_1, b_1)^{t(a_1, b_1)} l_c(a_2, b_2)^{t(a_2, b_2)} \dots l_c(a_m, b_m)^{t(a_m, b_m)} \quad (4.2)$$

Apply the RSK algorithm to  $w_c(t)$  to find the SSYT  $P(w_c(t))$ , and define an RPP  $\rho$  of shape  $\lambda$  whose diagonal of content  $c$ , read from bottom-right to top-left, is  $\text{sh}(P(w_c(t)))$  (i.e. the partition given by its row lengths). If this process ever produces an SSYT with more rows than can fit into the content- $c$  diagonal, then we terminate with failure and declare that  $\mathcal{O}$  and  $\mathcal{L}$  are an invalid combination of order and labels (and that  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is not well-defined).

**Example 4.1.5.** Let  $\mathcal{O}$ ,  $\mathcal{L}$ , and  $t$  be as in Figure 4.1.3. We compute  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}(t)$  by first computing the word  $w_c(t)$  for each of the five diagonals. When  $c = 0$ , for example, the rectangle  $r_0$  is

$$r_0 = \{(1, 1), (1, 2), (2, 1), (2, 2)\},$$

and listing these cells in order of increasing entries in  $\mathcal{O}$  gives the ordering

$$(1, 1), (2, 1), (1, 2), (2, 2).$$

We now form the word  $w_0(t)$  from the values of  $\mathcal{L}$  in these cells, with multiplicities given by  $t$ :

$$\begin{aligned} w_0(t) &= \mathcal{L}(1, 1)^{t(1,1)} \mathcal{L}(2, 1)^{t(2,1)} \mathcal{L}(1, 2)^{t(1,2)} \mathcal{L}(2, 2)^{t(2,2)} \\ &= 1^0 4^2 2^3 5^1 = 442225. \end{aligned}$$

Computing the rest of the words,

$$\begin{aligned} w_{-2}(t) &= 446 \\ w_{-1}(t) &= 44 \\ w_0(t) &= 442225 \\ w_1(t) &= 222 \\ w_2(t) &= 22233. \end{aligned}$$

We now apply the RSK algorithm to each word, producing the middle row of diagrams in Figure 4.1.3, and record their shapes in the corresponding diagonals of an empty tableau, finally resulting in  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}(t)$ , the top-right diagram in the figure.

We briefly introduce some useful notation before summarizing the core result of [GP17].

**Definition 4.1.6.** Fix a partition  $\lambda$ . Let  $\mathcal{L}_{\text{PS}}$  be the tableau of shape  $\lambda$  given by listing the numbers  $1, 2, \dots, |\lambda|$  in row-major order (i.e. so that the cells  $(i, j)$  are listed in lexicographic order). Similarly, let  $\mathcal{O}_{\text{PS}}$  be the tableau of shape  $\lambda$  given by listing the numbers  $1, 2, \dots, |\lambda|$  in column-major order (i.e. so that the transpose of  $\mathcal{O}_{\text{PS}}$  is in row-major order). Finally, let  $\mathcal{L}_{\text{HG}}$  and  $\mathcal{O}_{\text{HG}}$  be identical to  $\mathcal{L}_{\text{PS}}$  and  $\mathcal{O}_{\text{PS}}$ , respectively, but with each row reversed.

In [GP17], the authors prove that for any partition  $\lambda$ , if  $\mathcal{O} = \mathcal{O}_{\text{PS}}$  and  $\mathcal{L} = \mathcal{L}_{\text{PS}}$ , then the resulting algorithm  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is a well-defined bijection that coincides with  $\text{PS}^{-1}$ ; we note that this is the case in Example 4.1.5, and that the RPP and tableau in that example are equal

to those in Example 2.3.10. Similarly, if  $\mathcal{O} = \mathcal{O}_{\text{HG}}$  and  $\mathcal{L} = \mathcal{L}_{\text{HG}}$ , then  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  coincides with the inverse Hillman–Grassl algorithm  $\text{HG}^{-1}$ . In the remainder of this chapter, we examine other possibilities for  $\mathcal{O}$  and  $\mathcal{L}$  that make  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  well-defined, weight-preserving, and bijective.

## 4.2. Properties of GP

For a pair  $(\mathcal{O}, \mathcal{L})$  to produce a valid (i.e. well-defined and bijective) algorithm  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ , we need to understand the shape of the output of the RSK algorithm. Given a word  $w$ , the number of rows and columns of  $P(w)$  was given in terms of the maximum increasing and decreasing subsequences of  $w$  by Schensted [Sch61b]. This result was later extended by Greene to one relating all row and column lengths to generalizations of increasing and decreasing subsequences [Gre74]; it is this result that we use.

**Definition 4.2.1.** [Gre74] Let  $w = j_1 j_2 \cdots j_n$  for  $j_k \in \mathbb{N}$  be a word. A  **$k$ -increasing subword** of  $w$  is a (not necessarily contiguous) subword that does not itself contain a strictly decreasing subword (again, not necessarily contiguous) of length  $k + 1$ . Similarly, a  **$k$ -decreasing subword** of  $w$  is one that does not contain a **weakly** increasing subword of length  $k + 1$ . We note that  $k$ -increasing subwords are unions of  $k$  disjoint weakly increasing subwords, and we denote by  $a_k(w)$  and  $d_k(w)$  the lengths of the longest  $k$ -increasing and  $k$ -decreasing subwords of  $w$ .

**Theorem 4.2.2.** [Gre74] Let  $w = j_1 j_2 \cdots j_n$  for  $j_k \in \mathbb{N}$  be a word and let  $\lambda = \text{sh}(P(w))$ . Then for all  $k \in \mathbb{N}$ ,

$$\begin{aligned} a_k(w) &= \lambda_1 + \cdots + \lambda_k \\ d_k(w) &= \lambda'_1 + \cdots + \lambda'_k. \end{aligned}$$

In particular, the length of the maximum strictly decreasing subword,  $d_1(w)$ , is the number of rows of  $P(w)$ .

In the case of the Garver–Patrias algorithm, the words  $w_c(t)$  are particularly well-behaved: in fact, we can determine  $a_k(w_c(t))$  and  $d_k(w_c(t))$  simply by computing  $w_c$  for the tableau of all 1s.

**Proposition 4.2.3.** Fix a partition  $\lambda$  and order and label tableaux  $\mathcal{O}$  and  $\mathcal{L}$ , let  $\mathbf{1}$  be the tableau of shape  $\lambda$  consisting of all 1s, and let  $t$  be any tableau of shape  $\lambda$ . Let  $c$  be any diagonal of  $\lambda$ ,  $w_c(\mathbf{1}) = l_1 l_2 \cdots l_n$ , and  $w_c(t) = w_c(\mathbf{1}) = l_1^{j_1} l_2^{j_2} \cdots l_n^{j_n}$ . Let  $S_{\mathbf{1}}$  be the set of subwords of  $w_c(\mathbf{1})$ , and let  $S_t$  be the set of subwords of  $w_c(t)$ .

Define a map  $r : S_{\mathbf{1}} \rightarrow S_t$  by

$$r(l_{i_1} l_{i_2} \cdots l_{i_k}) = l_{i_1}^{j_1} l_{i_2}^{j_2} \cdots l_{i_k}^{j_k};$$

that is,  $r(s)$  is the maximum-length subword of  $w_c(t)$  consisting of only letters in  $s$ . Let  $A_{\mathbf{1}}^k$  be the set of maximum-length  $k$ -increasing subwords of  $w_c(\mathbf{1})$ , and let  $A_t^k$  be the set of maximum-length  $k$ -increasing subwords of  $w_c(t)$ . Then  $A_t^k \subseteq r(A_{\mathbf{1}}^k)$ .

*Proof.* We show the result for  $k$ -increasing subwords. If  $a$  is a maximum  $k$ -increasing subword of  $w_c(t)$ , then

$$a = l_{i_1}^{j_1} l_{i_2}^{j_2} \cdots l_{i_k}^{j_k}$$

for some  $i_1, \dots, i_k$ : every letter  $l_{i_k}$  present must have the maximum multiplicity, since increasing the multiplicity of any letter present does not affect whether the word is  $k$ -increasing. Let  $a' \in S_{\mathbf{1}}$  be the word

$$a' = l_{i_1} l_{i_2} \cdots l_{i_k},$$

which is necessarily  $k$ -increasing. It may not be a maximal  $k$ -increasing subword of  $w_c(\mathbf{1})$ , but if not, we can by definition add more letters of  $w_c(\mathbf{1})$  to produce a word  $a''$ . These added letters (i.e.  $a'' \setminus a'$ ) are necessarily not in  $w_c(t)$ , so  $r(a'') = a$ .  $\square$

We can also efficiently determine when a potential algorithm  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is guaranteed to be well-defined or not surjective.

**Proposition 4.2.4.** Fix a partition  $\lambda$  and order and label tableaux  $\mathcal{O}$  and  $\mathcal{L}$ , and let  $\mathbf{1}$  be the tableau of shape  $\lambda$  consisting of all 1s. Let  $k_c = \min \{i_c, j_c\}$  be the length of diagonal  $c$ . Then  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is well-defined if and only if  $d_1(w_c(\mathbf{1})) \leq k_c$ , and if  $d_1(w_c(\mathbf{1})) < k_c$ ,  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is not surjective.

*Proof.* By Theorem 4.2.2,  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  being well-defined is equivalent to  $d_1(w_c(t)) \leq k_c$  for all tableaux  $t$  of shape  $\lambda$  and all diagonals  $c$ . For  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  to be surjective, more must be true: for each diagonal  $c$ , there must exist at least one tableau  $t$  so that  $d_1(w_c(t)) = k_c$ .

Now  $d_1(w_c(t))$  is the length of the maximum strictly decreasing subword of  $w_c(t)$ , and  $w_c(t)$  consists of blocks of labels repeated zero or more times (see Equation (4.2)); therefore, no label can be used more than once in a decreasing subword of  $w_c(t)$ . In other words,  $d_1(w_c(t)) \leq d_1(w_c(\mathbf{1}))$ , and so  $d_1(w_c(\mathbf{1})) \leq k_c$  if and only if  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is well-defined. Moreover, if  $d_1(w_c(\mathbf{1})) < k_c$ , then  $d_1(w_c(t)) < k_c$  for all  $t$ , meaning  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  can never output RPPs with nonzero values in the top-left-most box of diagonal  $c$  and therefore fails to be surjective.  $\square$

We also need  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  to be weight-preserving for it to count as a valid algorithm; the next proposition guarantees that this always occurs so long as  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is well-defined.

**Proposition 4.2.5.** Let  $\mathcal{O}$  and  $\mathcal{L}$  be order and label tableaux so that  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is well-defined. Then  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is also weight-preserving in the sense of the PS and HG algorithms: if its inputs are weighted by hook length, and its outputs are weighted as usual as summing the entries, then the weight of any input and its corresponding output are equal.

*Proof.* Let  $t$  be a tableau and  $(i, j)$  be a box in it. The value of  $t(i, j)$  causes the label  $\mathcal{L}(i, j)$  to appear  $t(i, j)$  times in exactly the words  $w_c$  whose corresponding rectangles  $r_c$  contain  $(i, j)$ . These are exactly the diagonals  $c$  that intersect the (down-right) hook with pivot  $(i, j)$ . Therefore, label  $\mathcal{L}(i, j)$  appears a total of  $t(i, j)h(i, j)$  times across all words  $w_c$ , and each occurrence increases the weight of the output RPP by one.  $\square$

We next relate the validity of an algorithm  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  to the validity of algorithms given by rectangular chunks of  $\mathcal{O}$  and  $\mathcal{L}$ .

**Proposition 4.2.6.** Let  $\mathcal{O}$  and  $\mathcal{L}$  be order and label tableaux of shape  $\lambda$  so that  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is well-defined and bijective, and let  $B = ([i_m, i_M] \times [j_m, j_M]) \cap \mathbb{N}^2$  be a rectangular set of cells so that  $B \subseteq \lambda$ . Let  $\mathcal{O}' = \mathcal{O} \cap B$  and  $\mathcal{L}' = \mathcal{L} \cap B$ , reindexed so that both  $\mathcal{O}'$  and  $\mathcal{L}'$  have top-left cells of  $(1, 1)$ . Then  $\text{GP}_{\mathcal{L}'}^{\mathcal{O}'}$  is well-defined and bijective.

*Proof.* First suppose  $i_m = j_m = 1$ , so that no reindexing is required. Then  $B$  is a rectangle of boxes in the top-left of  $\lambda$ , as in Figure 4.2.1. Let  $(i_c, j_c)$  be the bottom-right-most box in diagonal  $d$  of  $\lambda$ , and let  $r_c$  be the rectangle of boxes weakly up and left of  $(i, j)$ , as in Algorithm 4.1.4. Similarly, let  $(i'_d, j'_d)$  and  $r'_d$  be the bottom-right-most boxes and their

			3		3	3	3
			2	2	2		
-2	-1	0	1	1			
-2	-1						
-2							

Figure 4.2.1: A Young diagram of shape  $\lambda = (7, 6, 4, 2)$  and a rectangle  $B$  with  $B \subseteq \lambda$ . The bottom-right-most cells in  $B$  have their diagonal's content written in them, and the bottom-right-most cells in diagonal  $c$  in  $\lambda$  have  $d = b(c)$  written in them in the notation of Proposition 4.2.6. The diagonal containing the corner box  $(i_M, j_M) = (3, 4)$  of  $B$  is  $d_0 = 1$ .

rectangles in  $B$ , for the diagonals  $d$  for which they are defined. Finally, let  $d_0$  be the diagonal of  $B$  along which the corner box  $(i_M, j_M)$  lies.

Define a function  $b$  on the set of diagonals of  $\lambda$  by  $b(c) = d$  if  $r'_d = r_c \cap B$ . Since every rectangle  $r'_d$  is distinct and every rectangle  $r_c$  satisfies  $r'_d = r_c \cap B$  for some  $d$ ,  $b$  is well-defined.

If  $d$  is a diagonal in  $B$  with  $d < d_0$ , then the preimage  $b^{-1}(d)$  consists of the diagonals  $c$  with  $j_c = j'_d$ . Similarly, if  $d > d_0$ , then  $b^{-1}(d)$  consists of the diagonals  $c$  with  $i_c = i'_d$  (see Figure 4.2.1). By writing these diagonal conditions in terms of the corner box  $(i'_{d_0}, j'_{d_0})$ , we have that if  $d = b(c)$ , then

$$(i'_d, j'_d) = \begin{cases} (i'_d, j_c), & \text{if } j_c < j'_{d_0} \\ (i_c, j'_d), & \text{if } i_c < i'_{d_0} \\ (i'_{d_0}, j'_{d_0}), & \text{else} \end{cases}. \quad (4.3)$$

The function  $b$  is well-behaved; we now list some of its properties.

1. Since the set of all  $(i_c, j_c) \in \lambda$  (the so-called “ribbon”) is connected and contains at least one box in the first row and in the first column of  $\lambda$ ,  $b$  is surjective.
2. By Equation (4.3), if  $b(c) = d$  and  $d \neq d_0$ , then the length of diagonal  $c$  in  $\lambda$  and diagonal  $d$  in  $B$  are the same.
3. Since the set of boxes  $(i_c, j_c)$  with  $b(c) = d_0$  are weakly right of and below  $(i'_{d_0}, j'_{d_0})$ , the

length of any diagonal  $c$  in  $\lambda$  with  $b(c) = d_0$  is at least as large as the length of diagonal  $d_0$  in  $B$ , and there is at least one  $c$  for which the two are equal (take any diagonal  $c$  with  $(i_c, j_c)$  directly to the right or below  $(i'_{d_0}, j'_{d_0})$ , which exists since the ribbon in  $\lambda$  is connected).

Let  $t'$  be a tableau of shape  $B$ , and let  $t$  be the tableau of shape  $\lambda$  given by  $t'$  in  $B$  and all 0s elsewhere. Let  $\rho = \text{GP}_{\mathcal{L}}^{\mathcal{O}}(t)$  and  $\rho' = \text{GP}_{\mathcal{L}'}^{\mathcal{O}'}(t')$ . We know by hypothesis that  $\rho$  is an RPP, and that  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  acts bijectively; we wish to show the same for  $\rho'$  and  $\text{GP}_{\mathcal{L}'}^{\mathcal{O}'}$ .

If  $d$  is a diagonal of  $B$ , then there is some diagonal  $c$  of  $\lambda$  with  $b(c) = d$ , since  $b$  is surjective. For any such diagonal  $c$ ,  $r'_d = r_c \cap B$  by definition. Since  $t$  contains no nonzero entries other than those in  $B$ , the words  $w_c(t)$  and  $w_d(t')$  are equal, and therefore diagonal  $c$  of  $\rho$  and diagonal  $d$  of  $\rho'$  are equal (as partitions, reading from bottom-right to top-left). In particular, since  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is well-defined,  $P(w_c(t)) = P(w_d(t'))$  has a number of rows no larger than the length of diagonal  $c$  in  $\lambda$ . By properties 2 and 3 of  $b$ , this guarantees that the shape of  $P(w_d(t'))$  fits in  $B$ , making  $\text{GP}_{\mathcal{L}'}^{\mathcal{O}'}$  well-defined.

Let  $T$  be the set of tableaux of shape  $\lambda$  that are zero outside of  $B$ , and let  $T'$  be the set of tableaux of shape  $B$ . Let  $R = \text{GP}_{\mathcal{L}}^{\mathcal{O}}(T)$  and  $R' = \text{GP}_{\mathcal{L}'}^{\mathcal{O}'}(T')$ , and define a map  $\varphi : R' \rightarrow R$ , where  $\varphi(\rho')$  is given by writing diagonal  $d$  of  $\rho'$  in every diagonal  $c$  of  $\rho$  with  $b(c) = d$ , from bottom-right to top-left. Then the following diagram commutes, where  $\iota$  is the inclusion of  $T'$  into  $T$ :

$$\begin{array}{ccc} T' & \xrightarrow{\text{GP}_{\mathcal{L}'}^{\mathcal{O}'}} & R' \\ \iota \downarrow & \circlearrowleft & \downarrow \varphi \\ T & \xrightarrow{\text{GP}_{\mathcal{L}}^{\mathcal{O}}} & R \end{array}$$

Since  $\text{GP}_{\mathcal{L}'}^{\mathcal{O}'} = \varphi^{-1} \circ \text{GP}_{\mathcal{L}}^{\mathcal{O}} \circ \iota$  and all three of the maps on the right are bijective, so is  $\text{GP}_{\mathcal{L}'}^{\mathcal{O}'}$ .

It remains to show that the result holds when  $(i_m, j_m) \neq (1, 1)$ . Let  $B_1 = ([1, i_M] \times [1, j_M]) \cap \mathbb{N}^2$  be the rectangle given by extending  $B$  up and left to the boundary of  $\lambda$ , let  $\mathcal{O}_1$  and  $\mathcal{L}_1$  be the restrictions of  $\mathcal{O}$  and  $\mathcal{L}$  to  $B_1$ , and apply the existing logic to them to show that  $\text{GP}_{\mathcal{L}_1}^{\mathcal{O}_1}$  is well-defined and bijective. Since  $B$  is in the bottom-right corner of  $B_1$ ,  $\text{GP}_{\mathcal{L}'}^{\mathcal{O}'}$  is a restriction of  $\text{GP}_{\mathcal{L}_1}^{\mathcal{O}_1}$  (since both algorithms place the shapes of the RSK outputs into diagonals starting

from the bottom-right). Therefore,  $\text{GP}_{\mathcal{L}'}^{\mathcal{O}'}$  is well-defined and bijective.  $\square$

For a fixed shape  $\lambda$ , let  $T_n$  be the set of hook-length-weighted tableaux of shape  $\lambda$  and weight exactly  $n$ , where  $n \in \mathbb{N}_{\geq 0}$ . Similarly, let  $R_n$  be the set of RPPs of shape  $\lambda$  and weight  $n$ . If an algorithm  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is well-defined and injective, then by Proposition 4.2.5, it is weight-preserving, and so  $\text{GP}_{\mathcal{L}}^{\mathcal{O}} : T_n \rightarrow R_n$  injectively. Since  $\text{PS} : T_n \rightarrow R_n$  is a bijection,  $|T_n| = |R_n|$ , and since that cardinality is finite,  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  must be bijective if it is injective.

### 4.3. The Injectivity of GP as an Integer Program

Our goal in this section is to determine which algorithms  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  are well-defined and injective. Given an order-label pair  $(\mathcal{O}, \mathcal{L})$ , Proposition 4.2.4 lets us determine whether  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is well-defined by examining  $d_1(w_c(\mathbf{1}))$ , and if it is well-defined, then it is weight-preserving. The question that remains is how we can prove a given algorithm is injective: that is, that no solution to  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}(t) = \text{GP}_{\mathcal{L}}^{\mathcal{O}}(t')$  exists with  $t \neq t'$ . We approach this restriction by encoding it as an integer program; we briefly review integer programs in general before discussing our work.

An integer program is an optimization problem of many variables over the positive integers and is a standard tool in combinatorial optimization. For variables  $x_i \in \mathbb{N}_{\geq 0}$ , we wish to maximize or minimize an  $\mathbb{R}$ -linear combination of the  $x_i$ , subject to any number of affine linear equations involving the  $x_i$  (and potentially max or min functions) with coefficients in  $\mathbb{R}$ . Solving integer programs in general is computationally difficult (specifically, it is NP-hard), and so software for solving integer programs can be slow even when heavily optimized. However, such software can verify that no solution to a given integer program is possible; to that end, if we encode each entry of two different tableaux  $t$  and  $t'$  as a different variable  $x_i$ , the condition that  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}(t) = \text{GP}_{\mathcal{L}}^{\mathcal{O}}(t')$  and  $t \neq t'$  as a set of linear constraints, and the sum of all of the  $x_i$  as a constraint to minimize, then we can provably verify whether a given algorithm  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is injective.

The difficulties with creating a complete list of bijections now amount to the following: first,

$\mathcal{O}$	$\mathcal{L}$
$\begin{array}{ c c c } \hline 1 & 2 & 5 \\ \hline 3 & 4 & 6 \\ \hline \end{array}$	$\begin{array}{ c c c } \hline 1 & 2 & 4 \\ \hline 3 & 5 & 6 \\ \hline \end{array}$
$\mathcal{O}'_1$	$\mathcal{L}'_1$
$\begin{array}{ c c } \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 1 & 2 \\ \hline 3 & 5 \\ \hline \end{array}$
$\mathcal{O}'_2$	$\mathcal{L}'_2$
$\begin{array}{ c c } \hline 2 & 5 \\ \hline 4 & 6 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 2 & 4 \\ \hline 5 & 6 \\ \hline \end{array}$
$\begin{array}{ c c } \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}$
$\begin{array}{ c c } \hline 1 & 3 \\ \hline 2 & 4 \\ \hline \end{array}$	$\begin{array}{ c c } \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}$

Figure 4.3.1: A pair  $(\mathcal{O}, \mathcal{L})$  for a potential  $2 \times 3$  algorithm  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  (top). The order and label tableaux produced by restricting  $\mathcal{O}$  and  $\mathcal{L}$  to the left and right  $2 \times 2$  blocks (middle). Reindexing each to contain the numbers 1–4 in the same relative order, which represents the same algorithm (bottom).

the search space is extremely large. We consider  $3 \times 3$  tableaux, so there are  $9!^2 \approx 1.32 \cdot 10^{11}$  possibilities for pairs  $(\mathcal{O}, \mathcal{L})$ . Second, determining injectivity is possible with an integer program, but prohibitively slow for our purposes. To address both problems, we aggressively filter out as many potential algorithms as possible with tests that are faster than solving a full integer program. In a lower-level compiled language (Julia), we implement the following:

1. A function which determines whether every word  $w_c(\mathbf{1})$  for a given pair  $(\mathcal{O}, \mathcal{L})$  has the correct length of a maximum decreasing subword.
2. A memoized version of RSK that we use to implement  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  for arbitrary  $\mathcal{O}$  and  $\mathcal{L}$ .
3. A function that evaluates  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  on all tableaux of the appropriate shape and with entries at most a given bound (typically 1) and determines if the outputs are all unique.

We then run this final function in parallel on all  $2 \times 2$  order-label pairs (for tableaux with maximum entry 1) to produce a list of 24 potentially valid  $2 \times 2$  algorithms. We then do the same for all  $2 \times 3$  algorithms, but before exhaustively testing each pair  $(\mathcal{O}, \mathcal{L})$  on all tableaux with maximum entry 2, we check if every  $2 \times 2$  block of  $\mathcal{O}$  and  $\mathcal{L}$  is itself a valid  $2 \times 2$

algorithm; if not, then Proposition 4.2.6 guarantees  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is not valid. For example, we can determine that the order-label pair in Figure 4.3.1 is invalid before plugging in any inputs: while restricting to the rightmost  $2 \times 2$  block in both  $\mathcal{O}$  and  $\mathcal{L}$  produces the pair  $(\mathcal{O}_2, \mathcal{L}_2)$ , which is a valid  $2 \times 2$  algorithm (in fact, it is  $\text{PS}^{-1}$ ), restricting to the leftmost  $2 \times 2$  block results in an algorithm not in the list of possible  $2 \times 2$  ones.

This filtering reduces the runtime to be tractable, though still long: filtering a list of possible  $3 \times 3$  algorithms took about 2 hours of runtime on a 12th generation Intel i5 using all 16 threads. In the end, we produced a list of about 1500  $3 \times 3$  order-label pairs that were well-defined and injective on tableaux with maximum entry 1. Given this list, what remains is to construct an integer program for each to determine whether it is in fact injective on all tableaux; we detail that process now.

Fix  $\mathcal{O}$  and  $\mathcal{L}$ . For  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  to fail to be injective, we need two distinct tableaux  $t$  and  $t'$  to produce identical outputs: that is, then we need  $\text{sh}(P(w_c(t))) = \text{sh}(P(w_c(t')))$  for all diagonals  $c$ . Equivalently,  $a_k(w_c(t)) = a_k(w_c(t'))$  for all  $c$  and all  $k$ . The benefit of presenting this equality in terms of  $k$ -increasing subwords instead of the RSK algorithm is that it is compatible with the language of integer programs, letting us encode the requirement that  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}(t) = \text{GP}_{\mathcal{L}}^{\mathcal{O}}(t')$ .

To implement these requirements, we use the Python interface to the Gurobi integer programming problem solver [GO25]; we include our code in the supplementary materials and also briefly summarize our approach here.

We assign each entry of  $t$  and  $t'$  a variable, and then for each diagonal  $c$ , we do the following:

1. Compute the word  $w_c(\mathbf{1})$  on the tableau of all 1s.
2. Determine the maximum-length  $k$ -increasing subwords of  $w_c(\mathbf{1})$ .
3. Determine the lengths of the possible maximum  $k$ -increasing subwords of  $w_c(t)$  and  $w_c(t')$  in terms of the entries of  $t$  and  $t'$ , using Proposition 4.2.3.
4. Set a variable  $m_c$  equal to the maximum length of all possible maximum  $k$ -increasing subwords of  $w_c(t)$ , and similarly for a variable  $m'_c$ .

- Add a requirement that  $m_c = m'_c$ .

Finally, we add a variable for each absolute difference  $|t(i, j) - t'(i, j)|$ , add another variable equal to the sum of all of these difference variables, and then add a requirement that this sum is at least 1. We then instruct Gurobi to find the solution for  $t$  and  $t'$  that minimizes the sum of all of the entries of both, or to prove that no such  $t$  and  $t'$  exist (i.e. that the model is “infeasible”).

After generating a list of all pairs  $(\mathcal{O}, \mathcal{L})$  producing valid maps  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ , further processing is required: two different maps  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  can in fact coincide. With  $\mathcal{O}$  and  $\mathcal{L}$  as given in Figure 4.1.3, for example,  $\text{GP}_{\mathcal{L}}^{\mathcal{O}} = \text{GP}_{\mathcal{O}}^{\mathcal{L}}$ . To group identical maps, we construct another integer program akin to the first; given two order-label pairs  $(\mathcal{O}_1, \mathcal{L}_1)$  and  $(\mathcal{O}_2, \mathcal{L}_2)$ , this program finds a minimal-weight tableau  $t$  with  $\text{GP}_{\mathcal{L}_1}^{\mathcal{O}_1}(t) \neq \text{GP}_{\mathcal{L}_2}^{\mathcal{O}_2}(t)$ , or determines that no such tableau exists (i.e. that  $\text{GP}_{\mathcal{L}_1}^{\mathcal{O}_1} = \text{GP}_{\mathcal{L}_2}^{\mathcal{O}_2}$ ). Searching through the list of valid order-label pairs and iteratively building a collection of distinct maps by checking each new one against the existing list, we can partition the order-label pairs into groups of pairs  $(\mathcal{O}, \mathcal{L})$  that all produce equal maps  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ .

## 4.4. Results for $3 \times 3$ Tableaux

Among the very large search space for maps  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  on  $3 \times 3$  tableaux, strikingly few choices of order and label tableaux produce well-defined bijections. Exactly 192 pairs  $(\mathcal{O}, \mathcal{L})$  are valid, but not all of them produce distinct maps — in fact, there are only 8 distinct maps, each of which is produced by exactly 24 different pairs  $(\mathcal{O}, \mathcal{L})$ . We list all 192 in the appendix and give a representative for each in Figure 4.4.1. These maps and tableaux exhibit a high degree of structure and symmetry, and with the aim of conjecturing results about larger tableaux, we detail much of that structure.

For each of the 192 pairs  $(\mathcal{O}, \mathcal{L})$  producing a valid map  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ , the pair  $(\mathcal{L}, \mathcal{O})$  also produces a valid map, and  $\text{GP}_{\mathcal{O}}^{\mathcal{L}} = \text{GP}_{\mathcal{L}}^{\mathcal{O}}$ . Each pair also has identical inequalities between adjacent terms (i.e.  $\mathcal{O}(i, j) < \mathcal{O}(i+1, j)$  if and only if  $\mathcal{L}(i, j) < \mathcal{L}(i+1, j)$ , and similarly for  $(i, j)$  and

1	4	7
2	5	8
3	6	9

1	2	3
4	5	6
7	8	9

$\text{PS}^{-1}$

7	4	1
8	5	2
9	6	3

3	2	1
6	5	4
9	8	7

$\text{HG}^{-1}$

2	5	8
1	4	7
3	6	9

4	5	6
1	2	3
7	8	9

$\text{PS}_r^{-1}$

8	5	2
7	4	1
9	6	3

6	5	4
3	2	1
9	8	7

$\text{HG}_r^{-1}$

4	1	7
5	2	8
6	3	9

2	1	3
5	4	6
8	7	9

$\text{PS}_c^{-1}$

4	7	1
5	8	2
6	9	3

2	3	1
5	6	4
8	9	7

$\text{HG}_c^{-1}$

5	2	8
4	1	7
6	3	9

5	4	6
2	1	3
8	7	9

$\text{PS}_{rc}^{-1}$

5	8	2
4	7	1
6	9	3

5	6	4
2	3	1
8	9	7

$\text{HG}_{rc}^{-1}$

Figure 4.4.1: Representatives for each of the 8 groups of order-label pairs producing distinct maps  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ . Left: variants of the Pak–Sulzgruber algorithm given by swapping rows 1 and 2, columns 1 and 2, or both in both  $\mathcal{O}$  and  $\mathcal{L}$ . Right: the same variants, but for the Hillman–Grassl algorithm. In each pair, the order tableau is given on the left and the label tableau on the right.

	Descending Diagonals	Ascending Diagonals
$\text{PS}^{-1}$	Main	All
$\text{PS}_r^{-1}$	Main	
$\text{PS}_c^{-1}$	Main	
$\text{PS}_{rc}^{-1}$	Main	All
$\text{HG}^{-1}$	All	Off
$\text{HG}_r^{-1}$		Off
$\text{HG}_c^{-1}$		Off
$\text{HG}_{rc}^{-1}$	Main	

Table 4.1: Symmetries of every order-label pair  $(\mathcal{O}, \mathcal{L})$  producing one of the eight distinct maps.

$(i, j + 1))$ .

Moreover, each group has a representative order-label pair that is similar to the pair for either  $\text{PS}^{-1}$  or  $\text{HG}^{-1}$ . One group has the first and second rows of both  $\mathcal{O}$  and  $\mathcal{L}$  swapped compared to  $\mathcal{O}_{\text{PS}}$  and  $\mathcal{L}_{\text{PS}}$ ; we denote the resulting map  $\text{PS}_r^{-1}$ . Similarly, another group has the first and second columns swapped, and a third has both the first and second rows and the first and second columns swapped. We denote these maps  $\text{PS}_c^{-1}$  and  $\text{PS}_{rc}^{-1}$ . There are equivalent groups for swapping the first and second rows, columns, or both from  $\mathcal{O}_{\text{HG}}$  and  $\mathcal{L}_{\text{HG}}$ ; we denote these maps  $\text{HG}_r^{-1}$ ,  $\text{HG}_c^{-1}$ , and  $\text{HG}_{rc}^{-1}$ .

Moreover, the diagonals of these 192 pairs  $(\mathcal{O}, \mathcal{L})$  often have symmetric patterns. Define a **descending** diagonal of a tableau to one from top-left to bottom-right, and an **ascending** diagonal to be one from bottom-left to top-right. For example, the main diagonal of a  $3 \times 3$  tableau is descending, and the off diagonal is ascending. Then many descending and ascending diagonals of  $\mathcal{O}$  and  $\mathcal{L}$  are equal up to permutation when  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is a valid algorithm.

- Let  $d$  be any descending diagonal. Then the entries of  $\mathcal{O}$  and  $\mathcal{L}$  along diagonal  $d$  are the same up to permutation for all 24 pairs  $(\mathcal{O}, \mathcal{L})$  producing the map  $\text{HG}^{-1}$ .

	Can reverse the entries of $\mathcal{O}$ and $\mathcal{L}$	Can transpose $\mathcal{O}$ and $\mathcal{L}$	Can rotate $\mathcal{O}$ and $\mathcal{L}$ 180°
$\text{PS}^{-1}$	Yes	Yes	Yes
$\text{PS}_r^{-1}$	Yes		
$\text{PS}_c^{-1}$	Yes		
$\text{PS}_{rc}^{-1}$	Yes	Yes	
$\text{HG}^{-1}$	Yes	Yes	Yes
$\text{HG}_r^{-1}$	Yes		
$\text{HG}_c^{-1}$	Yes		
$\text{HG}_{rc}^{-1}$	Yes	Yes	

Table 4.2: Operations we can apply to the order-label pairs  $(\mathcal{O}, \mathcal{L})$  in various groups that preserve the map produced.

- Let  $(\mathcal{O}, \mathcal{L})$  be a pair producing a map other than  $\text{HG}_r^{-1}$  or  $\text{HG}_c^{-1}$ . Then the entries along the main diagonal of  $\mathcal{O}$  and the main diagonal of  $\mathcal{L}$  are the same up to permutation.
- Let  $d$  be any descending diagonal. Then the entries of  $\mathcal{O}$  and  $\mathcal{L}$  along diagonal  $d$  are the same up to permutation for all 48 pairs  $(\mathcal{O}, \mathcal{L})$  producing the map  $\text{PS}^{-1}$  or  $\text{PS}_{rc}^{-1}$ .
- Let  $(\mathcal{O}, \mathcal{L})$  be a pair producing a map other than  $\text{PS}_r^{-1}$ ,  $\text{PS}_c^{-1}$ , and  $\text{HG}_{rc}^{-1}$ . Then the entries along the off diagonal of  $\mathcal{O}$  and the main diagonal of  $\mathcal{L}$  are the same up to permutation.

With the pair  $(\mathcal{O}, \mathcal{L})$  producing the map  $\text{PS}_{rc}^{-1}$  in Figure 4.4.1, for example, the main diagonals of both  $\mathcal{O}$  and  $\mathcal{L}$  consist of the numbers 1, 5, and 9. From top-left to bottom-right, the ascending diagonals of both consist of the following numbers:

$$\{5\} \quad \{2, 4\} \quad \{1, 6, 8\} \quad \{3, 7\} \quad \{9\}$$

We summarize these results in Table 4.1; note that this is not an exhaustive list of symmetries, and that we only note when the main/off diagonal or all ascending/descending diagonals are equal between  $\mathcal{O}$  and  $\mathcal{L}$  up to permutation. Perhaps more notable than these

particular symmetries is the fact that if *any* ascending or descending diagonal  $d$  is equal up to permutation between  $\mathcal{O}$  and  $\mathcal{L}$  for a pair  $(\mathcal{O}, \mathcal{L})$ , then the same is true for *every* order-label pair producing the same map.

Beyond symmetries between  $\mathcal{O}$  and  $\mathcal{L}$  for individual pairs  $(\mathcal{O}, \mathcal{L})$ , there are symmetries between different pairs within groups.

- Let  $f$  be the involution on  $\{1, 2, \dots, 9\}$  given by  $f(n) = 10 - n$ . Then if  $(\mathcal{O}, \mathcal{L})$  is a pair in any of the eight groups, the pair  $(f(\mathcal{O}), f(\mathcal{L}))$  given by applying  $f$  to each entry of both  $\mathcal{O}$  and  $\mathcal{L}$  produces a valid map  $\text{GP}_{f(\mathcal{L})}^{f(\mathcal{O})}$ . Moreover,  $\text{GP}_{f(\mathcal{L})}^{f(\mathcal{O})} = \text{GP}_{\mathcal{L}}^{\mathcal{O}}$ .
- If  $(\mathcal{O}, \mathcal{L})$  is a pair in any of the eight groups, then the pair  $(\mathcal{O}^\top, \mathcal{L}^\top)$  given by transposing both  $\mathcal{O}$  and  $\mathcal{L}$  produces a valid map  $\text{GP}_{\mathcal{L}^\top}^{\mathcal{O}^\top}$ . This map coincides with  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  exactly when  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is one of PS,  $\text{PS}_{rc}$ , HG, or  $\text{HG}_{rc}$ .
- If  $(\mathcal{O}, \mathcal{L})$  is a pair in any of the eight groups, then the pair  $(\mathcal{O}', \mathcal{L}')$  given by rotating both  $\mathcal{O}$  and  $\mathcal{L}$   $180^\circ$  produces a valid map  $\text{GP}_{\mathcal{L}'}^{\mathcal{O}'}$  only when  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is either PS or HG; otherwise,  $\text{GP}_{\mathcal{L}'}^{\mathcal{O}'}$  is not a valid map at all.

We summarize these results in Table 4.2.

## 4.5. Conjectures for Larger Tableaux

The eight maps listed in Table 4.4.1 present a high degree of structure: every map has a presentation  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  where both  $\mathcal{O}$  and  $\mathcal{L}$  are given by swapping a pair of rows, columns, or both from  $(\mathcal{O}_{\text{PS}}, \mathcal{L}_{\text{PS}})$  or  $(\mathcal{O}_{\text{HG}}, \mathcal{L}_{\text{HG}})$ . There are several conspicuous asymmetries, however: first, rows 2 and 3 or columns 2 and 3 cannot be swapped from either  $(\mathcal{O}_{\text{PS}}, \mathcal{L}_{\text{PS}})$  or  $(\mathcal{O}_{\text{HG}}, \mathcal{L}_{\text{HG}})$  while still producing a valid algorithm. Second,  $3 \times 3$  tableaux are slightly too small to properly distinguish PS from HG: swapping columns 1 and 3 in  $(\mathcal{O}_{\text{PS}}, \mathcal{L}_{\text{PS}})$  produces  $(\mathcal{O}_{\text{HG}}, \mathcal{L}_{\text{HG}})$  for a  $3 \times 3$  tableaux, so it is not clear if swapping columns 1 and 3 is a valid operation on larger tableaux or merely an artifact.

While it is currently prohibitively computationally expensive to completely investigate  $4 \times 4$  tableaux, preliminary examination of algorithms on larger tableaux shows that starting from either  $(\mathcal{O}_{\text{PS}}, \mathcal{L}_{\text{PS}})$  or  $(\mathcal{O}_{\text{HG}}, \mathcal{L}_{\text{HG}})$  and swapping rows/columns 1 and 2 or 1 and 3 still produces a valid map  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ , but no other row/column swaps. This includes tableaux with more than four rows or columns. For nonrectangular tableaux (for which directly swapping rows and columns of different lengths does not preserve the shape of the tableau), the notion of swapping rows and columns is more subtle. It is not amenable to being written completely formally; instead, we explain our notation with a handful of examples.

**Definition 4.5.1.** Let  $\lambda$  be a partition with  $\lambda_1 \geq 3$  and  $\lambda'_1 \geq 3$ , and let  $m, n \in \{1, 2, 3\}$ . We define  $\mathcal{O}_{\text{PS}}^{rm,cn}$  to be the tableau of shape  $\lambda$  written in column-major order, exactly as with  $\mathcal{O}_{\text{PS}}$ , but with the priority of rows 1 and  $m$  and columns 1 and  $n$  swapped. That is, we write the numbers  $1, 2, \dots, |\lambda|$  one column at a time from top to bottom, with the columns ordered left to right; but within each column, we write a number in row  $m$  whenever we would write one in row 1 (if possible), and vice versa. Similarly, when we would begin filling in column 1, we instead fill in column  $n$ , and vice versa.

We define  $\mathcal{L}_{\text{PS}}^{rm,cn}$  identically, but with an original order of row-major instead of column-major. Similarly, we define  $\mathcal{O}_{\text{HG}}^{rm,cn}$  and  $\mathcal{L}_{\text{HG}}^{rm,cn}$  to be the same modifications to reverse column- and row-major order (the orderings of  $\mathcal{O}_{\text{HG}}$  and  $\mathcal{L}_{\text{HG}}$ ), respectively. Note that if  $m = 1$  or  $n = 1$ , we can omit it since swapping a row or column's priority with itself does nothing. For example,  $\mathcal{O}_{\text{PS}}^1 = \mathcal{O}_{\text{PS}}$ .

We give examples of all eight variants in Figure 4.5.1. With this notation, we are prepared to make the following conjecture.

**Conjecture 4.5.2.** Let  $\lambda$  be a partition.

- If  $(\mathcal{O}, \mathcal{L})$  is equal to  $(\mathcal{O}_{\text{PS}}^{rm,cn}, \mathcal{L}_{\text{PS}}^{rm,cn})$  or  $(\mathcal{O}_{\text{HG}}^{rm,cn}, \mathcal{L}_{\text{HG}}^{rm,cn})$  for any  $m, n \in \{1, 2, 3\}$ , then  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is a valid map.
- These 18 maps are the only ones possible of the form  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ , and if  $\lambda'_1 \geq 4$  and  $\lambda_4 \geq 4$ , then they are distinct from one another.

$\begin{array}{ c c c c } \hline 1 & 5 & 8 & 10 \\ \hline 2 & 6 & 9 & \\ \hline 3 & 7 & & \\ \hline 4 & & & \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 1 & 2 & 3 & 4 \\ \hline 5 & 6 & 7 & \\ \hline 8 & 9 & & \\ \hline 10 & & & \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 7 & 4 & 2 & 1 \\ \hline 8 & 5 & 3 & \\ \hline 9 & 6 & & \\ \hline 10 & & & \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 4 & 3 & 2 & 1 \\ \hline 7 & 6 & 5 & \\ \hline 9 & 8 & & \\ \hline 10 & & & \\ \hline \end{array}$
$\mathcal{O}_{\text{PS}}$	$\mathcal{L}_{\text{PS}}$	$\mathcal{O}_{\text{HG}}$	$\mathcal{L}_{\text{HG}}$
$\begin{array}{ c c c c } \hline 2 & 6 & 9 & 10 \\ \hline 1 & 5 & 8 & \\ \hline 3 & 7 & & \\ \hline 4 & & & \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 4 & 5 & 6 & 7 \\ \hline 1 & 2 & 3 & \\ \hline 8 & 9 & & \\ \hline 10 & & & \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 8 & 5 & 3 & 1 \\ \hline 7 & 4 & 2 & \\ \hline 9 & 6 & & \\ \hline 10 & & & \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 7 & 6 & 5 & 4 \\ \hline 3 & 2 & 1 & \\ \hline 9 & 8 & & \\ \hline 10 & & & \\ \hline \end{array}$
$\mathcal{O}_{\text{PS}}^{r2}$	$\mathcal{L}_{\text{PS}}^{r2}$	$\mathcal{O}_{\text{HG}}^{r2}$	$\mathcal{L}_{\text{HG}}^{r2}$
$\begin{array}{ c c c c } \hline 6 & 3 & 1 & 10 \\ \hline 7 & 4 & 2 & \\ \hline 8 & 5 & & \\ \hline 9 & & & \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 3 & 2 & 1 & 4 \\ \hline 7 & 6 & 5 & \\ \hline 9 & 8 & & \\ \hline 10 & & & \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 2 & 6 & 9 & 1 \\ \hline 3 & 7 & 10 & \\ \hline 4 & 8 & & \\ \hline 5 & & & \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 2 & 3 & 4 & 1 \\ \hline 5 & 6 & 7 & \\ \hline 8 & 9 & & \\ \hline 10 & & & \\ \hline \end{array}$
$\mathcal{O}_{\text{PS}}^{c3}$	$\mathcal{L}_{\text{PS}}^{c3}$	$\mathcal{O}_{\text{HG}}^{c3}$	$\mathcal{L}_{\text{HG}}^{c3}$
$\begin{array}{ c c c c } \hline 6 & 3 & 9 & 10 \\ \hline 5 & 2 & 8 & \\ \hline 4 & 1 & & \\ \hline 7 & & & \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 7 & 6 & 8 & 9 \\ \hline 4 & 3 & 5 & \\ \hline 2 & 1 & & \\ \hline 10 & & & \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 6 & 10 & 3 & 1 \\ \hline 5 & 9 & 2 & \\ \hline 4 & 8 & & \\ \hline 7 & & & \\ \hline \end{array}$	$\begin{array}{ c c c c } \hline 8 & 9 & 7 & 6 \\ \hline 4 & 5 & 3 & \\ \hline 1 & 2 & & \\ \hline 10 & & & \\ \hline \end{array}$
$\mathcal{O}_{\text{PS}}^{r3,c2}$	$\mathcal{L}_{\text{PS}}^{r3,c2}$	$\mathcal{O}_{\text{HG}}^{r3,c2}$	$\mathcal{L}_{\text{HG}}^{r3,c2}$

Figure 4.5.1: Variants of row- and column-major order for shape  $(4, 3, 2, 1)$ , as defined in Definition 4.5.1.

- If  $(\mathcal{O}, \mathcal{L})$  is a pair that produces a valid map, then

$$\mathcal{O}(i, j) < \mathcal{O}(i + 1, j) \Leftrightarrow \mathcal{L}(i, j) < \mathcal{L}(i + 1, j)$$

$$\mathcal{O}(i, j) < \mathcal{O}(i, j) \Leftrightarrow \mathcal{L}(i, j + 1) < \mathcal{L}(i, j + 1)$$

for all choices  $(i, j)$  that keep the coordinates within the tableau.

- Any pair  $(\mathcal{O}, \mathcal{L})$  producing a valid map still produces the same one when the entries of both are reversed (i.e. replacing 1 with  $|\lambda|$ , 2 with  $|\lambda| - 1$ , and so on).
- Let  $(\mathcal{O}, \mathcal{L})$  be a order-label pair producing a valid map  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  and let  $o, l$  be the words given by reading the entries of  $\mathcal{O}$  and  $\mathcal{L}$  right-to-left, top-to-bottom. Let  $o'$  and  $l'$  be the words given by reversing  $o$  and  $l$ , respectively, and let  $\mathcal{O}'$  and  $\mathcal{L}'$  be the tableaux given by placing the entries of  $o'$  and  $l'$  back into a tableau of shape  $\lambda$ . Then  $\text{GP}_{\mathcal{L}'}^{\mathcal{O}'}$  is a valid algorithm if and only if  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$  is equal to PS or HG. Moreover, when this is the case,  $\text{GP}_{\mathcal{L}'}^{\mathcal{O}'} = \text{GP}_{\mathcal{L}}^{\mathcal{O}}$ .

Producing a complete list of bijections for  $4 \times 4$  tableaux (or at least some shape larger than  $3 \times 3$ ) would contribute considerably to addressing this conjecture; we plan to investigate this matter in future work. We also note that throughout all brute-force sieving of algorithms of any size, we have never found an algorithm that is injective on the set of tableaux with entries 0, 1, or 2, but not injective overall. It may be possible in general to test only a finite set of tableaux for injectivity and to avoid integer programs completely.

## Appendix A

# A Complete List of Bijections $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ for $3 \times 3$ Tableaux

The following is an exhaustive list of  $3 \times 3$  order-label pairs  $(\mathcal{O}, \mathcal{L})$  producing bijections  $\text{GP}_{\mathcal{L}}^{\mathcal{O}}$ , organized by the map they create in the notation of Section 4.4. The bolded order-label pairs are those given in Figure 4.4.1.

### Maps producing PS<sup>-1</sup>

9	8	7	9	6	3	2	9	8	2	7	4	2	9	8
6	5	4	8	5	2	7	6	5	9	6	3	6	5	4
3	2	1	7	4	1	4	3	1	8	5	1	7	4	1
9	4	7	9	2	3	2	5	8	2	3	4	2	5	8
2	5	8	4	5	6	3	6	9	5	6	7	5	6	7
3	6	1	7	8	1	4	7	1	8	9	1	3	6	9
1	9	8	1	7	4	1	7	4	1	9	8	1	3	4
7	6	5	9	6	3	9	6	3	7	6	5	3	6	9
4	3	2	8	5	2	8	5	2	4	3	2	4	7	1
9	7	6	9	5	2	9	5	2	9	7	6	1	3	4
5	4	3	7	4	1	7	4	1	5	4	3	5	6	7
2	1	8	6	3	8	6	3	8	2	1	8	4	7	2
8	7	6	8	5	2	1	8	7	1	6	3	9	1	2
5	4	3	7	4	1	6	5	4	8	5	2	5	4	3
2	1	9	6	3	9	3	2	9	7	4	9	7	4	9
8	3	6	8	1	2	1	4	7	1	2	3	1	4	7
1	4	7	3	4	5	2	5	8	4	5	6	4	5	6
2	5	9	6	7	9	3	6	9	7	8	9	7	8	9

Maps producing  $\text{PS}_r^{-1}$

6	5	4	8	5	2	7	6	5	9	6	3	7	6	5
9	8	7	9	6	3	2	9	8	2	7	4	9	8	7
3	2	1	7	4	1	4	3	1	8	5	1	7	4	1
2	5	8	4	5	6	3	6	9	5	6	7	4	5	6
9	4	7	9	2	3	2	5	8	2	3	4	9	4	7
3	6	1	7	8	1	4	7	1	8	9	1	7	8	1
7	6	5	9	6	3	9	6	3	7	6	5	3	6	9
1	9	8	1	7	4	1	7	4	1	9	8	1	5	8
4	3	2	8	5	2	8	5	2	4	3	2	4	7	2
5	4	3	7	4	1	7	4	1	5	4	3	1	4	7
9	7	6	9	5	2	9	5	2	9	7	6	9	1	2
2	1	8	6	3	8	6	3	8	2	1	8	2	5	8
5	4	3	7	4	1	6	5	4	1	4	7	3	4	5
8	7	6	8	5	2	1	8	7	1	6	3	9	1	2
2	1	9	6	3	9	3	2	9	7	4	9	6	7	8
1	4	7	3	4	5	2	5	8	4	5	6	6	5	4
8	3	6	8	1	2	1	4	7	1	2	3	2	5	8
2	5	9	6	7	9	3	6	9	7	8	9	7	8	9

Maps producing  $\text{PS}_c^{-1}$

8   9   7	6   9   3	4   9   7	2   9   3	9   2   8	7   2   4	5   2   8	3   2   4
5   6   4	5   8   2	5   2   8	5   4   6	6   7   5	6   9   3	6   3   9	6   5   7
2   3   1	4   7   1	6   3   1	8   7   1	3   4   1	5   8   1	7   4   1	9   8   1
6   9   3	8   9   7	2   9   3	4   9   7	7   2   4	9   2   8	3   2   4	5   2   8
5   8   2	5   6   4	5   4   6	5   2   8	6   9   3	6   7   5	6   5   7	6   3   9
4   7   1	2   3   1	8   7   1	6   3   1	5   8   1	3   4   1	9   8   1	7   4   1
9   1   8	7   1   4	5   1   8	3   1   4	7   1   4	9   1   8	3   1   4	5   1   8
6   7   5	6   9   3	6   3   9	6   5   7	6   9   3	6   7   5	6   5   7	6   3   9
3   4   2	5   8   2	7   4   2	9   8   2	5   8   2	3   4   2	9   8   2	7   4   2
7   9   6	5   9   2	3   9   6	1   9   2	5   9   2	7   9   6	1   9   2	3   9   6
4   5   3	4   7   1	4   1   7	4   3   5	4   7   1	4   5   3	4   3   5	4   1   7
1   2   8	3   6   8	5   2   8	7   6   8	3   6   8	1   2   8	7   6   8	5   2   8
7   8   6	5   8   2	3   8   6	1   8   2	8   1   7	6   1   3	4   1   7	2   1   3
4   5   3	4   7   1	4   1   7	4   3   5	5   6   4	5   8   2	5   2   8	5   4   6
1   2   9	3   6   9	5   2   9	7   6   9	2   3   9	4   7   9	6   3   9	8   7   9
5   8   2	7   8   6	1   8   2	3   8   6	6   1   3	8   1   7	2   1   3	4   1   7
4   7   1	4   5   3	4   3   5	4   1   7	5   8   2	5   6   4	5   4   6	5   2   8
3   6   9	1   2   9	7   6   9	5   2   9	4   7   9	2   3   9	8   7   9	6   3   9

Maps producing  $\text{PS}_{rc}^{-1}$

5 6 4	5 8 2	5 2 8	5 4 6	6 7 5	6 9 3	6 3 9	6 5 7
8 9 7	6 9 3	4 9 7	2 9 3	9 2 8	7 2 4	5 2 8	3 2 4
2 3 1	4 7 1	6 3 1	8 7 1	3 4 1	5 8 1	7 4 1	9 8 1
5 8 2	5 6 4	5 4 6	5 2 8	6 9 3	6 7 5	6 5 7	6 3 9
6 9 3	8 9 7	2 9 3	4 9 7	7 2 4	9 2 8	3 2 4	5 2 8
4 7 1	2 3 1	8 7 1	6 3 1	5 8 1	3 4 1	9 8 1	7 4 1
6 7 5	6 9 3	6 3 9	6 5 7	6 9 3	6 7 5	6 5 7	6 3 9
9 1 8	7 1 4	5 1 8	3 1 4	7 1 4	9 1 8	3 1 4	5 1 8
3 4 2	5 8 2	7 4 2	9 8 2	5 8 2	3 4 2	9 8 2	7 4 2
4 5 3	4 7 1	4 1 7	4 3 5	4 7 1	4 5 3	4 3 5	4 1 7
7 9 6	5 9 2	3 9 6	1 9 2	5 9 2	7 9 6	1 9 2	3 9 6
1 2 8	3 6 8	5 2 8	7 6 8	3 6 8	1 2 8	7 6 8	5 2 8
4 5 3	4 7 1	4 1 7	4 3 5	5 6 4	5 8 2	5 2 8	5 4 6
7 8 6	5 8 2	3 8 6	1 8 2	8 1 7	6 1 3	4 1 7	2 1 3
1 2 9	3 6 9	5 2 9	7 6 9	2 3 9	4 7 9	6 3 9	8 7 9
4 7 1	4 5 3	4 3 5	4 1 7	5 8 2	5 6 4	5 4 6	5 2 8
5 8 2	7 8 6	1 8 2	3 8 6	6 1 3	8 1 7	2 1 3	4 1 7
3 6 9	1 2 9	7 6 9	5 2 9	4 7 9	2 3 9	8 7 9	6 3 9

Maps producing  $HG^{-1}$

6	7	9	2	5	9	6	7	8	2	5	8	6	3	9	2	1	9	6	3	8	2	1	8
3	4	5	1	4	7	3	4	5	1	4	7	7	4	1	5	4	3	7	4	1	5	4	3
8	1	2	8	3	6	9	1	2	9	3	6	8	5	2	8	7	6	9	5	2	9	7	6
7	8	9	3	6	9	7	8	1	3	6	1	7	4	9	3	2	9	7	4	1	3	2	1
4	5	6	2	5	8	4	5	6	2	5	8	8	5	2	6	5	4	8	5	2	6	5	4
1	2	3	1	4	7	9	2	3	9	4	7	1	6	3	1	8	7	9	6	3	9	8	7
8	9	2	4	7	2	8	9	1	4	7	1	8	5	2	4	3	2	8	5	1	4	3	1
5	6	7	3	6	9	5	6	7	3	6	9	9	6	3	7	6	5	9	6	3	7	6	5
1	3	4	1	5	8	2	3	4	2	5	8	1	7	4	1	9	8	2	7	4	2	9	8
2	5	9	6	7	9	2	5	8	6	7	8	2	1	9	6	3	9	2	1	8	6	3	8
1	4	7	3	4	5	1	4	7	3	4	5	5	4	3	7	4	1	5	4	3	7	4	1
8	3	6	8	1	2	9	3	6	9	1	2	8	7	6	8	5	2	9	7	6	9	5	2
3	6	9	7	8	9	3	6	1	7	8	1	3	2	9	7	4	9	3	2	1	7	4	1
2	5	8	4	5	6	2	5	8	4	5	6	6	5	4	8	5	2	6	5	4	8	5	2
1	4	7	1	2	3	9	4	7	9	2	3	1	8	7	1	6	3	9	8	7	9	6	3
4	7	2	8	9	2	4	7	1	8	9	1	4	3	2	8	5	2	4	3	1	8	5	1
3	6	9	5	6	7	3	6	9	5	6	7	7	6	5	9	6	3	7	6	5	9	6	3
1	5	8	1	3	4	2	5	8	2	3	4	1	9	8	1	7	4	2	9	8	2	7	4

Maps producing  $\text{HG}_r^{-1}$

3	4	5	1	4	7	3	4	5	1	4	7	7	4	1	5	4	3
6	7	9	2	5	9	6	7	8	2	5	8	6	3	9	2	1	8
8	1	2	8	3	6	9	1	2	9	3	6	8	5	2	6	3	8
4	5	6	2	5	8	4	5	6	2	5	8	8	5	2	7	4	1
7	8	9	3	6	9	7	8	1	3	6	1	7	4	9	3	2	1
1	2	3	1	4	7	9	2	3	9	4	7	1	6	3	1	8	7
5	6	7	3	6	9	5	6	7	3	6	9	9	6	3	7	6	5
8	9	2	4	7	2	8	9	1	4	7	1	8	5	2	8	5	1
1	3	4	1	5	8	2	3	4	2	5	8	1	7	4	2	7	4
1	4	7	3	4	5	1	4	7	3	4	5	5	4	3	7	4	1
2	5	9	6	7	9	2	5	8	6	7	8	2	1	9	2	1	8
8	3	6	8	1	2	9	3	6	9	1	2	8	7	6	9	7	6
2	5	8	4	5	6	2	5	8	4	5	6	6	5	4	8	5	2
3	6	9	7	8	9	3	6	1	7	8	1	3	2	9	3	2	1
1	4	7	1	2	3	9	4	7	9	2	3	1	8	7	9	8	7
3	6	9	5	6	7	3	6	9	5	6	7	7	6	5	7	6	5
4	7	2	8	9	2	4	7	1	8	9	1	4	3	2	8	5	1
1	5	8	1	3	4	2	5	8	2	3	4	1	9	8	2	7	4

Maps producing  $\text{HG}_c^{-1}$

7	6	9	5	2	9	3	6	9	1	2	9	7	6	8	5	2	8	3	6	8	1	2	8
4	3	5	4	1	7	4	7	1	4	5	3	4	3	5	4	1	7	4	7	1	4	5	3
1	8	2	3	8	6	5	8	2	7	8	6	1	9	2	3	9	6	5	9	2	7	9	6
8	7	9	6	3	9	4	7	9	2	3	9	8	7	1	6	3	1	4	7	1	2	3	1
5	4	6	5	2	8	5	8	2	5	6	4	5	4	6	5	2	8	5	8	2	5	6	4
2	1	3	4	1	7	6	1	3	8	1	7	2	9	3	4	9	7	6	9	3	8	9	7
9	8	2	7	4	2	5	8	2	3	4	2	9	8	1	7	4	1	5	8	1	3	4	1
6	5	7	6	3	9	6	9	3	6	7	5	6	5	7	6	3	9	6	9	3	6	7	5
3	1	4	5	1	8	7	1	4	9	1	8	3	2	4	5	2	8	7	2	4	9	2	8
5	2	9	7	6	9	1	2	9	3	6	9	5	2	8	7	6	8	1	2	8	3	6	8
4	1	7	4	3	5	4	5	3	4	7	1	4	1	7	4	3	5	4	5	3	4	7	1
3	8	6	1	8	2	7	8	6	5	8	2	3	9	6	1	9	2	7	9	6	5	9	2
6	3	9	8	7	9	2	3	9	4	7	9	6	3	1	8	7	1	2	3	1	4	7	1
5	2	8	5	4	6	5	6	4	5	8	2	5	2	8	5	4	6	5	6	4	5	8	2
4	1	7	2	1	3	8	1	7	6	1	3	4	9	7	2	9	3	8	9	7	6	9	3
7	4	2	9	8	2	3	4	2	5	8	2	7	4	1	9	8	1	3	4	1	5	8	1
6	3	9	6	5	7	6	7	5	6	9	3	6	3	9	6	5	7	6	7	5	6	9	3
5	1	8	3	1	4	9	1	8	7	1	4	5	2	8	3	2	4	9	2	8	7	2	4

Maps producing  $\text{HG}_{rc}^{-1}$

4	3	5	4	1	7	4	7	1	4	7	1	4	5	3
7	6	9	5	2	9	3	6	9	1	2	9	5	2	8
1	8	2	3	8	6	5	8	2	7	8	6	1	9	2
5	4	6	5	2	8	5	8	2	5	6	4	5	2	8
8	7	9	6	3	9	4	7	9	2	3	9	8	7	1
2	1	3	4	1	7	6	1	3	8	1	7	2	9	3
6	5	7	6	3	9	6	9	3	6	7	5	6	9	3
9	8	2	7	4	2	5	8	2	3	4	2	7	4	1
3	1	4	5	1	8	7	1	4	9	1	8	3	2	4
4	1	7	4	3	5	4	5	3	4	7	1	6	3	9
5	2	9	7	6	9	1	2	9	3	6	9	5	2	8
3	8	6	1	8	2	7	8	6	5	8	2	3	9	6
5	2	8	5	4	6	5	6	4	5	8	2	5	4	6
6	3	9	8	7	9	2	3	9	4	7	9	6	3	1
4	1	7	2	1	3	8	1	7	6	1	3	4	9	7
6	3	9	6	5	7	6	7	5	6	9	3	6	5	7
7	4	2	9	8	2	3	4	2	5	8	2	7	4	1
5	1	8	1	8	2	9	1	8	7	1	4	3	4	1

# Bibliography

- [BR38] G. de B. Robinson. “On the Representations of the Symmetric Group”. In: *American Journal of Mathematics* 60 (1938), p. 745. URL: <https://api.semanticscholar.org/CorpusID:124877687>.
- [Sch61a] Craige E. Schensted. “Longest Increasing and Decreasing Subsequences”. In: *Canadian Journal of Mathematics* 13 (1961), pp. 179 –191. URL: <https://api.semanticscholar.org/CorpusID:121435038>.
- [Sch61b] Craige E. Schensted. “Longest Increasing and Decreasing Subsequences”. In: *Canadian Journal of Mathematics* 13 (1961), pp. 179 –191. URL: <https://api.semanticscholar.org/CorpusID:121435038>.
- [Knu70] Donald Ervin Knuth. “PERMUTATIONS, MATRICES, AND GENERALIZED YOUNG TABLEAUX”. In: *Pacific Journal of Mathematics* 34 (1970), pp. 709–727. URL: <https://api.semanticscholar.org/CorpusID:56223310>.
- [Sta71] Richard P. Stanley. “Theory and Application of Plane Partitions. Part 2”. In: *Studies in Applied Mathematics* 50 (1971), pp. 259–279. URL: <https://api.semanticscholar.org/CorpusID:126649639>.
- [Gre74] Curtis Greene. “An extension of Schensted’s theorem”. In: *Advances in Mathematics* 14.2 (1974), pp. 254–265. ISSN: 0001-8708. DOI: [https://doi.org/10.1016/0001-8708\(74\)90031-0](https://doi.org/10.1016/0001-8708(74)90031-0). URL: <https://www.sciencedirect.com/science/article/pii/0001870874900310>.
- [HG76] A. P. Hillman and R. M. Grassl. “Reverse Plane Partitions and Tableau Hook Numbers”. In: *J. Comb. Theory A* 21 (1976), pp. 216–221. URL: <https://api.semanticscholar.org/CorpusID:30944428>.
- [And84] George E. Andrews. *The Theory of Partitions*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1984.
- [Kac90] Victor G. Kac. *Infinite-Dimensional Lie Algebras*. 3rd ed. Cambridge University Press, 1990.

- [Bre99] David M. Bressoud. *Proofs and Confirmations: The Story of the Alternating-Sign Matrix Conjecture*. Spectrum. Cambridge University Press, 1999.
- [Oko00] Andrei Okounkov. *Infinite wedge and random partitions*. 2000. arXiv: math/9907127 [math.RT]. URL: <https://arxiv.org/abs/math/9907127>.
- [OR01] Andrei Okounkov and Nikolai Reshetikhin. “Correlation function of Schur process with application to local geometry of a random 3-dimensional Young diagram”. In: *Journal of the American Mathematical Society* 16 (2001), pp. 581–603. URL: <https://api.semanticscholar.org/CorpusID:11575202>.
- [Pak01] Igor Pak. “Hook Length Formula and Geometric Combinatorics”. In: *Séminaire Lotharingien de Combinatoire* 46 (2001). eprint: <https://www.mat.univie.ac.at/~slc/wpapers/s46pak.html>. URL: <https://www.mat.univie.ac.at/~slc/wpapers/s46pak.html>.
- [Ken03] Richard W. Kenyon. “An introduction to the dimer model”. In: *arXiv: Combinatorics* (2003). URL: <https://api.semanticscholar.org/CorpusID:3083216>.
- [Mau+03] Davesh Maulik et al. “Gromov–Witten theory and Donaldson–Thomas theory, I”. In: *Compositio Mathematica* 142 (2003), pp. 1263–1285. URL: <https://api.semanticscholar.org/CorpusID:5760317>.
- [Oko03] Andrei Okounkov. “Symmetric Functions and Random Partitions”. In: *arXiv: Combinatorics* (2003), pp. 223–252. URL: <https://api.semanticscholar.org/CorpusID:15558365>.
- [ORV03] Andrei Okounkov, Nicolai Reshetikhin, and Cumrun Vafa. “Quantum Calabi–Yau and Classical Crystals”. In: *Progr. Math.* 244 (Oct. 2003). DOI: 10.1007/0-8176-4467-9\_16.
- [OR05] Andrei Okounkov and Nicolai Reshetikhin. “Random Skew Plane Partitions and the Pearcey Process”. In: *Communications in Mathematical Physics* 269 (Apr. 2005). DOI: 10.1007/s00220-006-0128-8.
- [ORV06] Andrei Okounkov, Nicolai Reshetikhin, and Cumrun Vafa. “Quantum Calabi–Yau and Classical Crystals”. In: *The Unity of Mathematics: In Honor of the Ninetieth Birthday of I.M. Gelfand*. Boston, MA: Birkhäuser Boston, 2006, pp. 597–618. ISBN: 978-0-8176-4467-3. DOI: 10.1007/0-8176-4467-9\_16. URL: [https://doi.org/10.1007/0-8176-4467-9\\_16](https://doi.org/10.1007/0-8176-4467-9_16).
- [PT09] Rahul Pandharipande and Richard P Thomas. “The 3-fold vertex via stable pairs”. In: *Geometry & Topology* 13.4 (2009), pp. 1835–1876. DOI: 10.2140/gt.2009.13.1835. URL: <https://doi.org/10.2140/gt.2009.13.1835>.

- [BCY10] J. M. Bryan, Charles Cadman, and Benjamin Young. “The Orbifold Topological Vertex”. In: *arXiv: Algebraic Geometry* (2010). URL: <https://api.semanticscholar.org/CorpusID:50069951>.
- [Sta11] Richard P. Stanley. *Enumerative Combinatorics: Volume 1*. Cambridge University Press, 2011. URL: <https://api.semanticscholar.org/CorpusID:198489053>.
- [Bet+14] Daniela Betea et al. “Perfect sampling algorithm for Schur processes”. In: *ArXiv abs/1407.3764* (2014). URL: <https://api.semanticscholar.org/CorpusID:2892782>.
- [BCC14] Jérémie Bouttier, Guillaume Chapuy, and Sylvie Corteel. “From Aztec diamonds to pyramids: steep tilings”. In: *arXiv: Combinatorics* (2014). URL: <https://api.semanticscholar.org/CorpusID:119164929>.
- [Hop14] Samuel Hopkins. *RSK via local transformations*. Accessed 2024-06-22. 2014. URL: <https://www.samuelhopkins.com/docs/rsk.pdf>.
- [Bou+15] Cédric Boutillier et al. “Dimers on Rail Yard Graphs”. In: *arXiv: Mathematical Physics* (2015). URL: <https://api.semanticscholar.org/CorpusID:54520004>.
- [GP17] Alexander Garver and Rebecca Patrias. “Greene-Kleitman Invariants for Sulzgruber Insertion”. In: *Electron. J. Comb.* 26 (2017), p. 3. URL: <https://api.semanticscholar.org/CorpusID:119313391>.
- [Sul17] Robin Sulzgruber. “Inserting rim-hooks into reverse plane partitions”. In: *Journal of Combinatorics* (2017). URL: <https://api.semanticscholar.org/CorpusID:119621454>.
- [Def+22] Colin Defant et al. “Tilings of benzels via the abacus bijection”. In: *Combinatorial Theory* (2022). URL: <https://api.semanticscholar.org/CorpusID:252531638>.
- [JWY22] Helen Jenne, Gautam Webb, and Benjamin Young. *Double-dimer condensation and the PT-DT correspondence*. 2022. arXiv: 2109.11773 [math.CO]. URL: <https://arxiv.org/abs/2109.11773>.
- [Sta23] Richard Stanley. *Enumerative Combinatorics*. Vol. 2. Cambridge University Press, 2023.
- [GO25] LLC Gurobi Optimization. *Gurobi Python Interface*. 2025. URL: <https://pypi.org/project/gurobipy/>.