# eScholarship
## Combinatorial Theory

**Title**
A lattice model for super LLT polynomials

**Permalink**
https://escholarship.org/uc/item/24s1x99c

**Journal**
Combinatorial Theory, 3(2)

**ISSN**
2766-1334

**Authors**
Curran, Michael J.
Frechette, Claire
Yost-Wolff, Calvin
et al.

**Publication Date**
2023

**DOI**
10.5070/C63261979

**Supplemental Material**
https://escholarship.org/uc/item/24s1x99c#supplemental

Peer reviewed

# A LATTICE MODEL FOR SUPER LLT POLYNOMIALS

Michael J. Curran[*1], Claire Frechette[*†2], Calvin Yost-Wolff[*3], Sylvester W. Zhang[*4], and Valerie Zhang[*5]

[1]*Mathematical Institute, University of Oxford, Oxford, OX2 6GG, United Kingdom*
*michael.curran@ maths.ox.ac.uk*
[2]*School of Mathematics, Boston College, Chestnut Hill, MA 02467, United States*
*frechecl@bc.edu*
[3]*Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, United States*
*calvinyw@umich.edu*
[4]*School of Mathematics, University of Minnesota, Minneapolis, MN 55455, United States*
*swzhang@umn.edu*
[5]*Department of Mathematics, Harvard University, Cambridge, MA 02138, United States*
*vzhang@college.harvard.edu*

**Abstract**. We introduce a solvable lattice model for supersymmetric LLT polynomials, also known as *super LLT polynomials*, based upon particle interactions in super $n$-ribbon tableaux. Using related Heisenberg operators on a Fock space, we prove Cauchy and Pieri identities for super LLT polynomials, simultaneously generalizing the Cauchy, dual Cauchy, and Pieri identities for LLT polynomials. Lastly, we construct a solvable semi-infinite Cauchy lattice model with a surprising Yang–Baxter equation and examine its connections to the Pieri and Cauchy identities.

**Keywords.** Lattice models, super LLT polynomials

**Mathematics Subject Classifications.** 05E05, 82B20, 05E10

## 1. Introduction

LLT polynomials, also known as *ribbon functions*, are a $q$-analog of products of Schur functions and were first introduced by Lascoux, Leclerc, and Thibon in [LLT97] as a family of polynomials realized as generating functions over $n$-ribbon tableaux. Like many other families of functions which originated over tableaux, LLT polynomials have since been shown to satisfy certain nice properties:

---

- they are symmetric,

- they may be written as operators on a Fock space representation, in this case that of the quantum group $U_q(\mathfrak{sl}_n)$,

- they satisfy Cauchy and dual Cauchy identities

- they satisfy Pieri identities and branching rules, and

- they may be realized as the partition function of a lattice model.

The Cauchy and Pieri identities for LLT polynomials were first proved by Thomas Lam in [Lam05], in which Lam also constructed a supersymmetric analogue of LLT polynomials, called *super LLT polynomials*, defined as a generating function over super $n$-ribbon tableaux. By adding an extra set of parameters, super LLT polynomials enable us to track vertical and horizontal information about a tableaux at the same time, corresponding to considering two different types of operators on the Fock space. In this paper, we prove a general Cauchy/dual Cauchy identity for the super LLT polynomials, making a supersymmetric analogue of the approach used in [Lam06] to prove general Cauchy identities for symmetric polynomials using Heisenberg algebras. In fact, we find that the Cauchy identity for super LLT polynomials specializes to both the Cauchy and dual Cauchy identities for LLT polynomials. This remarkable property comes from the fact that the super LLT polynomials may be specialized to LLT polynomials in two separate ways, corresponding to the relationship between a partition and its conjugate, or equivalently to the duality between horizontal and vertical ribbon strips. Consequently, a dual Cauchy identity for super LLT polynomials is merely a rephrasing of the Cauchy identity. Our Cauchy identity also generalizes the Cauchy identity for metaplectic symmetric functions proven by Brubaker, Buciumas, Bump, and Gustafsson in [BBBG20] and the (dual) Cauchy identity for supersymmetric Schur polynomials; see Chapter 5 and Appendix A of [CW12] for a convenient source. We also use a similar approach to prove Pieri rules and branching formulae for super LLT polynomials, generalizing those for LLT polynomials proven by Lam in [Lam06].

A lattice model for LLT polynomials was first developed by Curran, Yost-Wolff, Zhang, and Zhang [CYWZZ19], using the definition over $n$-ribbon tableaux, as a project co-advised by the second author at the 2019 Algebra and Combinatorics REU at the University of Minnesota. This model was proven solvable for $n = 1, 2, 3$, and was conjectured to be solvable for all $n$. Independently, Corteel, Gitlin, Keating, Meza [CGKM22] developed a lattice model for coinversion LLT polynomials using an alternate definition as generating polynomials over tuples of skew tableaux. These tuples are related to ribbon tableaux by a weight-preserving bijection developed by Stanton and White [SW85]. Since the coinversion LLT polynomials give the LLT polynomials up to a correction factor of $q$ depending only on the shape $\lambda/\mu$, these lattice models are equivalent under this bijection up to that factor of $q$, but the underlying structures of each reveal different properties of the polynomials. Later, Aggarwal, Borodin and Wheeler [ABW21] developed a general lattice model that specializes to both of the above models, which they use to reprove many interesting combinatorial identities satisfied by these and similar families of symmetric polynomials.

In this paper, we generalize the model in [CYWZZ19] to a novel lattice model whose partition function produces the super LLT polynomials. We then prove that this supersymmetric model

is solvable, thus covering the solvability conjecture of [CYWZZ19], and use this solvability to provide a lattice model proof of certain symmetry properties of the super LLT polynomials. Recently, Gitlin and Keating [GK21] have independently generalized the coinversion LLT model of [CGKM22] to produce coinversion supersymmetric LLT polynomials; see section 5 of [GK21] for a discussion of the relationship between our supersymmetric models. Similarly, pieces of the various models from Aggarwal, Borodin, and Wheeler [ABW21] can also be combined to give superLLT models; see Section 4 of their sequel paper with Petrov [ABPW21] for an analogous construction for supersymmetric Schur polynomials.

In Section 2, we recall the details of the definition of super LLT polynomials as a generating function over $n$-ribbon tableaux. In Section 3, we reformulate these polynomials in terms of operators generated from a Heisenberg algebra acting on a Fock space. In Section 4, we prove Cauchy and Pieri identities for super LLT polynomials using these operators. Changing gears, Section 5 introduces the (super)ribbon lattice model and proves that its partition function gives the super LLT polynomial, thereby realizing the Heisenberg operators in lattice form. We then show in Section 6 that this model is solvable for all $n$ and use this solvability to reprove interesting symmetry relations on the super LLT polynomials. Section 7 uses the lattice model to prove a branching rule for super LLT polynomials. Finally, Section 8 introduces a semi-infinite Cauchy model built from the ribbon model and Section 9 uses the Cauchy model to provide lattice model proofs of the Pieri and Cauchy identities.

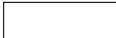## 2. Background on tableaux and symmetric functions

Like many other interesting symmetric functions, super LLT polynomials arise as generating functions over a specific class of tableaux.

A *partition* $\lambda$ is a decreasing sequence $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_r)$ of nonnegative integers. The *length*, or number of parts, of $\lambda$ is denoted $\ell(\lambda)$ and the *size* of $\lambda$ by $|\lambda| = \lambda_1 + \lambda_2 + \ldots + \lambda_r$. We will occasionally use the notation $\lambda \vdash n$ to say $\lambda$ partitions $n$. A *composition* $\alpha = (\alpha_1, \ldots, \alpha_r)$ is a non-ordered list of nonnegative integers; it is important not to confuse these with partitions. We will also define $m_k(\lambda)$ as the number of parts $\lambda_i = k$ and set $\lambda'$ as the conjugate partition of $\lambda$.

A partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_r)$ can be visualized by its *Young diagram*, which consists of horizontal boxes arranged in left-justified rows, where the $i^{\text{th}}$ row contains $\lambda_i$ boxes; abusing notation, we will frequently conflate a partition with its Young diagram. If $\mu \subset \lambda$ for two partitions $\lambda, \mu$, that is if $\mu_i \leqslant \lambda_i$ for all $i$, then $\lambda/\mu$ is a *skew shape* or *skew partition* with size $|\lambda/\mu| = |\lambda| - |\mu|$. We will identify a partition $\lambda$ with the skew shape $\lambda/\varnothing$, and will also identify partitions that differ only by a sequence of parts $\lambda_i = 0$ when it should not cause any confusion.

Fillings of the Young diagram according to a given alphabet are called *tableaux*: there are many different kinds of tableaux, each defined by different rules for how one fills the shape. In this paper, we focus on *ribbon tableaux*, first defined by Stanton and White in [SW85]. Let $n$ be a fixed positive integer. An $n$-*ribbon* $R$ is a skew shape $\lambda/\mu$ containing $n$ boxes that is connected and contains no 2 by 2 squares. The *spin* of a ribbon $R$ is defined to be $\text{spin}(R) := \text{height}(R) - 1$.

**Example 2.1.** Of the six arrangements possible of 3 boxes, four of them are 3-ribbons, which we see below labelled by their spin. The last two are not 3-ribbons because they cannot be realized as skew shapes $\lambda/\mu$.



Given a skew partition $\lambda/\mu$, a tiling of $\lambda/\mu$ by $n$-ribbons is called a *n-horizontal strip* if the top-right-most square of each ribbon touches the northern boundary of $\lambda/\mu$ (Figures 2.1 and 2.2). A tiling is called a *n-vertical strip* if the bottom-left-most square of each ribbon lies along the western boundary of $\lambda/\mu$. Equivalently, a vertical strip is a collection of ribbons that, when collectively flipped over the $y = -x$ antidiagonal, produces a $n$-horizontal strip.
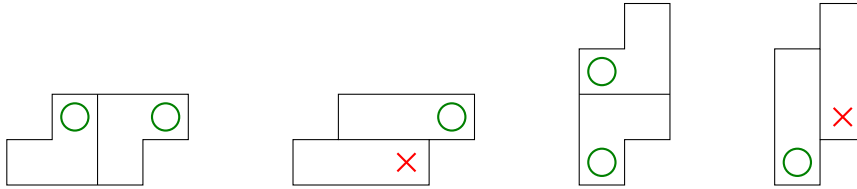


Figure 2.1: From left to right: a 3-horizontal strip, a non-example of a horizontal strip, a 3-vertical strip, and a non-example of a vertical strip.
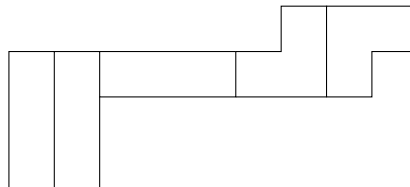


Figure 2.2: A horizontal 3-ribbon strip of shape $(9, 8, 2, 2)/(6)$.

A tiling of $\lambda/\mu$ by $n$-ribbons labelled with positive integers is called a *ribbon tableau*. We often fix an alphabet $A = \{1, \ldots, \ell\}$; then, given such a tableau, we can define a sequence of partitions $\mu = \lambda^0 \subset \lambda^1 \subset \cdots \subset \lambda^\ell = \lambda$, where $\lambda_i$ consists of the subshape of the tableau with labels less than or equal to $i$.

**Definition 2.2.** A *semistandard n-ribbon tableau* (SSRT) of skew shape $\lambda/\mu$ for the alphabet $A = \{1, \ldots, \ell\}$ is a tiling of $\lambda/\mu$ with $n$-ribbons such that the induced sequence $\mu = \lambda^0 \subset \lambda^1 \subset \cdots \subset \lambda^\ell = \lambda$ has the property that for each $i$ the skew shape $\lambda_i/\lambda_{i-1}$ is a horizontal $n$-ribbon strip. We define the *weight* of such a tableau to be the composition $\mathrm{wt}(T)$ such that

$$\mathrm{wt}(T)_i = \#\{n\text{-ribbons labelled } i \text{ in } T\}$$

and we define the *spin* of $T$ to be the sum of the spins of the $n$-ribbon tiles of $T$. The set of all semistandard $n$-ribbon tableaux is denoted by $\mathrm{SSRT}_n(\lambda)$. Note that is not always possible to tile a given partition $\lambda$ with $n$-ribbons, and we will restrict our attention to partitions or skew shapes that are tileable by $n$-ribbons.

We can now define the polynomials formerly known as *n-ribbon Schur functions*, first introduced by Lascoux, Leclerc, and Thibon in [LLT97].

**Definition 2.3.** Let $n \geqslant 1$ be fixed and $\lambda/\mu$ a skew shape tileable by $n$-ribbons for $A = \{1, \ldots, \ell\}$. Then for $X = \{x_1, \ldots, x_\ell\}$, the *(n-)LLT polynomial of shape* $\lambda/\mu$ is defined as the generating function

$$\mathcal{G}_{\lambda/\mu}^{(n)}(X; q) = \sum_{T \in \mathrm{SSRT}_n(\lambda/\mu)} q^{\mathrm{spin}(T)} x^{\mathrm{wt}(T)}. \tag{2.1}$$
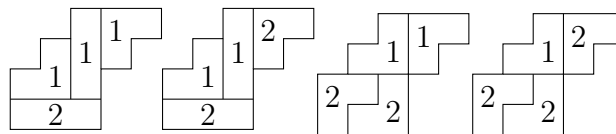
**Example 2.4.** For example, suppose that $\lambda = (3,3)$, $A = \{1,2\}$, and $n = 2$. The only possible 2-ribbon tableaux with shape $\lambda$ and labels in $A$ are those depicted below.
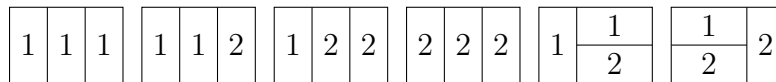


Then,

$$\mathcal{G}_{(3,3)}^{(2)}(x_1, x_2; q) = q^3(x_1^3 + x_1^2 x_2 + x_1 x_2^2 + x_2^3) + q(x_1^2 x_2 + x_1 x_2^2).$$

**Example 2.5.** For a more complicated example, consider $\mathcal{G}_{\lambda/\mu}^{(3)}(X; q)$ where $\lambda = (5, 4, 3, 3)$, $\mu = (2, 1)$, and $A = \{1, 2\}$. We then consider tableaux of shape $\lambda/\mu$ filled with 3-ribbons.



So $\mathcal{G}_{(5,4,3,3)/(2,1)}(x_1, x_2; q) = q^4 (x_1^3 x_2 + 2x_1^2 x_2^2 + x_1 x_2^2)$. In contrast, $\mathcal{G}_{(5,4,3,3)/(2,1)}(x_1; q) = 0$, since filling this entire skew shape with ribbons labelled 1 will violate the requirement that $\lambda^1/\mu$ is a horizontal ribbon strip. In the other direction, the reader may verify that there will be many more than 4 terms in $\mathcal{G}_{(5,4,3,3)/(2,1)}(x_1, x_2, x_3; q)$ or $\mathcal{G}_{(5,4,3,3)/(2,1)}(x_1, x_2, x_3, x_4; q)$.

We will generally write $\mathcal{G}_{\lambda/\mu}^{(n)}(X; q) = \mathcal{G}_{\lambda/\mu}(X; q)$ when $n$ is understood to be fixed. Note: the LLT polynomials are $q$-analogues of the Schur functions in the sense that $\mathcal{G}_{\lambda/\mu}(X; 1)$ is equal to a product of $n$-Schur functions, as proven in [LLT97].

The central object of this paper is the supersymmetric generalization of the LLT polynomials, known as *super LLT polynomials*. As with LLT polynomials, these polynomials may be expressed as a generating function over tableaux, in this case *super ribbon tableaux*, which were first defined in Lam in [Lam05].

**Definition 2.6.** Choose a total order $\prec$ on two ordered alphabets $A, A'$. Let a *super $n$-ribbon tableau* of shape $\lambda/\mu$ be a tiling by $n$-ribbons labeled in the alphabets $A$ and $A'$ such that the ribbon shapes labeled by $a \in A$ form horizontal ribbon strips and those labeled by $b \in A'$ form vertical ribbon strips. Furthermore, we require that the skew shapes generated by this labelling respect the total order on $A$ and $A'$. That is to say, the shape formed by removing the ribbons $\succ i$ gives a skew shape $\lambda_{\prec i}/\mu$ for every $i \in A \cup A'$. We denote the set of super $n$-ribbon tableaux of shape $\lambda/\mu$ by ${}^{s}RT_n(\lambda/\mu)$.

In this paper, we will consistently choose the first alphabet to be a subset of our usual alphabet of positive integers, so $A = \{1, 2, 3, \ldots, r\}$ under the usual ordering, and set $A' = \{1', 2', \ldots, s'\}$ under a similar ordering. The total ordering between the two will vary, although two common choices we will use in examples will be $1 \prec 1' \prec 2 \prec 2' \prec \cdots$ and $1 \prec 2 \prec \cdots \prec r \prec 1' \prec 2' \prec \cdots \prec s'$.

**Definition 2.7.** Fix $n \geqslant 1$ and let $\lambda/\mu$ be a shape tileable by $n$-ribbons. The *super LLT polynomial*, or *super ribbon function*, $\mathcal{G}^{(n)}_{\lambda/\mu}(X/Y; q)$ is the generating function

$$\mathcal{G}^{(n)}_{\lambda/\mu}(X/Y; q) = \sum_{T \in {}^{s}RT_n(\lambda/\mu)} q^{\mathrm{spin}(T)} x^{\mathrm{wt}(T)}(-y)^{\mathrm{wt}'(T)}$$

where $\mathrm{wt}(T), \mathrm{wt}'(T)$ are the weights in the alphabets $A, A'$ respectively.

**Example 2.8.** Suppose that $\lambda = (3, 3), \mu = \varnothing$ and $n = 2$. Let $A = \{1, 2\}$ and $A' = \{1'\}$ under the total order $1 < 1' < 2$. The super ribbon tableaux in ${}^{s}RT(\lambda/\mu)$ for this ordering, which include those from Example 2.4, are

| 1 | 1 | 1 |
| 1 | 1 | 2 |
| 1 | 2 | 2 |
| 2 | 2 | 2 |
| 1 | 1 | 1' |
| 1 | 1' | 2 |
| 1' | 2 | 2 |

We then compute that

$$\mathcal{G}^{(2)}_{(3,3)}(X/Y; q) = q^3(x_1^3 + x_1^2 x_2 + x_1 x_2^2 + x_2^3 - x_1^2 y_1 - x_1 x_2 y_1 - x_2^2 y_1)$$
$$+ q(x_1^2 x_2 + x_1 x_2^2 - x_1^2 y_1 - 2x_1 x_2 y_1 - x_2^2 y_1 + x_1 y_1^2 + x_2 y_1^2).$$

**Example 2.9.** To continue the second skew shape example, consider $\mathcal{G}^{(3)}_{\lambda/\mu}(X/Y; q)$ where $\lambda = (5, 4, 3, 3), \mu = (2, 1), A = \{1, 2\}$, and $A' = \{1', 2'\}$. We then consider tableaux of shape $\lambda/\mu$ filled with 3-ribbons. For this example, let us use the ordering $1 < 1' < 2 < 2'$.

This polynomial has many terms, so for brevity, we include only the subset that are order 2 in each of $X, Y$ and have spin $q^4$ to show how quickly adding the second alphabet expands the

number of terms. There are 18 such tableaux, which are shown below,
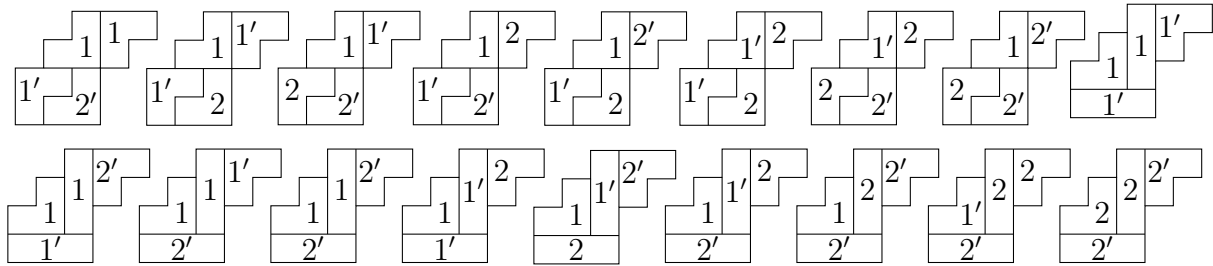


which correspond to the terms

$$q^4 \left( x_1^2 y_1^2 + 2x_1^2 y_1 y_2 + x_1^2 y_2^2 + 2x_1 x_2 y_1^2 + 6x_1 x_2 y_1 y_2 + 2x_1 x_2 y_2^2 + x_2^2 y_1^2 + 2x_2^2 y_1 y_2 + x_2^2 y_2^2 \right).$$

The reader may be interested to note that there are 29 more terms with order 2 in each variable, 3 with spin 1 and 26 with spin $q^2$, which come from fillings of the following shapes:



Observe that in these examples, the super LLT polynomials are symmetric in the $X$ and $Y$ variables separately. Also, while the set of tableaux in each example depends on the total order, it turns out the super LLT polynomial does not: the reader may find it interesting to compute the tableaux for the total order $1 < 2 < 1'$ in Example 2.8 or the total order $1 < 2 < 1' < 2'$ in Example 2.9 and note that they generate t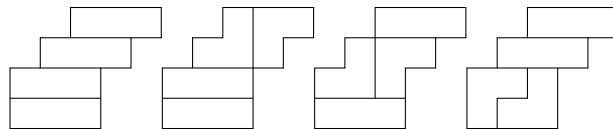he same polynomials. These are in fact both general properties of super LLT polynomials, and were originally proven in [Lam05] using the machinery we will discuss in Section 3. We also reprove them using the solvability of lattice models in Section 5.

**Proposition 2.10** ([Lam05], Prop 30)**.** *The function $\mathcal{G}_{\lambda/\mu}(X/Y; q)$ is symmetric in each of $X$ and $Y$ and does not depend on the total order fixed between $A$ and $A'$.*

It would be natural to suppose that the super LLT polynomials for a skew partition $\lambda/\mu$ and its conjugate skew partition $\lambda'/\mu'$ are closely related, since flipping a super ribbon tableau results in a new super ribbon tableau for the conjugate shape. Combining this with Proposition 2.10 gives us a precise relationship between the two polynomials.

**Proposition 2.11.** *Let $\lambda'$ denote the conjugate partition of $\lambda$. For any $n$ and any skew partition $\lambda/\mu$,*

$$\mathcal{G}_{\lambda/\mu}(X/Y; q) = (-1)^{\frac{|\lambda-\mu|}{n}} q^{(n-1)\frac{|\lambda-\mu|}{n}} \mathcal{G}_{\lambda'/\mu'}(Y/X; q^{-1}).$$

*Proof.* Consider the left hand side as a generating function over super ribbon tableaux of shape $\lambda/\mu$ under the alphabet $A, A'$. Given one of these tableau $T$, flipping it over the anti-diagonal produces a new ribbon tableau $T'$ of shape $\lambda'/\mu'$ with the alphabets flipped, since horizontal

ribbon strips become vertical strips and vice versa. We then compare the weights of $T$ and $T'$. In $T$, ribbons labeled $i \in A$ add a factor of $x_i$ and those labeled $j \in A'$ add a factor of $(-y_j)$. In $T'$, the alphabets swap roles, which amounts to changing the sign of each ribbon. That is, if we let $wt(-)$ be the weight in the horizontal alphabet and $wt'(-)$ that in the vertical alphabet on each side, we have

$$x^{wt(T)}(-y)^{wt'(T)} = (-1)^{\frac{|\lambda - \mu|}{n}}(-x)^{wt'(T')}y^{wt(T')}.$$

For the sake of clarity, we maintain the same total order on our alphabets, regardless of the fact that their roles have changed, since by Proposition 2.10 the polynomials are independent of the chosen ordering. Regarding the power of $q$, each individual ribbon $r$ has the property that its spin satisfies $s(r) + s(r') = n - 1$, so the full ribbon tableaux are related by

$$s(T) + s(T') = (n - 1)\frac{|\lambda - \mu|}{n}. \qquad \qquad \Box$$

## 3. SuperLLT polynomials and Heisenberg algebra operators

Many symmetric polynomials may be represented using representations of a Heisenberg algebra generalized from the fermion side of the classical Boson–Fermion correspondence. In this context, elements of the algebra become operators on a Fock space indexed by partitions $\lambda$, encoding the algebraic framework behind adding or removing boxes in the generating tableaux shapes. Examining the interactions of these operators provides elegant proofs of identities that are difficult or impossible to prove from the tableaux definition. Conversely, families of polynomials generated by such operators may frequently be written as generating functions over tableaux satisfying a similar list of identities (see [Lam06] for a convenient reference). In this section, we introduce the operator definition for super LLT polynomials, which will be used in Section 4 to prove Cauchy and Pieri identities.

In order to properly construct and examine relations between these operators, we first recall a few concepts from symmetric function theory. Let $\Lambda$ denote the ring of symmetric functions over $\mathbb{C}$, or more generally, over a field of characteristic 0. As is standard, let $h_k$ denote the homogeneous symmetric functions, $e_k$ the elementary symmetric functions, and $p_k$ the power sum symmetric functions. Define $p_\lambda := p_{\lambda_1}p_{\lambda_2} \cdots p_{\lambda_r}$ for $\lambda = (\lambda_1, \lambda_2 \ldots, \lambda_r)$. Noting that $p_\lambda$ form a basis for $\Lambda$, the expansions for $h_k$ and $e_k$ are

$$h_k = \sum_{\lambda \vdash k} z_\lambda^{-1} p_\lambda \qquad \qquad e_k = (-1)^k \sum_{\lambda \vdash k} (-1)^{\ell(\lambda)} z_\lambda^{-1} p_\lambda$$

where $z_\lambda = 1^{m_1(\lambda)} m_1(\lambda)! \cdots n^{m_n(\lambda)} m_n(\lambda)!$.

For the remainder of this section, fix a positive integer $n$. Let the Heisenberg algebra $H = H[a_i]$ be the associative algebra with identity generated by the set $\{B_k : k \in \mathbb{Z} \setminus 0\}$ such that

$$[B_j, B_k] = j \cdot a_k \cdot \delta_{j,-k} \qquad \qquad (3.1)$$

for some $a_k \in K$, where $K$ is a field of characteristic 0. For the purpose of this paper, we will typically specialize to $a_k := \frac{1-q^{2nk}}{1-q^{2k}} = 1 + q^{2k} + \cdots + q^{2k(n-1)} \in \mathbb{C}$. This algebra and its

representations are well studied, first by Lascoux, Leclerc, and Thibon [LLT97] and then by Lam for generic $a_i$ [Lam06]. Here, we only need the original representation used by Lascoux, Leclerc, and Thibon on the Fock space $\boldsymbol{F}$ generated over $\mathbb{C}$ by the symbols $|\lambda\rangle$ for partitions $\lambda$. This space is equipped with an inner product $\langle -, - \rangle$, so for notational convenience, we will simplify $|\lambda\rangle$ to $\lambda$ when it appears within the inner product, which is defined by $\langle \lambda, \mu \rangle := \delta_{\lambda\mu}$. It is useful to note that the actions of $B_k$ and $B_{-k}$ on $\boldsymbol{F}$ are adjoint with respect to this inner product.

Let $B_\lambda := B_{\lambda_1} B_{\lambda_2} \cdots B_{\lambda_{\ell(\lambda)}}$ and define $B_{-\lambda}$ similarly. We define four operators within the Heisenberg algebra:

$$
\begin{aligned}
D_k &:= \sum_{\lambda \vdash k} z_\lambda^{-1} B_\lambda & \widetilde{D}_k &:= (-1)^k \sum_{\lambda \vdash k} (-1)^{\ell(\lambda)} z_\lambda^{-1} B_\lambda \\
U_k &:= \sum_{\lambda \vdash k} z_\lambda^{-1} B_{-\lambda} & \widetilde{U}_k &:= (-1)^k \sum_{\lambda \vdash k} (-1)^{\ell(\lambda)} z_\lambda^{-1} B_{-\lambda}.
\end{aligned}
$$

Note that we have designed $D_k$ (respectively $\widetilde{D}_k$) to relate to $B_\lambda$ in the same way as $h_k$ (respectively $e_k$) does to $p_\lambda$.

**Definition 3.1.** Let $\lambda, \mu$ be partitions. Then, we define the generating functions

$$
F_{\lambda/\mu}(X/Y; q) = \sum_{\alpha,\beta} X^\alpha (-Y)^\beta \langle U_{\alpha_l} U_{\alpha_{l-1}} \cdots U_{\alpha_1} \widetilde{U}_{\beta_m} \widetilde{U}_{\beta_{m-1}} \cdots \widetilde{U}_{\beta_1} \cdot \mu, \lambda \rangle,
$$

$$
G_{\lambda/\mu}(X/Y; q) = \sum_{\alpha,\beta} X^\alpha (-Y)^\beta \langle D_{\alpha_l} D_{\alpha_{l-1}} \cdots D_{\alpha_1} \widetilde{D}_{\beta_m} \widetilde{D}_{\beta_{m-1}} \cdots \widetilde{D}_{\beta_1} \cdot \lambda, \mu \rangle
$$

where the sums run over all compositions $\alpha, \beta$.

It is helpful to think of these as tableaux style definitions: the $U$ operators add $n$-ribbon strips to a tableaux of shape $\mu$: $U_k$ adding horizontal strips and $\widetilde{U}_k$ vertical strips, and similarly, the $D$ operators remove $n$-ribbon strips from a tableaux of shape $\lambda$: $D_k$ horizontal and $\widetilde{D}_k$ vertical. Note: in [Lam05], the operators $D_k$ are denoted $\mathcal{V}_k$. From this viewpoint, we realize that both $F_{\lambda/\mu}$ and $G_{\lambda/\mu}$ recover the super LLT polynomials defined in the previous section. This reformulation allowed Lam to prove the symmetry properties of super LLT polynomials stated in Proposition 2.10, which are much more difficult to see from the tableaux definition. On the other hand, using the operator definition, symmetry in $X$ (respectively $Y$) results from the fact that the $D_\alpha$ (respectively $\widetilde{D}_\alpha$) operators commute with themselves. Likewise, the fact that the total order does not affect the polynomial comes from the fact that the $D_\alpha$ and $\widetilde{D}_\alpha$ operators commute.

We next consider the interaction between the actions of adding and removing ribbon strips of different types. Let $\kappa : \Lambda \to \mathbb{C}$ be the algebra homomorphism defined by $\kappa(p_k) = a_k$.

**Lemma 3.2.** *We have the following identities as elements in $H[a_i]$:*

$$
\begin{aligned}
D_b U_a &= \sum_{j=0}^{m} \kappa(h_j) U_{a-j} D_{b-j} & \widetilde{D}_b \widetilde{U}_a &= \sum_{j=0}^{m} \kappa(h_j) \widetilde{U}_{a-j} \widetilde{D}_{b-j} \\
\widetilde{D}_b U_a &= \sum_{j=0}^{m} \kappa(e_j) U_{a-j} \widetilde{D}_{b-j} & D_b \widetilde{U}_a &= \sum_{j=0}^{m} \kappa(e_j) \widetilde{U}_{a-j} D_{b-j}
\end{aligned}
$$

*where $m = \min\{a, b\}$. Equivalently, setting $U(x) := 1 + \sum_{i \geqslant 0} U_i x^i$ and defining $D(x), \widetilde{U}(x),$ $\widetilde{D}(x)$ similarly, we have the following commutation relations*

$$[D(y), U(x)] = [\widetilde{D}(y), \widetilde{U}(x)] = 1 + \sum_{i \geqslant 0} \kappa(h_i)(xy)^i$$

$$[\widetilde{D}(y), U(x)] = [D(y), \widetilde{U}(x)] = 1 + \sum_{i \geqslant 0} \kappa(e_i)(xy)^i$$

*Proof.* The relation of $D_b$ and $U_a$ is proven as Lemma 8 in [Lam06]. We follow the general thread of that proof for the remaining relations: consider the second desired identity with $\widetilde{D}_b$ and $\widetilde{U}_a$. Expanding via definition, the left hand side becomes

$$\widetilde{D}_b \widetilde{U}_a = (-1)^a (-1)^b \sum_{\substack{\rho \vdash a \\ \pi \vdash b}} (-1)^{\ell(\rho)+\ell(\pi)} z_\rho^{-1} z_\pi^{-1} B_\pi B_{-\rho}$$

Then, apply the commutation relation (3) to swap the order of positively and negatively-signed $B$ operators. Effectively, this process builds a smaller partition $\lambda$ out of parts that are in both $\pi$ and $\rho$; we must account for all the ways to build a given $\lambda$ in this manner. If $\mu = \rho \setminus \lambda$ and $\nu = \pi \setminus \lambda$, the expression above becomes

$$= (-1)^a (-1)^b \sum_{\substack{\rho \vdash a \\ \pi \vdash b}} (-1)^{\ell(\pi)+\ell(\rho)} \sum_{\substack{\lambda \subset \rho \\ \lambda \subset \pi}} z_\mu^{-1} z_\nu^{-1} z_\lambda^{-1} \kappa(p_\lambda) B_{-\mu} B_\nu$$

Reindexing over the possible sizes $j$ of $\lambda$, cancelling matching powers of -1, and using the expansion of $h_j$ in terms of $p_\lambda$, we have

$$= \sum_{j=0}^{m} (-1)^{a-j} (-1)^{b-j} \kappa(h_j) \sum_{\substack{\mu \vdash a-j \\ \nu \vdash b-j}} (-1)^{\ell(\mu)+\ell(\nu)} z_\mu^{-1} z_\nu^{-1} B_{-\mu} B_\nu$$

which by definition is the desired right hand side. The remaining identities follow similarly, except that since we only have one tilde-type operator, the power of $(-1)^{j+\ell(\lambda)}$ doesn't cancel out. Instead, it folds in with the $\kappa(p_\lambda)$ according to the expansion of $e_j$ in terms of $p_\lambda$, to obtain a coefficient of $\kappa(e_j)$ instead of $\kappa(h_j)$. $\qquad\square$

# 4. SuperLLT Cauchy and Pieri identities

Using this operator definition, we can construct generalized Cauchy and Pieri identities for $F_{\lambda/\delta}$ and $G_{\lambda/\delta}$. Note that these theorems will apply to any function that can be achieved through specialization of $a_i$ in the operators, not just to super LLT polynomials, however, we will focus on the implications for super LLT polynomials in this paper. See also Hardt [Har21] or Aggarwal, Borodin, Petrov, and Wheeler [ABPW21] for interesting investigations of this identity in terms of Hamiltonian operators. For instance, our Theorem 4.1 is similar to their Proposition 2.10 and Proposition 3.7, respectively.

**Theorem 4.1.** *Let $\mu, \nu$ be partitions. Then*

$$\sum_\lambda F_{\lambda/\mu}(X/Y)G_{\lambda/\nu}(W/Z) = [*] \cdot \sum_\lambda F_{\nu/\lambda}(X/Y)G_{\mu/\lambda}(W/Z)$$

*where*

$$[*] = \prod_{i,j,k,\ell} [D(w_k), U(x_i)] \cdot [\widetilde{D}(-z_\ell), \widetilde{U}(-y_j)] \cdot [\widetilde{D}(-z_\ell), U(x_i)] \cdot [D(w_k), \widetilde{U}(-y_j)].$$

*Proof.* By definition, we have $F_{\lambda/\mu}(X/Y) = \langle \cdots U(x_2)U(x_1) \cdots \widetilde{U}(-y_2)\widetilde{U}(-y_1) \cdot \mu, \lambda \rangle$ and $G_{\lambda/\mu}(X/Y) = \langle \cdots D(x_2)D(x_1) \cdots \widetilde{D}(-y_2)\widetilde{D}(-y_1) \cdot \lambda, \mu \rangle$. Then using the properties of the inner product and the commutation relation of Lemma 3.2, we have

$$
\begin{aligned}
\sum_\lambda F_{\lambda/\mu}(X/Y)G_{\lambda/\nu}(W/Z) &= \sum_\lambda \langle \cdots U(x_2)U(x_1) \cdots \widetilde{U}(-y_2)\widetilde{U}(-y_1) \cdot \mu, \lambda \rangle \\
&\qquad \cdot \langle \cdots D(w_2)D(w_1) \cdots \widetilde{D}(-z_2)\widetilde{D}(-z_1) \cdot \lambda, \nu \rangle \\
&= \langle \cdots D(w_2)D(w_1) \cdots \widetilde{D}(-z_2)\widetilde{D}(-z_1) \cdots U(x_2)U(x_1) \cdots \widetilde{U}(-y_2)\widetilde{U}(-y_1) \cdot \mu, \nu \rangle \\
&= [*]\langle \cdots U(x_2)U(x_1) \cdots \widetilde{U}(-y_2)\widetilde{U}(-y_1) \cdots D(w_2)D(w_1) \cdots \widetilde{D}(-z_2)\widetilde{D}(-z_1) \cdot \mu, \nu \rangle \\
&= [*] \cdot \sum_\lambda \langle \cdots D(w_2)D(w_1) \cdots \widetilde{D}(-z_2)\widetilde{D}(-z_1) \cdot \mu, \lambda \rangle \\
&\qquad \cdot \langle \cdots U(x_2)U(x_1) \cdots \widetilde{U}(-y_2)\widetilde{U}(-y_1) \cdot \lambda, \nu \rangle \\
&= [*] \cdot \sum_\lambda F_{\nu/\lambda}(X/Y)G_{\mu/\lambda}(W/Z). \qquad \square
\end{aligned}
$$

Specializing $\mu$ and $\nu$ to an $n$-core partition gives a more "standard" Cauchy identity formula with a sum side and a product side.

**Corollary 4.2.** *For $\delta$ an $n$-core,*

$$\sum_{\lambda:\widetilde{\lambda}=\delta} \mathcal{G}_{\lambda/\delta}(X/Y;q)\mathcal{G}_{\lambda/\delta}(W/Z;q) = \prod_{i,j,k,\ell} \prod_{t=0}^{n-1} \frac{(1 - q^{2t}x_i z_\ell)(1 - q^{2t}y_j w_k)}{(1 - q^{2t}x_i w_k)(1 - q^{2t}y_j z_\ell)}.$$

*Proof.* Set $\mu = \nu = \delta$ and consider Theorem 4.1, recalling that for the given Heisenberg parameters $a_k = 1 + q^{2k} + \cdots q^{2k(n-1)}$, both $F_{\lambda/\mu}$ and $G_{\lambda/\mu}$ produce the super LLT polynomial $\mathcal{G}_{\lambda/\mu}(X/Y;q)$. There is only one nonzero term in the right-hand sum, because you cannot take any $n$-ribbon strips away from an $n$-core, so the only term is 1 from $\alpha = \beta = \varnothing$. It remains then to simplify the commutation factor: we have

$$[D(y), U(x)] = \prod_{t=0}^{n-1} \frac{1}{1 - q^{2t}xy} \qquad \text{and} \qquad [\widetilde{D}(y), U(x)] = \prod_{t=0}^{n-1}(1 + q^{2t}xy)$$

so taking the product over the appropriate combinations of signed $x, y, z, w$ variables produces the desired statement. $\square$

*Remark* 4.3. Specializing $Y = vX$ and $Z = vW$ in Corollary 4.2 recovers the Cauchy identity for metaplectic Whittaker functions $\mathcal{M}_\lambda^n(\boldsymbol{z})$ given in [BBBG20], as $\mathcal{M}_{\lambda/\mu}^n(\boldsymbol{z}) = \mathcal{G}_{\lambda/\mu}^n(\boldsymbol{z}/v\boldsymbol{z})$.

In search of a dual Cauchy Identity to match this Cauchy Identity, we revisit the Cauchy and dual Cauchy identities for LLT polynomials: since $\mathcal{G}_{\lambda/\mu}(X/0; q) = \mathcal{G}_{\lambda/\mu}(X; q)$, Corollary 4.2 recovers the Cauchy identity for LLT polynomials:

**Corollary 4.4** (Cauchy identity for LLT). *For an $n$-core $\delta$,*

$$\sum_{\lambda:\widetilde{\lambda}=\delta} \mathcal{G}_{\lambda/\delta}(X/0; q) \cdot \mathcal{G}_{\lambda/\delta}(Y/0; q) = \prod_{i,j} \prod_{t=0}^{n-1} \frac{1}{1 - x_i y_j q^{2t}}.$$

Using the relationship between $\mathcal{G}(X/Y)$ and $\mathcal{G}(Y/X)$ developed in Proposition 2.11, we see that $q^{(n-1)\frac{|\lambda-\mu|}{n}} \mathcal{G}_{\lambda'/\mu'}(X; q^{-1}) = \mathcal{G}_{\lambda/\mu}(0/-X; q)$, so Corollary 4.2 also recovers the dual Cauchy identity for LLT polynomials.

**Corollary 4.5** (Dual Cauchy identity for LLT). *For an $n$-core $\delta$,*

$$\prod_{i,j} \prod_{t=0}^{n-1}(1 + x_i y_j q^{2t}) = \sum_\lambda q^{(n-1)\frac{|\lambda-\mu|}{n}} \mathcal{G}_{\lambda/\mu}(X/0; q) \cdot \mathcal{G}_{\lambda'/\mu'}(Y/0; q^{-1})$$

$$= \sum_\lambda \mathcal{G}_{\lambda/\mu}(X/0; q) \cdot \mathcal{G}_{\lambda/\mu}(0/-Y; q).$$

We may think of this relationship between dual Cauchy and Cauchy identities for the LLT polynomials as the fact that not all semi-standard ribbon tableaux are also semi-standard (vertical) ribbon tableaux, so comparing horizontal to vertical fillings will return only finitely many options for tableaux shapes, but comparing horizontal to horizontal or vertical to vertical have infinitely many partitions that may be admissibly filled. However, since super ribbon tableaux combine the notions of "horizontally" and "vertically" semi-standard, the analogue of a dual Cauchy Identity for super LLT polynomials is in fact the consequence of combining Proposition 2.11 with Corollary 4.2.

**Corollary 4.6** ("Dual" Cauchy Identity for super LLT). *For $\delta$ an $n$-core,*

$$\sum_{\lambda:\widetilde{\lambda}=\delta} q^{(n-1)\frac{|\lambda-\mu|}{n}} \mathcal{G}_{\lambda/\delta}(X/Y; q) \mathcal{G}_{\lambda'/\delta'}(W/Z; q^{-1}) = \prod_{i,j,k,\ell} \prod_{t=0}^{n-1} \frac{(1 + q^{2t} x_i w_k)(1 + q^{2t} y_j z_\ell)}{(1 + q^{2t} x_i z_\ell)(1 + q^{2t} y_j w_k)}.$$

Note that Corollary 4.6 can also be proved directly using an argument similar to the proof of Corollary 4.2, which we leave as an exercise to the reader.

We may apply the same machinery to obtain Pieri and branching rules for superLLT polynomials. Let $\theta : \Lambda \to \Lambda$ be the map defined by sending $\theta(p_k) = a_k p_k$. Then, let

$$\boldsymbol{h}_k[a_i](X) := \theta(h_k(X)) \text{ and } \boldsymbol{e}_k[a_i](X) := \theta(e_k(X)).$$

Equivalently, in plethystic notation, $\boldsymbol{h}_k[a_i](X) = h_k[a_i X]$ and $\boldsymbol{e}_k[a_i](X) = e_k[a_i X]$. The classical Pieri rules for $F_{\lambda/\delta}$ and $G_{\lambda/\delta}$, for $\delta$ an $n$-core partition, are written in terms of $\boldsymbol{h}_k$.

**Theorem 4.7** (Lam [Lam06], Theorem 7). *Let $k \geqslant 1$ and let $\lambda$ be a partition with $n$-core $\delta$. Then $F$ and $G$ satisfy generalized Pieri identities*

$$\boldsymbol{h}_k[a_i](X)\, F_{\lambda/\delta}(X) = \sum_\mu \langle D_k \cdot \mu, \lambda \rangle\, F_{\mu/\delta}(X),$$

$$\boldsymbol{h}_k[a_i](X)\, G_{\lambda/\delta}(X) = \sum_\mu \langle U_k \cdot \lambda, \mu \rangle\, G_{\mu/\delta}(X).$$

For supersymmetric polynomials, the interaction of different types of operators $U_k, D_k$ and $\widetilde{U}_k, \widetilde{D}_k$ results in slightly more complicated Pieri rules.

**Theorem 4.8.** *(Generalized supersymmetric Pieri identities) Let $k \geqslant 1$ and let $\lambda$ be a partition with $n$-core $\delta$. Then, $F$ and $G$ satisfy four generalized Pieri identities*

$$\left( \sum_{\ell+m=k} \boldsymbol{e}_\ell[a_i](Y) \cdot \boldsymbol{h}_m[a_i](X) \right) F_{\lambda/\delta}(X/Y) = \sum_\mu \langle D_k \cdot \mu, \lambda \rangle\, F_{\mu/\delta}(X/Y),$$

$$\left( \sum_{\ell+m=k} \boldsymbol{e}_\ell[a_i](X) \cdot \boldsymbol{h}_m[a_i](Y) \right) F_{\lambda/\delta}(X/Y) = \sum_\mu \langle \widetilde{D}_k \cdot \mu, \lambda \rangle\, F_{\mu/\delta}(X/Y),$$

$$\left( \sum_{\ell+m=k} \boldsymbol{e}_\ell[a_i](Y) \cdot \boldsymbol{h}_m[a_i](X) \right) G_{\lambda/\delta}(X/Y) = \sum_\mu \langle U_k \cdot \lambda, \mu \rangle\, G_{\mu/\delta}(X/Y),$$

$$\left( \sum_{\ell+m=k} \boldsymbol{e}_\ell[a_i](X) \cdot \boldsymbol{h}_m[a_i](Y) \right) G_{\lambda/\delta}(X/Y) = \sum_\mu \langle \widetilde{U}_k \cdot \lambda, \mu \rangle\, G_{\mu/\delta}(X/Y).$$

*Proof.* Let $H(X; t)$ and $E(X; t)$ be generating polynomials over $\boldsymbol{h}$ and $\boldsymbol{e}$. That is,

$$H(X; t) = \sum_{k=0}^\infty \boldsymbol{h}_k[a_i](X) t^k \text{ and } E(X; t) = \sum_{k=0}^\infty \boldsymbol{e}_k[a_i](X) t^k.$$

We will use these to simultaneously prove the Pieri rules for all $k$. To prove the first statement, consider

$$\sum_\mu \langle D(t) \cdot \mu, \lambda \rangle F_{\mu/\delta}(X/Y) = \sum_\mu \langle D(t) \cdot \mu, \lambda \rangle \langle \cdots U(x_2)U(x_1) \cdots \widetilde{U}(-y_2)\widetilde{U}(-y_1) \cdot \delta, \mu \rangle$$

$$= \langle D(t) \cdots U(x_2)U(x_1) \cdots \widetilde{U}(-y_2)\widetilde{U}(-y_1) \cdot \delta, \lambda \rangle$$

$$= [\star] \cdot \langle \cdots U(x_2)U(x_1) \cdots \widetilde{U}(-y_2)\widetilde{U}(-y_1)D(t) \cdot \delta, \lambda \rangle,$$

where $[\star] = \prod_{x_i \in X}[D(t), U(x_i)] \cdot \prod_{y_j \in Y}[D(t), \widetilde{U}(-y_j)]$,

$$= [\star] \cdot \langle \cdots U(x_2)U(x_1) \cdots \widetilde{U}(-y_2)\widetilde{U}(-y_1) \cdot \delta, \lambda \rangle$$

$$= [\star] \cdot F_{\lambda/\delta}(X/Y).$$

where the penultimate line follows from the fact that the $D(t)$ operator removes horizontal strips from a partition, but $\delta$ is an $n$-core, so $D(t) \cdot \delta = \delta$. A straightforward computation gives that $H(X;t) = \prod_{x_i \in X}[D(t), U(x_i)]$ and $E(Y;t) = \prod_{y_j \in Y}[D(t), \widetilde{U}(-y_j)]$. Then, extracting the coefficient of $t^k$ in each side of the equation yields the desired formula. The second equation follows similarly, exchanging $D_k$ and thus $D(t)$ for $\widetilde{D}_k$ and $\widetilde{D}(t)$.

For the third formula, the same method applies, however we note one slight difference after we have commuted the operator $U(t)$ past the operators generating $G$:

$$\sum_\mu \langle U(t) \cdot \lambda, \mu \rangle G_{\mu/\delta}(X/Y) = \sum_\mu \langle U(t) \cdot \lambda, \mu \rangle \langle \cdots D(x_2)D(x_1) \cdots \widetilde{D}(-y_2)\widetilde{D}(-y_1) \cdot \mu, \delta \rangle$$

$$\vdots$$

$$= [\star]\langle U(t) \cdots D(x_2)D(x_1) \cdots \widetilde{D}(-y_2)\widetilde{D}(-y_1)\lambda, \delta \rangle,$$

where $[\star] = \prod_{x_i \in X}[D(t), U(x_i)] \cdot \prod_{y_j \in Y}[D(t), \widetilde{U}(-y_j)]$,

$$= [\star]\langle \cdots D(x_2)D(x_1) \cdots \widetilde{D}(-y_2)\widetilde{D}(-y_1)\lambda, \delta \rangle.$$

Since $\delta$ is again an $n$-core partition and $U(t)$ adds horizontal strips, the only constribution to the $\delta$-th component of $U(t) \cdots D(x_2)D(x_1) \cdots \widetilde{D}(-y_2)\widetilde{D}(-y_1)\lambda$ comes from the constant term $1$ in $U(t)$. Rephrasing in terms of $G$ and taking the coefficient of $t^k$, we then recover the third formula, and the proof of the fourth formula is analogous. $\qquad\square$

Inspired by this proof method, we may also state generalized branching rules for the $F$ and $G$ polynomials.

**Proposition 4.9.** *(Generalized branching rules) Let $\lambda$ be a partition with $n$-core $\delta$. Then, $F$ and $G$ satisfy four generalized branching rules:*

$$F_{\lambda/\delta}((X \cup \{t\})/Y) = \sum_\mu \langle U(t) \cdot \mu, \lambda \rangle \, F_{\mu/\delta}(X/Y),$$

$$F_{\lambda/\delta}(X/(Y \cup \{t\})) = \sum_\mu \langle \widetilde{U}(t) \cdot \mu, \lambda \rangle \, F_{\mu/\delta}(X/Y),$$

$$G_{\lambda/\delta}((X \cup \{t\})/Y) = \sum_\mu \langle D(t) \cdot \lambda, \mu \rangle \, G_{\mu/\delta}(X/Y),$$

$$G_{\lambda/\delta}(X/(Y \cup \{t\})) = \sum_\mu \langle \widetilde{D}(t) \cdot \lambda, \mu \rangle \, G_{\mu/\delta}(X/Y).$$

*Proof.* These identities follow similarly to the Pieri rules, except without eliminating the $t$ parameter. For instance, for the first identity,

$$F_{\lambda/\delta}((X \cup \{t\})/Y) = \langle U(t) \cdots U(x_2)U(x_1) \cdots \widetilde{U}(-y_2)\widetilde{U}(-y_1) \cdot \delta, \lambda \rangle$$

$$= \sum_\mu \langle U(t) \cdot \mu, \lambda \rangle \langle \cdots U(x_2)U(x_1) \cdots \widetilde{U}(-y_2)\widetilde{U}(-y_1) \cdot \delta, \mu \rangle$$

$$= \sum_\mu \langle U(t) \cdot \mu, \lambda \rangle \, F_{\mu/\delta}(X/Y).$$

Recalling that the $U$ and $\widetilde{U}$ operators commute, as do the $D$ and $\widetilde{D}$ operators, the remaining three proofs are analogous. $\qquad\square$

Since both $F$ and $G$ specialize to superLLT polynomials under this choice of $a_i$, we specialize to the following corollary for superLLT polynomials.

**Corollary 4.10.** *Let $\lambda$ be a partition with $n$-core $\delta$. Then the super LLT polynomial satisfies the Pieri rules*

$$E(Y;t)H(X;t)\mathcal{G}_{\lambda/\delta}(X/Y;q) = \sum_{\mu} \mathcal{G}_{\mu/\lambda}(\{t\}/0;q) \cdot \mathcal{G}_{\mu/\delta}(X/Y;q),$$

$$E(X;-t)H(Y;-t)\mathcal{G}_{\lambda/\delta}(X/Y;q) = \sum_{\mu} \mathcal{G}_{\mu/\lambda}(0/\{t\};q) \cdot \mathcal{G}_{\mu/\delta}(X/Y;q),$$

*and the branching formulae*

$$\mathcal{G}_{\lambda/\delta}((X \cup \{t\})/Y;q) = \sum_{\mu} \mathcal{G}_{\lambda/\mu}(\{t\}/0;q) \cdot \mathcal{G}_{\mu/\delta}(X/Y;q),$$

$$\mathcal{G}_{\lambda/\delta}(X/(Y \cup \{t\});q) = \sum_{\mu} \mathcal{G}_{\lambda/\mu}(0/\{t\};q) \cdot \mathcal{G}_{\mu/\delta}(X/Y;q).$$

Specifically, we prefer the generating function version that incorporates the $t$ parameter as this more closely mimics the lattice model constructions we will investigate in later sections. Also, note that in the first identity, the only nonzero terms on the right hand side will be those for which $\mu/\lambda$ is a horizontal strip, as otherwise $\mathcal{G}_{\mu/\lambda}(\{t\}/0;q) = 0$. Similarly, the only nonzero terms on the right hand side in the second identity are those for which $\mu/\lambda$ is a vertical strip. The third and fourth are analogous upon replacing $\mu/\lambda$ with $\lambda/\mu$.

Examining the way we apply these operators to add or remove horizontal and vertical strips, from the view of particle interactions inside the individual ribbon, we can construct a lattice model whose partition function gives the super LLT polynomials. In Section 8, we will then investigate how we can see relationships between the Cauchy and Pieri identities above and a solvable *Cauchy lattice model* built out of this lattice model.

## 5. A lattice model for super LLT polynomials

Many interesting classes of polynomials and special functions can be represented as the partition functions of *solvable* lattice models, i.e. weighted functions on a grid system that satisfy Yang–Baxter equations. In this section, we show that super LLT polynomials appear as the partition functions of a *n-ribbon* lattice model, an $n$-stranded generalization of the 5 vertex lattice model for Schur polynomials.

Throughout this section, fix an integer $n \geqslant 1$. To define a *n-ribbon lattice model*, construct a two-dimensional square grid and place vertices at the intersections. Every vertical edge will be assigned a *label* from the set $\{\wedge, \vee\}$; this label is sometimes also called a "spin," but we will refrain from using that term to avoid confusion with the spin ($q$-power) of a ribbon. Every

horizontal edge will be assigned an $n$-tuple of labels from the set $\{<,>\}$. However, distinguishing this model from merely being a collapsed version of a grid $n$ times taller, we will require a curious condition on these tuple labels: given a vertex $v$, denote the north, south, east and west entries of a vertex $v$ by $v_N, v_S, v_E$ and $v_W$ respectively. Since $v_E$ and $v_W$ are both $n$-tuples, we denote the $i$-th entry by $v_E(i)$ or $v_W(i)$ respectively.

**Definition 5.1.** We say a $n$-ribbon vertex $v$ is *admissible* if it satisfies the following conditions:
  1. The number of arrows pointing inwards equals the number of arrows pointing outwards.
  2. The labels $v_E(i) = v_W(i+1)$ for all $i \in \{1, 2, \ldots, n-1\}$. Note that $v_E(n)$ need not be equal to $v_W(1)$.

We may visualize this condition by expanding the $n$-tuple-labelled horizontal edge of a vertex into $n$ horizontal edges with one distinguished edge to represent that from $v_W(1)$ to $v_E(n)$ (see Figure 5.1). We will call this distinguished edge the *twisted edge*, and every other horizontal edge will be called *straight*. Then we can summarize the second condition concisely by requiring that arrows do not change along any straight (horizontal) edge.
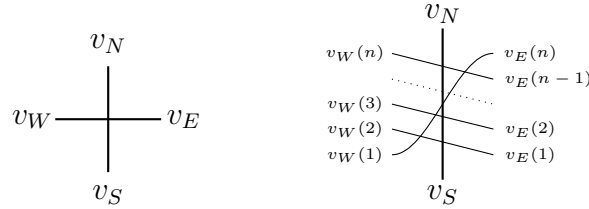
Figure 5.1: Two renderings of a $n$-ribbon vertex, where the one on the left uses tuple labels $v_W = (v_W(1), \cdots, v_W(n))$ and $v_E = (v_E(1) \cdots v_E(n))$ to collapse the strands. In both renderings, $v_N, v_S \in \{\wedge, \vee\}$ and $v_E(i), v_W(i) \in \{<,>\}$. We will tend to use the right one, as it is more diagramatically convenient for our applications.

Note that while we display the twisted edge as "crossing" the straight edges, we think of this vertex as a diagrammatic fusion of the $n$ vertices obtained by crossing every horizontal edge with the vertical edge (as opposed to $2n-1$ crossings seemingly displayed in Figure 5.1). (See Remark 5.6 for a more detailed explanation of this process and its connection to other types of lattice model fusion.)

*Remark* 5.2. The independent coinversion LLT polynomial lattice models of [CGKM22], [ABW21], and [GK21] follow a similar rendering process, with the additional step that their models fuse $n$ consecutive columns together to obtain a vertex with $n$-tuple labels on every edge. After this additional fusion, their weights may be obtained by careful specialization of the $R$-matrix for the quantum group $U_q(\widehat{\mathfrak{sl}}(1|n))$, since each $n$-labelled strand may be associated to an $n$-dimensional evaluation module for this quantum group. (See Section 5 of [ABW21] for more detailed explanation.)

It is difficult to identify a quantum group module corresponding to our vertical edges, which have 2 label options and would thus require a 2-dimensional module, so we cannot obtain the weights in Figures 5.3 through this sort of specialization. However, fusing columns in our

model in the same manner as [CGKM22] and [ABW21] would obtain a lattice model arising from $U_q(\widehat{\mathfrak{sl}}(1|n))$ under a similar specialization. We hope that investigating this distinction in future work will identify a suitable quantum module for the vertical edges in our existing model, as this conundrum has arisen in several other lattice models relating to special functions [BBBG19, BBBG21b, Fre20, Gra17].

We call a set of label conditions on the boundary edges of the grid a *system*, allowing the labels on interior edges to vary. A given assignment of labels for all these interior edges is called a *state* of that system. We say a state is *admissible* each of its vertices is admissible. Under the stranded rendering, we see that an admissible state has chains of arrows travelling $n$ vertices without changing label. That is, that every left (respectively right) arrow occurring at $v_E(n)$ for some vertex $v$ must be followed by $n - 1$ consecutive left (respectively right) arrows as we travel to the right along its strand before it is allowed to change label (see Figure 5.2). It is helpful to think of this condition in terms of a path model for particles: if we consider particles travelling through our lattice model along "up" and "left" arrows (where then edges labelled "down" or "right" have no particle), this condition tells us that particles *must* travel in steps of $n$ vertices at a time. (See Figures 5.4 and 5.5 for examples of admissible states with only a single particle travelling in one row, or Figures 5.6 and 5.7 for examples of admissible states with multiple particles travelling in a single row.)
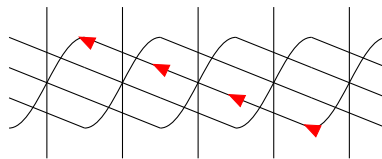


Figure 5.2: The $v_E(4)$ entry of the leftmost vertex is a left arrow, which must be following by 3 left arrows from the right.

**Definition 5.3.** Let $\lambda/\mu$ be a skew partition, $r$ a positive integer, and $A$, $A'$ two ordered alphabets such that $|A| + |A'| = r$ with a total ordering between them that respects their individual orderings. Define the system of boundary conditions $\mathcal{B}_{\lambda/\mu}(X/Y)$ on a $n$-ribbon lattice model, letting $\rho = (r, r - 1, \ldots, 2, 1)$, as:

- there are $r$ rows and $\lambda_1 + r$ columns,

- all edges on the left and right boundaries are labelled $>$,

- numbering the columns left to right starting with 1, edges on the bottom boundary that appear as parts of $\lambda + \rho$ are labelled $\wedge$, and all others are labelled $\vee$, and

- edges on the top boundary appearing as parts of $\mu + \rho$ are labelled $\wedge$, while all others are labelled $\vee$.

- *horizontal* rows labelled $i \in A$ have spectral parameters $x_i$,

- *vertical* rows labelled $i' \in A'$ have spectral parameters $y_{i'}$, and

- reading from top to bottom, rows are labelled in increasing order according to the total order.

We then define a generating function on the lattice model.

**Definition 5.4.** Let $\mathrm{wt}(v)$ denote the Boltzmann weight of a vertex. Define the Boltzmann weight of a state $\mathfrak{s}$ to be the product over the weights of its vertices and the Boltzmann weight of a system, commonly known as the *partition function* of the system, to be the sum over the weights of its states. That is,

$$\mathrm{wt}(\mathfrak{s}) = \prod_{v \in \mathfrak{s}} \mathrm{wt}(v) \qquad \text{and} \qquad Z(B) := \mathrm{wt}(\mathcal{B}) = \sum_{\mathfrak{s} \in \mathcal{B}} \mathrm{wt}(\mathfrak{s}).$$

Any vertex that is not admissible has weight $0$. Given any admissible vertex $v$, its weight will depend on whether it is in a *horizontal* strip row or a *vertical* strip row. We will suggestively label the horizontal strip rows with numbers $i \in A$ and assign them spectral parameters $x_i$, while the vertical strip rows will be labelled with numbers $i' \in A'$ and have parameters $y_{i'}$.

We now define two sets of Boltzmann weights for the admissible vertices, all of which are monomials in $q$ and the spectral parameter $x_i$ or $y_{i'}$ assigned to the row in which a vertex appears. These weights are shown in Figure 5.3. Note that in this table, we label the different types of vertices based on the edges pointing into the vertex from either the column edge or the twisted edge. We may generally think of the twisted edge as controlling the power of the spectral parameter and the straight edges as controlling the power of $q$.

| Label | SW | NS | SE | NW | EW | NE |
|---|---|---|---|---|---|---|
| Vertex | | | | | | |
| $wt_H$ | $q^s$ | $q^s$ | $0$ | $1$ | $q^s x_i$ | $q^s x_i$ |
| $wt_V$ | $q^s$ | $1$ | $-y_{i'}$ | $1$ | $-y_{i'}$ | $0$ |

Figure 5.3: The weights $\mathrm{wt}_H(v)$ of a vertex $v$ in *horizontal strip row* $i \in A$ and $\mathrm{wt}_V(v)$ of a vertex $v$ in a *vertical strip row* $i' \in A'$ for the $n$ super ribbon lattice model, where $s$ is the number of left arrows in the circled area. To match the particle interpretation, "up" and "left" arrows are colored red (contain a particle) and "down" and "right" arrows blue (no particle).

*Remark* 5.5. The two sets of weights may be related by the following standard procedure: in the weights for a horizontal strip row (Figure 5.3), substitute $q^{-1}$ for $q$ and $-y_{i'}$ for $x_i$, then multiply each weight by $q^s$. Lastly, flip each vertical strand over the horizontal diagonal (e.g., SW becomes NW, but NS stays NS).

This duality arises from the fact that flipping a horizontal ribbon strip over the antidiagonal $y = -x$ obtains a vertical ribbon strip: compare the 3-ribbons in Figure 5.4 to each other: the first and third examples trade places, while the second and fourth examples are sent to themselves.

As discussed in Proposition 2.11, if $T$ is an $n$-ribbon and $T'$ its flip, then the spins of $T$ and $T'$ add to $n - 1$, the number of possible strands that can carry a power of $q$ in the $n$-vertex. Similarly, the two vertices that carry a power of $x_i$ for a horizontal ribbon strip are those on which a particle leaves the ribbon (EW and NE), which after this process become (EW and SE), the two vertices on which a particle can leave a ribbon in a vertical ribbon strip. The flip arises from the fact that horizontal edges become vertical edges and vice versa under the flip, so arrows that were red are now blue and vice versa.

*Remark* 5.6. The horizontal strip row weights depicted in Figure 5.3 may be obtained through a graphical "fusion" procedure on the $n$ smaller vertices obtained by taking each horizontal strand crossing the single vertical strand. Given a vertex coming from a "straight" strand, let it have weight $q$ if the straight edge labels point left and weight 1 if the straight edge labels point right. Given a vertex for a twisted horizontal edge, let it have weight 1 if it is type SW, or NS; weight 0 if it is type SE; weight $x_i$ if it is type EW or NE; and weight $q^{-s}$ if it is type $NW$. Overlaying these $n$ smaller vertices to obtain an $n$-ribbon vertex, multiply the weights of the individual pieces together to obtain the weight of the $n$-ribbon vertex. The vertical strip row weights may be obtained through an analogous procedure. Note that this process requires us to consider only $n$ of the $2n - 1$ strand "intersections" visible in the way we have displayed the $n$-ribbon vertex.

These processes are combinatorial fusions in the sense of [BW20, BBBG20, BBBG21a, BBBG21b], which generalize the graphical interpretation of the traditional fusion procedure by Kulish, Reshetikhin, and Sklyanin for tensor products of quantum group modules and their subquotients [KRS81]. (See Appendix B of [BW20] for a description of when combinatorial fusion coincides with traditional fusion.)

**Theorem 5.7.** *Given a skew partition $\lambda/\mu$ and a total order on alphabets $A$, $A'$, we have*

$$Z\left(\mathcal{B}_{\lambda/\mu}(X/Y)\right) = \mathcal{G}^{(n)}_{\lambda/\mu}(X/Y; q).$$

To prove Theorem 5.7, we construct a weight preserving bijection between the set of all super ribbon tableaux and the set of all admissible ribbon lattices with the corresponding boundary conditions. We start with an important lemma about individual ribbons.

**Lemma 5.8.** *There is a weight-preserving bijection between $n$-ribbons and ribbon lattice models with one row and $n + 1$ columns, using either set of weights from Figure 5.3.*

*Proof.* The bijection itself is easy to describe: send the $n$-ribbon of shape $\lambda/\mu$ to the one-row lattice model with boundary conditions $\mathcal{B}_{\lambda/\mu}$. To show that it is weight-preserving is a bit more complicated, and we take this time to develop some machinery that will be useful later.

Consider an $n$-ribbon of shape $\lambda/\mu$. Starting in the bottom right corner, label each vertical (resp. horizontal) edge red (resp. blue) and number their positions increasingly along each of the upper and lower edges of the ribbon. We call this the *edge sequence path* of the ribbon. Note that red edges occur on labels that occur as parts of $\mu + \rho$ on the top boundary and on labels in $\lambda + \rho$ on the bottom boundary of the ribbon, and the largest label will always be $n + 1$. We may therefore think of the bijection heuristically as "stretching" the ribbon straight to lay atop the lattice model.
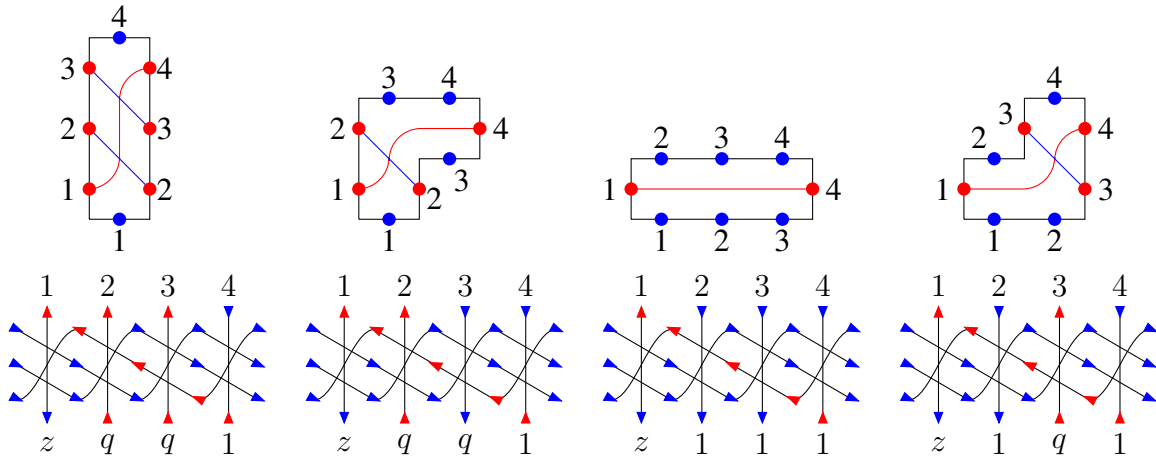
Figure 5.4: Edge sequence paths for the four 3-ribbons given in Example 2.1 and their corresponding lattice states with weights from either weight set in Figure 5.3, where $z = x$ or $-y$.
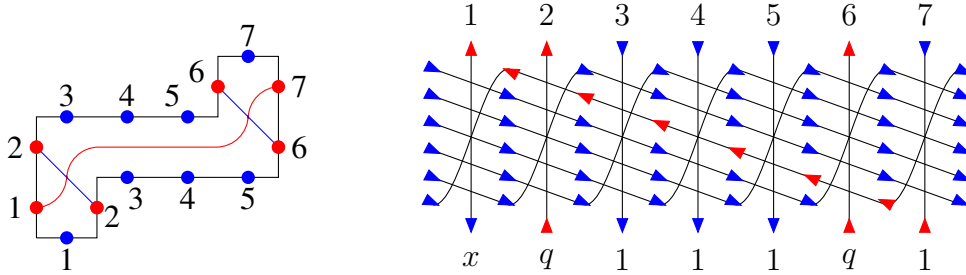


Figure 5.5: The 6-ribbon for $\lambda = (4, 4, 1), \mu = (3, 0, 0)$ and its corresponding lattice model state with weights included.

Claim: there is only one state for this model. Since a ribbon may not contain any 2 by 2 squares, we may assign a particle action to the red edges of the ribbon: along each $n$-ribbon, the red dot in the upper right corner (labelled $n + 1$) will travel to the lower left corner (labelled 1), and all the remaining reds will travel up and to the left while retaining the same label. Thus, aside from column 1 and column $n + 1$, all other columns have the same label on top and on bottom. Coupling this with the fact that the side boundaries are all right arrows, the inner inner labels of the lattice model are fully determined: the vertex in column 1 must be type EW, that in column $n + 1$ must be type NS, and the remainder are either type NW (if column $i$ is blue in the ribbon) or type SW (if column $i$ is red in the ribbon). The remaining labels are determined by the fact that straight edges don't change label across a vertex, so the strand from column 1 to column $n + 1$ is entirely left arrows and all the rest are right arrows.

Vertex types in hand, we return to the weighting. There are no SE or NE vertices in this model, and the only internal left arrows occur on type SW vertices, so the two sets of weights in Figures 5.3 give the same partition function for this model up to choice of the spectral parameter $z$, which we specialize to $x_i$ or $-y_{i'}$ respectively. Since we have a single state, the partition function

is a monomial: the EW vertex gives a power of $z$ and each SW vertex a power of $q$, while the NW and NS vertices don't change the weight, so our lattice model weight is $q^{\#SW}z$.

Recall that a lone ribbon has weight $q^{\text{height} -1}z$, where $z$ is the desired spectral parameter. Referring to Figure 5.5, note that the spin $q^{\text{height} -1}$ counts the number of intersections between particles (i.e. the number of red columns between 1 and $n + 1$), which are precisely the columns that become SW vertices. $\square$

**Proposition 5.9.** *Using the horizontal strip row weights from Figure 5.3, there is a weight-preserving bijection between horizontal $n$-ribbon strips and one-row lattice models with non-zero weight.*

*Proof.* Sends a horizontal ribbon strip of shape $\lambda/\mu$ to the one row model with boundary conditions $B_{\lambda/\mu}$. To show this map is a bijection, we define an assignment of internal arrows that turns a ribbon filling of this strip into an admissible state by "peeling off" successive ribbons. Starting at the rightmost ribbon in the strip, assign left arrows as in Lemma 5.8 for each ribbon, momentarily ignoring the right arrows. That is, a ribbon with edge sequence labels $k, \ldots, k + n$ for some $k$ will assign left arrows to the lattice model edges $v_{k,E}(n), v_{k+1,E}(n-1), \ldots, v_{k+n,W}(1)$, which comprise the straight edge strand connecting vertices $k$ and $k + n$. Once all ribbons have been considered, fill the remaining edges with right arrows. Since no two ribbons start (or end) with the same labels, these choices of left arrow edges will be distinct over all ribbons. Since the path of left arrows assigned by the last ribbon starts with $v_E(n)$ on $v_1$ and that of the first ribbon ends with $v_E(1)$ on the last vertex, this assignment of left arrows will not impede the assignment of straight edge right arrow paths dictated by the left and right boundary conditions.
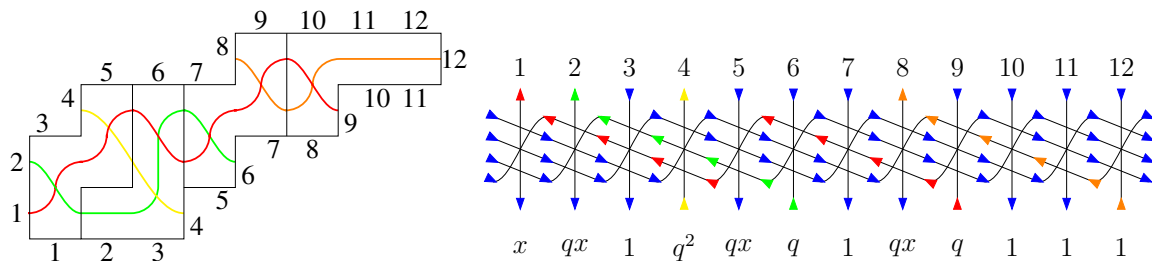


Figure 5.6: The edge sequence path of a horizontal strip, and the corresponding single row lattice state, with colors indicating movements of particles. The weight of this tableaux/state is $q^7x^4$.

To check that this is an admissible lattice state, we need to check that each vertex is one of our admissible types: our assignment scheme ensures that straight edges won't change label (i.e. that $v_E(i) = v_W(i + 1)$ for all vertices $v$), so it suffices to check the twisted edge/vertical edge combinations. As in Lemma 5.8, categorize by the coloring on the edge sequence, extending the edge sequence through each individual ribbon involved. Reading from bottom boundary to top boundary, if a column remains blue (no particle) throughout the entire strip, it generates a NW vertex; if remains red (particle) throughout the entire strip, it gives a SW vertex. If it begins red and ends blue, it makes a NS vertex, whereas if it begins blue and ends red it makes an EW vertex. The final option is for a label to start blue, turn red, and return to blue, which occurs when

a moving path passes through an intermediate ribbon to become another ribbon's moving path (see column 5 of Figures 5.6), which gives a NE vertex. Notice that there are no type SE vertices, so this lattice will have non-zero weight.

We may reverse this entire process to produce a ribbon filling from a lattice model: starting with the rightmost particle entering the lattice, start at the bottom boundary label for that column and step the ribbon left for each column it passes with no other particles on the vertical or twisted edge, and down if it "encounters" another particle. Repeat this process for the remainder of the unfilled horizontal strip. Here it is important that the SE vertex is excluded, because this vertex comes from allowing the bottom-left-most square of a ribbon to touch the upper boundary of another ribbon rather than the bottom of the ribbon strip, which would violate the horizontal ribbon strip condition.

That the map is weight preserving follows naturally from the path interpretation of the ribbon lattices. In Lemma 5.8, the power of $x_i$ of the ribbon was assigned to the vertex where the long path exited the ribbon (type EW). In a ribbon strip, we see that path can exit the ribbon in either an EW vertex (if it is exiting the whole strip) or a NE vertex (if it is continuing into another ribbon), so both of these vertices contribute to the power of $x_i$. On the other hand, the power of $q$ counts the number of intersection of the paths, which is exactly the value of the spin as before, and all vertex types with path intersections (all except type NW) increment these powers of $q$ accordingly.                                                                                    $\square$

**Proposition 5.10.** *Using the vertical strip row weights from Figure 5.3, there is a weight-preserving bijection between vertical $n$-ribbon strips and one-row lattice models with non-zero weight.*

*Proof.* Consider the same bijection as in Proposition 5.9, sending a vertical ribbon strip of shape $\lambda/\mu$ to the one row model with boundary conditions $B_{\lambda/\mu}$ and assigning left arrows according to Lemma 5.8 for each ribbon, then filling the remaining edges with right arrows. Label colorings of the edge sequence that produce NW, SW, NS, and EW vertices remain the same, but we also see SE vertices from labels that start red (particle), become blue (no particle), and return to red. Note that there will be no NE vertices, since these come from allowing the top-right-most square of a ribbon to touch the upper boundary of another ribbon rather than the bottom boundary of the ribbon strip, which is not allowed in a vertical ribbon strip, so the state will have nonzero weight. This also ensures that the reverse map will produce a vertical ribbon strip.

Again, the preservation of weight follows naturally from the path interpretation of the ribbon lattices. The weight of the lattice model obtains a power of $(-y_{i'})$ from each time a travelling particle exits out the top boundary. These exits occur on vertices of type SE and EW, and each of them will come from a single $n$-ribbon in the vertical strip. To match the power of $q$, notice that unlike the horizontal strip lattice, we cannot "bounce" an exiting travelling particle of one ribbon out through another ribbon, so all the intersections contributing to the spin in the vertical ribbon lattice happen on type SW vertices, which are precisely the labels mid-ribbon at which a given ribbon takes a step up, and therefore contribute a single power of $q$ to the weight of that ribbon.                                                                                    $\square$

*Proof of Theorem 5.7.* Consider a super ribbon tableau of shape $\lambda/\mu$ under the total order $\prec$
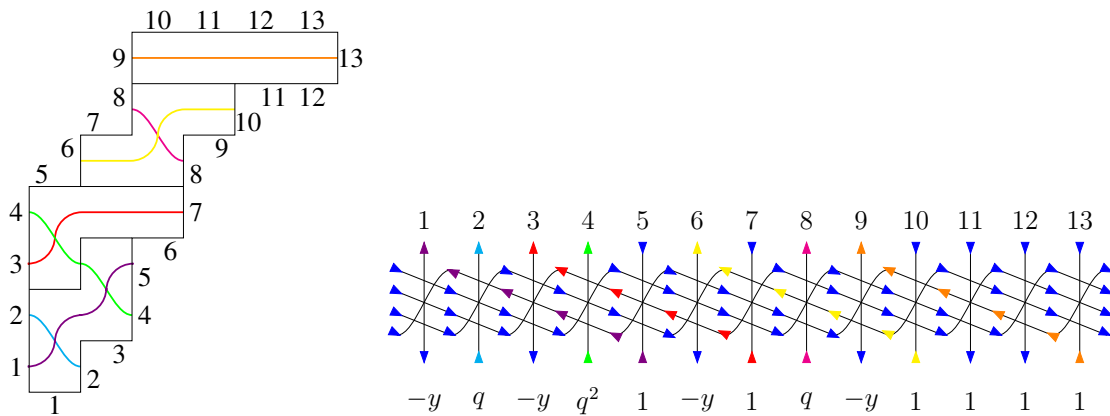
Figure 5.7: The edge sequence path of a vertical strip, and the corresponding single row lattice state, with colors indicating movements of particles. The weight of this tableaux/state is $q^4(-y)^4$.
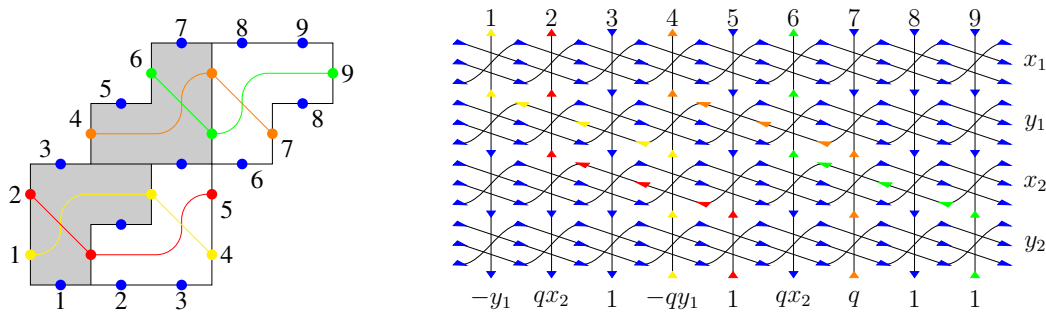


Figure 5.8: The super ribbon tableaux from Example 2.9 with weight $q^4 x_2^2 y_1^2$, where the shaded ribbons are labelled $1' \in A'$ and the blank ribbons $2 \in A$, along with the corresponding lattice model state. Recall that in this example, $A = \{1, 2\}, A' = \{1', 2'\}$ and the ordering is $1 < 1' < 2 < 2'$. Note that all of the colored particles travel straight up on rows $x_1$ and $y_2$, as the labels $1, 2'$ do not appear in this tableau.

on alphabets $A, A'$. Stripping off ribbons label by label generates a series of skew shapes according to the total order: we define a sequence of partitions $\lambda_j$ such that the ribbon strip labelled $i_j \in A \cup A'$ will have shape $\lambda_{i_j}/\lambda_{i_{j-1}}$. Using this sequence, assign vertical edge labels to the state with boundary conditions $B_{\lambda/\mu}(X/Y)$. Thus, the columns will read $\mu = \lambda_0, \lambda_1, \ldots, \lambda_{r+s} = \lambda$, so that the $i_j$-th row is labelled $\lambda_{i_j}$ on the bottom and $\lambda_{i_{j-1}}$ on the top. (See Figure 5.8 for an example of this process.) Marshalling Propositions 5.9 and 5.10 together, this process defines a weight-preserving bijection and thus summing over all ribbon tableaux/states, the partition function of this system equals the desired super LLT polynomial. $\square$

## 6. Solvability of the ribbon lattice model

Lattice models satisfying a Yang–Baxter equation are called *exactly solvable*, or *integrable*. From the lattice model perspective, the Yang Baxter equation gives a consistent way of effectively permuting the weights of an admissible state without altering the partition function, thus providing a useful tool for showing symmetry or recursion relations on the partition functions.

The Yang–Baxter Equation for a $n$-ribbon lattice model requires choosing Boltzmann weights for a set of new *diagonal* vertices of in-degree $2n$ and out-degree $2n$, called $R^{(n)}$-vertices, such that the partition functions of the two lattice models in (6.1) are equal for any set of fixed boundary conditions. To differentiate, we will call the vertices of Section 5 *rectangular* from now on. For a complete treatment of Yang–Baxter equations of 2-$d$ square lattice models, we refer the reader to [BBF11, Section 1 & 5] and [Bax89, Chapter 8 & 9].

$$Z\left(\begin{array}{c} R_{i,j}^{(n)} \\ v_i \\ v_j \end{array}\right) = Z\left(\begin{array}{c} v_j \\ R_{i,j}^{(n)} \\ v_i \end{array}\right) \tag{6.1}$$

As we did with the horizontal and vertical strip row vertices (see Remark 5.6), we shall view these $R^{(n)}$-vertices as a combination of $n$ diagonal $R^{(1)}$-vertices, rather than the $n^2$ intersections visible in this graphical depiction. This combinatorial fusion will allow us to define the weight of a $R^{(n)}$-vertex as a product of weights of $n$ overlaid $R^{(1)}$-vertices.

**Definition 6.1.** Let $r$ be a $R^{(n)}$-vertex depicted as $n$ stacked $R^{(1)}$ vertices. That is, depicting $r$ as



where $I_k, J_k, K_k, L_k \in \{>, <\}$ for $k \in [n]$. Then define $\mathrm{wt}(r) = \prod_{k=1}^n \mathrm{wt}(r_k)$, where $\mathrm{wt}(r_k)$ is chosen from Figure 6.1 and depends on the types of rows being crossed.

*Remark* 6.2. As this construction is substantially different from visually similar $R^{(n)}$-vertices used in more traditional (i.e. non-twisted) lattice models, we may alternately index every intersection of strands $i$ and $j$ shown in the $R^{(n)}$ vertex as $r_{ij}$, with labels $I_{ij}, J_{ij}, K_{ij}, L_{ij}$ as in the diagram above. Then define the weight of $r_{ij}$ for $i \neq j$ to be

$$r_{ij} = \begin{cases} 1 & \text{if } I_{ij} = K_{ij} \text{ and } J_{ij} = L_{ij} \\ 0 & \text{else} \end{cases}$$

so these off-central vertices must transmit their labels straight through along each strand to the central $n$ vertices $r_{ii} := r_i$ (which we will often mark $\bullet$ to remind the viewer of this distinction).

Owing to the $q$-powers on the weights in Figures 5.3 necessary to match the super-LLT polynomials, the weights of certain types of $R^{(1)}$-vertices depend on the other types of vertices appearing in the $R^{(n)}$ vertex. In each of the cases of swapping rows of the same type (horizontal - horizontal or vertical - vertical) this dependence affects only powers of $q$ on one type of vertex. Considering swapping horizontal strip rows first, for the $k$-th piece $r_k$ in an $R^{(n)}$ vertex, let

$$\theta = 2 \cdot \#\{r_t = \text{S} | t > k\},$$
$$\sigma = \#\{r_t = \text{SS} | t > k\} + \#\{r_t = \text{NN} | t < k\} + \#\{r_t = \text{W}\}.$$

Graphically, $\theta$ is 2 times the number of S-vertices below, and $\sigma$ is the sum of the number of SS-vertices below, the number of NN-vertices above, and the number of all W-vertices. To encode this interpretation, we will thus write $\theta = \#S \downarrow$ and $\sigma = \#SS \downarrow + \#NN \uparrow + \#W$. Similarly, for $r_k$ in a $R^{(n)}$-vertex swapping vertical strip rows, we will need:

$$\theta' = 2 \cdot \#N \downarrow \qquad \text{and} \qquad \sigma' = \#SS \uparrow + \#NN \downarrow + \#W.$$

Lastly, for $r_k$ in vertices that swap rows of different type, we need the quantities

$$\theta'' = 2 \cdot \#E \downarrow \qquad \text{and} \qquad \sigma'' = \#SS \uparrow + \#NN \downarrow + \#S.$$

| label | N | SS | W | E | NN | S |
|---|---|---|---|---|---|---|
| $r_k$ |  |  |  |  |  |  |
| $\text{wt}_{HH}(r_k)$ | 0 | $x_j$ | $x_j$ | $x_i$ | $x_i$ | $\frac{q^\theta x_i - x_j}{q^{\theta+\sigma}}$ |
| $\text{wt}_{VV}(r_k)$ | $q^{\sigma'}(y_j - q^{\theta'} y_i)$ | $y_j$ | $y_i$ | $y_j$ | $y_i$ | 0 |
| $\text{wt}_{HV}(r_k)$ | $q^{\sigma''}(-y_j)$ | $q^{\sigma''}(-y_j)$ | 0 | $\frac{q^{2(n-1)} x_i - q^{\theta''} y_j}{q^{\theta''}}$ | $q^{\sigma''} x_i$ | $q^{\sigma''} x_i$ |

Figure 6.1: The weights for the $k$-th $R^{(1)}$-vertex in an $R^{(n)}$-vertex swapping rows $i, j$, where $HH$ denotes swapping horizontal strip rows, $VV$ swapping vertical strip rows, and $HV$ swapping a horizontal strip row $i$ with a vertical strip row $j$. Note that type VH weights do exist, however as we do not need them for our proofs, we leave their computation as an exercise for the reader.

**Example 6.3.** For instance, if $n = 4$ and we consider the two vertices below, where the first

vertex is type HH and the second is type HV, we see that

$$
wt_{HH}\left(\begin{matrix} \text{NN} \\ \text{S} \\ \text{SS} \\ \text{E} \end{matrix}\right) \;=\; x_i \cdot \frac{x_i - x_j}{q^2} \cdot x_j \cdot x_i
$$

$$
wt_{HV}\left(\begin{matrix} \text{NN} \\ \text{E} \\ \text{SS} \\ \text{SS} \end{matrix}\right) \;=\; q^0 x_i \cdot \frac{q^{2(n-1)} x_i - y_j}{q^0} \cdot q^0(-y_j) \cdot q^1(-y_j).
$$

**Theorem 6.4.** *Together with the horizontal strip row weights in Figure 5.3, the $HH$ Boltzmann weights for $R^{(n)}$-vertices given in Definition 6.1 give a solution to the Yang–Baxter equation for any $n \geqslant 1$.*

*Proof.* The key step to solving the Yang–Baxter equation for arbitrary $n$ lies in using the property that arrows do not change along the straight edges of rectangular vertices. This implies that most of the interior edges are fixed under a given choice of boundary conditions: notice that the bottom $n - 1$ pieces of the $R^{(n)}$-vertex displayed in Equation (6.2) on the left hand side are identical to the top $n - 1$ pieces of the $R^{(n)}$-vertex on the right hand side.

The only edges that can vary are $\phi, \xi, \psi$ on the left hand side and $\theta, \delta, \sigma$ on the right (labelled in red in Equation (6.2)), which are connected to twisted or vertical edges (highlighted in blue in Equation (6.2)), suggesting that these strands must be considered differently from the rest of the lattice model. We thus divide the lattice model into two parts. The first part is the strands connected to the twisted or vertical edges, which we will call the "underlying YBE." The second part is the remaining $n - 1$ $R^{(1)}$ vertices of the $R^{(n)}$ vertex, which we will call the "block" as it is the same on both sides of the YBE, and its connected straight edge strands in the rectangular vertices.



(6.2)

To prove the YBE we must examine how the block interacts with the rectangular vertices and with the underlying YBE. However, since the block is the same on both sides, its weight will be the same on both sides, once we account for any interaction with the $R^{(1)}$ matrix of the underlying YBE on each side. Therefore, it doesn't matter where in the block a given vertex

type occurs, only whether its weight is affected by or affects the weight of the underlying $R^{(1)}$ vertex on each side. This approach allows us to divide the $2^{4n+2}$ possible boundary conditions into cases according to the boundary conditions $(a, b_n, c_n, \alpha, \beta_1, \gamma_1)$ of the underlying YBE, then consider the effect of each of the 6 types of vertices that might show up in the block.

Since each vertex involved in the underlying YBE has an equal number of arrows going in and coming out on these strands, the boundary conditions must observe the same property in order to be admissible. Thus we have $\binom{6}{3} = 20$ cases determined by choosing 3 "in" arrows from these 6 edges. However, several cases will have all states having weight 0 on both sides and thus be vacuously true: specifically, when $\alpha = \text{in}$, $\beta_1 = \text{in}$ (cases B1, B2, B4, and B5) or when $b_n = \text{out}$, $a = \text{out}$ (cases B1, B3, B4, and B7). Note: in most of these cases, matching SE vertices cause the 0 weight, but in B5 and B7, the extra term has a type N vertex, which is precisely why type $N$ must have weight 0.

Eliminating these six sets of boundary conditions, we are left with 14 sets to check, given in Appendix A. In fact, since many of the vertices have similar weights, we may group into cases by the number of states and the number of $NW$ ("unusual") vertices on each side of the YBE.

In Cases 1-3, there is only one state on each side, so the weight of either side will have the form $q^a x_i^b x_j^c \prod_{k=0}^{d-1}(q^{2k} x_i - x_j)$ for some integers $a, b, c$, and $d = \#S$. However, setting $q = 1$ recovers an 180 degree rotation of the Schur model $S^\Delta$ of [BBF11] with $t = 0$, and our diagonal weights specialize to their YBE solution, accounting for this rotation. Thus, the powers $x_i^b x_j^c$ are equal on both sides, and it suffices to check that the powers of $q$ appearing in each factor match on both sides. Unfortunately, this method will not work for Cases 4 and 5, as there will be multiple terms on one side that need to cancel properly to match the single term on the other side: however, since the factors of $x$ in the $R$-vertex weights do not depend on other vertices, we will only need to consider those factors of $x$ in the underlying YBE, as those in the block will be equal on both sides. It may be helpful to the reader to refer to Examples 6.5 (for Case 1) and 6.6 (for Case 4) during the proof.

**Case 1: Equations (A.1) to (A.8)**   First consider Equations (A.1) to (A.8), which each have one state on each side of the YBE, but no NW vertices on either side. For example, Equation (A.1) is given below.



$$\text{(A.1)}$$

Denote the $R^{(n)}$-vertex on the left hand side $\mathcal{L}$ and the one on the right hand side $\mathcal{R}$. Denote the ribbon vertices on the left hand side $v_j, v_i$ and those on the right hand side $w_i, w_j$ as illustrated above, where $i, j$ are row indices. Note that $\mathcal{L}_1$ and $\mathcal{R}_n$ are the "underlying" $R^{(1)}$ vertices. The rest of the $R^{(1)}$ vertices are part of the "block," so $\mathcal{L}_{t+1} = \mathcal{R}_t$ for $t = 1, \ldots, n-1$. To prove the

YBE, we need to show

$$\mathrm{wt}(v_i)\,\mathrm{wt}(v_j)\prod_{t=1}^{n}\mathrm{wt}(\mathcal{L}_t) = \mathrm{wt}(w_i)\,\mathrm{wt}(w_j)\prod_{t=1}^{n}\mathrm{wt}(\mathcal{R}_t) \qquad\qquad (*)$$

In Equations (A.1) to (A.8), none of $v_i, v_j, w_i, w_j$ is of type NW, meaning that every left arrow in $\{b_1, \cdots, b_{n-1}, c_1, \cdots, c_{n-1}\}$ will contribute to a power of $q$ in the weight of $v_j$ or $v_i$. For the same reason, every left arrow in $\{\gamma_2, \cdots, \gamma_n, \beta_2, \cdots, \beta_n\}$ will contribute to a power of $q$ in the weight of $w_i$ or $w_j$. Let $s_\gamma$ be the number of left arrows in $\gamma_2, \ldots, \gamma_n$, $s_b$ be the number of left arrows in $b_1, \ldots, b_{n-1}$, and define $s_\beta, s_c$ similarly.

Then, by Definition 6.1, since any piece of $\mathcal{L}$ and $\mathcal{R}$ has the same number of in and out arrows, we have $s_b + s_c = s_\gamma + s_\beta$, so the powers of $q$ in $\mathrm{wt}(v_i)\,\mathrm{wt}(v_j)$ and $\mathrm{wt}(w_i)\,\mathrm{wt}(w_j)$ match. Furthermore, neither $\mathcal{L}_1$ nor $\mathcal{R}_n$ are of type S, so the weight of the underlying YBE does not depend on the vertices in the block. Therefore, it suffices to check that the part of the block's weight by the underlying YBE is the same on both sides, i.e., that

$$\prod_{t=2}^{n}\mathrm{wt}(\mathcal{L}_t) = \prod_{t=1}^{n-1}\mathrm{wt}(\mathcal{R}_t)$$

Only type S vertices in the block can be affected by $\mathcal{L}_1$ or $\mathcal{R}_n$, so the equality holds when there is no S-vertex in the block. Otherwise, in Equations (A.1) to (A.4), $\mathcal{L}_1 = NN$ or $W$ and $\mathcal{R}_n = SS$ or $W$, so each S vertex in the block "sees" $\mathcal{L}_1$ above on the left hand side and $\mathcal{R}_n$ below on the right in its power of $q^{-\sigma}$. Since the number of $S$ vertices is the same in the block on the right and left, this gives a power $q^{-\#S}$ on both sides. In Equations (A.5) to (A.8), $\mathcal{L}_1 = SS$ or $E$ and $\mathcal{R}_1 = NN$ or $E$, so they do not affect the weight of $S$ vertices in the block and we are done.

**Example 6.5.** Consider the following boundary conditions within Equation (A.1) of Case 1.
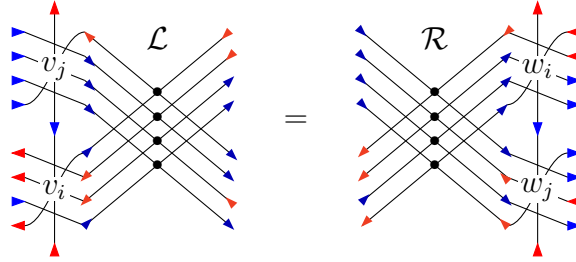


On the left hand side, $wt(v_i) = q^2, wt(v_j) = x_j$, and $\mathcal{L}_1$ is type NN and the S, SS, E vertices are the block. We calculated in Example 6.3 that $wt(\mathcal{L}) = x_i \cdot q^{-1} \cdot (q^{-1}(x_i - x_j)x_i x_j)$, where the first factor is $\mathrm{wt}(\mathcal{L}_1)$, the last factor is the internal weight of the block, and the middle $q^{-1}$ is the part of the block weight affected by $\mathcal{L}_1$. Thus this side has weight $(x_i - x_j)x_i^2 x_j^2$. On the right hand side, $wt(w_i) = qx_i, wt(w_j) = q$, and $\mathcal{R}$ is made of vertices S, SS, E, SS from top to bottom, where $\mathcal{R}_n$ is the SS. Similarly, $wt(\mathcal{R}) = x_j \cdot q^{-1} \cdot (q^{-1}(x_i - x_j)x_i x_j)$, so this side also has weight $(x_i - x_j)x_i^2 x_j^2$ and the YBE holds.

**Case 2: Equation (A.9)** Next we look at Equation (A.9), which has one state on each side, each with one NW vertex. Note that both $\mathcal{L}_1$ and $\mathcal{R}_n$ are type S.



$$(A.9)$$

As in Case 1, we examine the different types of vertices that might appear in the block and how each affects the underlying YBE. For each of type SS, NN, W, and E appearing in the block, the total power of $q$ acquired on either side is $0$: for instance, a type SS vertex garners a power of $q^{-1}$ from $\mathcal{L}_1$ but a power of $q$ from $\mathrm{wt}(v_i)$ on the left hand side, and has no effect on the right. Types NN, W, and E follow similarly.

Unlike Case 1, type S vertices in the block are slightly more complicated, as now $\mathcal{L}_1$ and $\mathcal{R}_n$ affect the factor $q^{\theta}x_i - x_j$. Checking the power $q^{\sigma}$ follows analogously to in Case 1. For this additional factor, note that this part of the weight on each side depends only on the total number of $S$ vertices in the $R^{(n)}$ vertex: i.e., it will always be $\prod_{k=0}^{d-1}(x_i - q^{-2k}x_j)$, where $d = \#S$, counting $\mathcal{L}_1$ on the left or $\mathcal{R}_n$ on the right. Thus, the factors coming from each $S$ vertex cycle to match on the left and right, with $\mathcal{L}_1$'s weight replacing that of the first S vertex on the right and $\mathcal{R}_n$'s weight replacing that of the last S vertex on the left.

**Case 3: Equation (A.10)** This case is the simplest case to check: the underlying YBE and the block $R$-vertices have no interaction with each other, and none of the rectangular weights admit additional powers of $q$, so we are done.

**Case 4: Equations (A.11) and (A.12)** The last four sets of boundary conditions each have one side with two states and one side with one: we begin with those with the extra state on the left. Consider Equation (A.11).



$$(A.11)$$

As in Case 2, type E, W, NN, and SS only affect the partition function by contributing powers of $q$, which the reader may check match on each term and thus on each side.

However, unlike Cases 1-3, type S vertices in this block require us to consider the factors involving $x_i, x_j$ in the underlying YBE as well, since we need terms on the left to cancel correctly to match the weight on the right. Let $\#(-)$ return the number of vertices of a given type in the block, and recall the definitions of $s_b, s_c, s_\beta$, and $s_\gamma$ from Case 1.

We consider the right hand side first: ignoring weights that are internal to the block and thus constant across all states, we have a factor of $q^{s_\gamma} x_i$ from $w_i$ and a factor of $x_i$ from $\mathcal{R}_n$.

On the left hand side, the first state has $\mathrm{wt}(u_j) = q^{s_b} x_j$ and a factor of $q^{-\#(S)} x_i$ from $\mathcal{K}_1$, since it is type NN and therefore contributes a power of $q^{-1}$ to every type S vertex below it. In the second term, $\mathrm{wt}(v_j) = q^{s_b}$, $\mathrm{wt}(v_i) = q^{s_c} x_i$, and $\mathrm{wt}(\mathcal{L}_1) = \frac{q^{2\#(S)} x_i - x_j}{q^{2\#(S)+\#(W)+\#(SS)}}$. So, ignoring the weight internal to the block, the left hand side is

$$x_i \left( q^{s_b - \#(S)} x_j + q^{s_b + s_c - \#(W) - \#(SS)} x_i - q^{s_b + s_c - 2\#(S) - \#(W) - \#(SS)} x_j \right).$$

Noting that $s_b = \#(W) + \#(NN)$ (recall that type N has weight 0) and $s_c = \#(W) + \#(S) + \#(SS)$, so the $x_j$ terms cancel and the power of $q$ on the $x_i$ term becomes $\#(W) + \#(NN) + \#(S) = s_\gamma$, so our YBE is satisfied.

Since rectangular vertex types SW and NS have the same weight, as do types EW and NE, Equation (A.12) is follows similarly.

**Example 6.6.** Consider the following boundary conditions within Equation (A.13) of Case 4, which use the same block (S,SS,E) as Example 6.5.
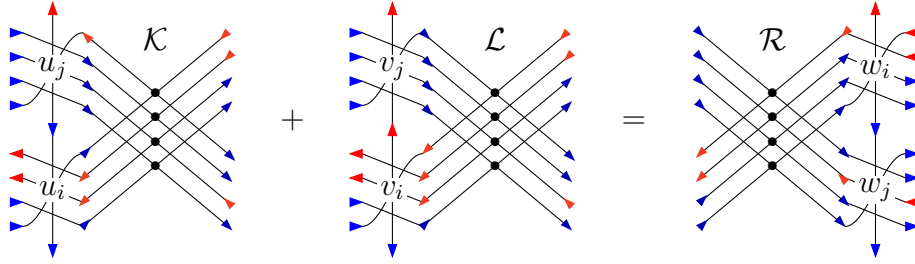


On the first left hand state, $wt(u_i) = 1, wt(u_j) = x_j$, and $\mathcal{K} = $ NN, S, SS, E, so $wt(\mathcal{K}) = q^{-1} x_i \left( q^{-1}(x_i - x_j) x_i x_j \right)$, where again the last factor is the internal block weight. On the second state, $wt(v_i) = q^2 x_i, wt(v_j) = 1$, and $wt(\mathcal{L}) = q^{-3}(q^2 x_i - x_j) \left( q^{-1}(x_i - x_j) x_i x_j \right)$, because the composition of $\mathcal{L}$ is S, S, SS, E. Ignoring the internal block factor, adding these weights gives

$$q^{-1} x_i x_j + q^{-1} x_i (q^2 x_i - x_j) = q x_i^2.$$

On the right hand side, $wt(w_i) = q x_i, wt(w_j) = 1$, and $\mathcal{R}$ is S, SS, E, E, so $wt(\mathcal{R}) = x_i \left( q^{-1}(x_i - x_j) x_i^2 x_j \right)$, so this side also has weight $q x_i^2 \left( q^{-1}(x_i - x_j) x_i^2 x_j \right)$.

**Case 5: Equations (A.13) and (A.14)**   The last two sets of boundary conditions work analogously to Case 4, except that the additional state falls on the right hand side, and the two sets of boundary conditions Equation (A.13) and Equation (A.14) differ by one extra power of $x_j$ appearing in every term of Equation (A.14).                                                                 $\square$

**Theorem 6.7.** *Together with the $V$ weights in Figure 5.3, the $VV$ Boltzmann weights for $R^{(n)}$-vertices given in Figure 6.1 give a solution to the Yang–Baxter equation for any $n \geqslant 1$.*

*Proof.* The proof is nearly identical to that of Theorem 6.4, so we sketch the broad strokes and leave the details to the reader. Since the $V$ weights in Figure 5.3 set type NE vertices to have weight 0 instead of type SE as before, the set of boundary conditions to check is slightly different.

The six weight $0$ conditions in the proof of Theorem 6.4 will be nonzero cases, which the reader may find in Appendix B as (B.1), (B.2), (B.3), (B.4), (B.5), and (B.7). Cases (A.2), (A.3), (A.4), (A.9), (A.12), and (A.14) replace them as the weight $0$ cases, and cases (A.6) and (A.8) both acquire an additional state on one side, which we reproduce as cases (B.6) and (B.8). From there, cases split by number of states per side of the YBE and number of SW vertices, as SW is now the "unusual" vertex in terms of power of $q$:

- Case 1 (one state per side, no SW vertices): cases (A.1), (A.7), (A.10), (A.11), (A.13), (B.1), (B.2), (B.3).

- Case 2 (one state per side, one SW vertex per side): cases (B.4).

- Case 3 (one state per side, two SW vertices per side): cases (A.5).

- Case 4 (two states on the left, one on the right): cases (B.5), (B.6), and (B.5) = (B.6).

- Case 5 (one state on the left, two on the right): cases (B.7), (B.8), and (B.7) = $y_i \cdot$(B.8).  $\square$

**Theorem 6.8.** *Together with the weights in Figure 5.3, the HV Boltzmann weights for $R^{(n)}$-vertices given in Figure 6.1 give a solution to the Yang–Baxter equation for any $n \geqslant 1$.*

*Proof.* Again, the proof is analogous to Theorem 6.4; however, it is important to note the order of the row types: we can only attach a type HV vertex if the horizontal and vertical strip rows are in the correct order, i.e., horizontal connects to the "$i$"-strand of the $R^{(n)}$-vertex and vertical connects to the "$j$"-strand of the vertex. With this in mind, the boundary conditions fall into the following cases, as indexed in Appendices A and B:

- Case 0 (weight $0$): (A.2), (A.3), (A.4), (B.1), (B.2), (B.3).

- Case 1 (one state per side, no unusual vertices): (A.1), (A.7), (A.9), (A.12), (A.14), (B.4), (B.5), (B.7).

- Case 2 (one state per side, a horizontal NW): (A.10).

- Case 3 (one state per side, a vertical SW): (A.5).

- Case 4 (two states on the left, one on the right): (A.11), (B.6), (A.11) = (B.6).

- Case 5 (one state on the left, two on the right): conditions (A.13), (B.8), and (A.13) = (B.8).  $\square$

**Proposition 6.9** ([Lam05], Prop 30). *The function $\mathcal{G}_{\lambda/\mu}(X/Y;q)$ is symmetric in each of $X$ and $Y$ and does not depend on the total order fixed between $A$ and $A'$.*

*Proof.* For a variable set $X = (x_1, \ldots, x_m)$, $Y = (y_{1'}, \ldots, y_{r'})$, consider the total ordering $1 < 2 < \cdots < m < 1' < 2' \cdots < r'$. To prove each of these symmetries, we will use train arguments with the Yang–Baxter equation for different row types proven in Theorems 6.4, 6.7, and 6.8. For symmetry in $X$, consider the partition function of the following lattice model, where the $R^{(n)}$-vertex is attached to horizontal strip rows $i, i + 1$:
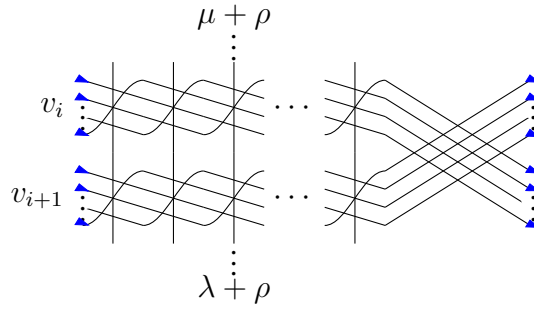
Since all of the edges on the right boundary point right, the only choice for each $R^{(1)}$-vertex is type E, which gives this system a total weight of $x_i^n \cdot \mathcal{G}_{\lambda/\mu}^{(n)}(X/Y; q)$. As in Theorem 6.4, push the diagonal vertex all the way to the right, column by column, until it emerges into the right boundary. This process results in the following system:
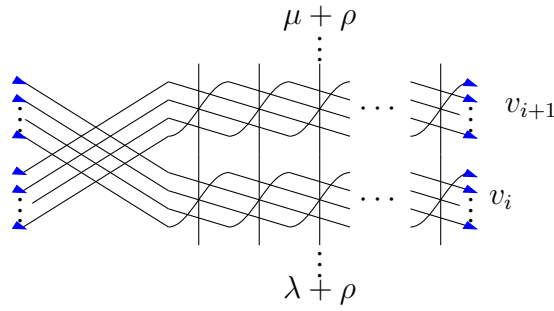


Similarly, the $R^{(n)}$-vertex in this system must be $n$ copies of the type $E$ vertex, so this side has weight $\mathcal{G}_{\lambda/\mu}^{(n)}(s_i X/Y; q) \cdot x_i^n$, where $s_i X = (x_1, \ldots, x_{i+1}, x_i, \ldots, x_m)$. Using Theorem 6.4, these weights are equal, so $\mathcal{G}_{\lambda/\mu}^{(n)}(X/Y; q)$ is symmetric in the $X$ variables. The proof of symmetry in the $Y$ variables is nearly identical, attaching the $R^{(n)}$-vertex to rows $i'$ and $(i+1)'$. To prove total order, start with the system where all horizontal strip rows appear above all vertical strip rows, and use the train argument with type HV $R^{(n)}$ vertices to move horizontal strip rows gradually below vertical strip rows; since any total order respects the internal orderings on $\{1, \ldots, m\}$ and $\{1', \ldots, r'\}$, it is possible to achieve any total ordering desired and the resulting braid of all $E$ vertices will have the same weight on either side and thus cancel off as in previous cases.       □

## 7. Branching rules

The structure of the ribbon lattice gives rise to natural combinatorial proofs of branching rules for super LLT polynomials: split the lattice between any two rows and compute the partition function of a grid as two pieces versus as one whole. Splitting off a single row in the following proposition (i.e. setting $k = 1$) will reproduce the branching rules of Corollary 4.10. In this way, we see that the lattice model diagrammatically encodes the operators $U(x)$ and $\widetilde{U}(-y)$.

**Proposition 7.1.** *Given alphabets $A$, $A'$ and a total order, let $z_i$ stand in for the $x$ or $y$ variable at position $i$ in the total order. Let $r = |A| + |A'|$ and fix some $k = 1, \ldots, r$. Then the super LLT polynomials satisfy the following general branching rule:*

$$\mathcal{G}_{\lambda/\mu}^{(n)}(z_1, \ldots, z_r; q) = \sum_\gamma \mathcal{G}_{\lambda/\gamma}^{(n)}(z_{k+1}, \cdots, z_r; q) \cdot \mathcal{G}_{\gamma/\mu}^{(n)}(z_1, \cdots, z_k; q).$$

*where the sum runs over all partitions $\gamma$. Note that the only nonzero terms in the sum will be those $\gamma$ for which both $\lambda/\gamma$ and $\gamma/\mu$ admit super $n$-ribbon tableaux in the respective subsets of the total order.*

*Specifically, we may describe these $\gamma$ as partitions for which there exists a sequence of compositions $\gamma_0 = \mu + \rho, \gamma_1, \ldots, \gamma_k = \gamma + \rho, \gamma_{k+1}, \ldots, \gamma_r = \lambda + \rho$ such that:*

1. *$\gamma_i - \gamma_{i-1} \in nZ^{\ell(\lambda)}$ for all $i$.*

2. *If row $i$ is a horizontal strip row, $(\gamma_i)_j \neq (\gamma_{i-1})_k$ for all $j \neq k$.*

3. *If row $i$ is a vertical strip row, $(\gamma_i - \gamma_{i-1})_j \in \{0, n\}$ for all $j$.*

*Proof.* Consider the boundary conditions $\mathcal{B}_{\lambda/\mu}(X/Y)$ and let $z_i$ denote the spectral parameter on row $i$. Slicing the diagram horizontally along the vertical edges between rows $k$ and $k + 1$, consider all choices of labels for the sliced edges, which will each be of the form $\gamma + \rho$ for some partition $\gamma$.
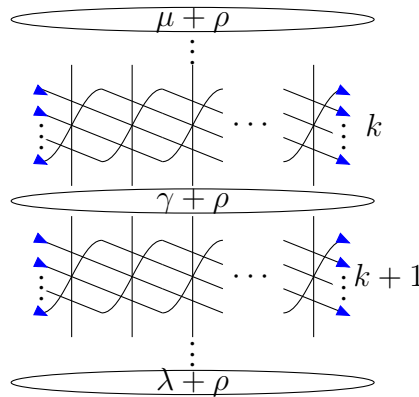


Then rows $1, \ldots, k$ will have boundary conditions $\mathcal{B}_{\gamma/\mu}$ and rows $k + 1, \ldots, r$ will have boundary conditions $\mathcal{B}_{\lambda/\mu}$. Note that the only nonzero terms in this sum will be those for $\gamma$ such that $\gamma/\mu, \lambda/\gamma$ are both tileable with $n$-ribbons such that ribbons labelled $i$ form a horizontal strip if $z_i = x_i$ and a vertical strip if $z_i = y_i$.

Using the particle description of the model, the composition $\gamma_i$ labels the columns between row $i - 1$ and row $i$ containing a particle. The first condition follows from the fact that particles can only move along a row in multiples of $n$ vertices. Within this condition, horizontal rows require that the same label only appear in $\gamma_{i-1}$ and $\gamma_i$ if it comes from a particle passing straight up through the row, i.e. a type SW vertex as opposed to a type SE. On the other hand, vertical rows require that each particle must either pass straight through or travel one single step of $n$ particles, since the type NE vertex which allows a particle to travel multiple sets of $n$ particles has weight 0. $\square$

# 8. The superLLT Cauchy lattice model

A slight adaptation of the super LLT lattice model, obtained by reversing the direction of travel of the particles from left to right, gives a combined solvable *Cauchy model*, suggestively named to relate to our Cauchy identity in Section 4. In the manner of similar lattice model Cauchy identities for other polynomials, our Cauchy model is formed by stacking rows of the alternate model together with rows of the original model. We will use this combined model in Section 9 to provide lattice model proofs of the superLLT Cauchy and Pieri identities developed in Section 4.
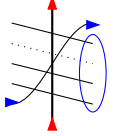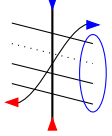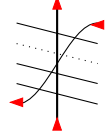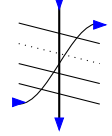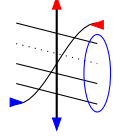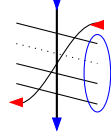
| Label | SW | NS | SE | NW | EW | NE |
|-------|-----|-----|-----|-----|-----|-----|
| Vertex | | | | | | |
| $wt_{\widetilde{H}}$ | $0$ | $q^t x_i$ | $q^t$ | $q^t x_i$ | $q^t$ | $1$ |
| $wt_{\widetilde{V}}$ | $-y_i$ | $-y_i$ | $q^t$ | $0$ | $1$ | $1$ |

Figure 8.1: The weights of vertices lying in row $i$ for the alternate $n$-super ribbon lattice model, where $t$ is the number of *right* arrows in the blue circled area.

**Definition 8.1.** Given a total ordering on alphabets $B, B'$, a skew partition $\lambda/\mu$, and a positive integer $r$, let the *alternate model* $\widetilde{\mathcal{B}}_{\lambda/\mu}(X/Y)$ be the lattice model with weights determined from Figure 8.1 and boundary conditions as follows:

- edges on the left and right boundaries are labelled $<$,

- edges on the top boundary are labelled $\wedge$ on parts of $\lambda + \rho$ and $\vee$ else (numbering columns left to right starting with 1 as before),

- edges on the bottom boundary are labelled $\wedge$ on parts of $\mu + \rho$ and $\vee$ else,

- *horizontal* strip rows labelled $i \in B$ have spectral parameters $x_i$ and use weights $\widetilde{H}$,

- *vertical* strip rows labelled $i' \in B'$ have spectral parameters $y_{i'}$ and use weights $\widetilde{V}$,

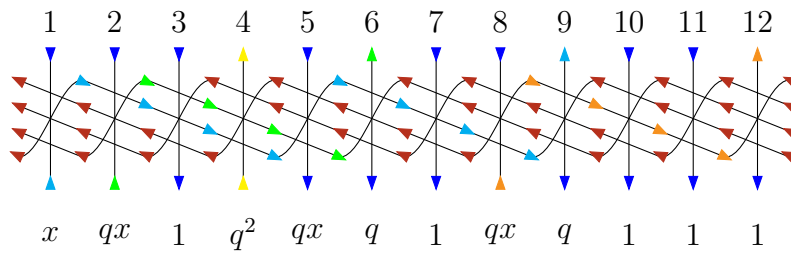- reading from top to bottom, rows are labelled in increasing order by to the total order.

Figure 8.2: The single horizontal strip row lattice state for $\mu = (4, 1)$, $\lambda = (8, 6, 4, 3)$, and $n = 4$ in the alternate model. Note: the weight is $q^7 x^4$, equal to that of the matching state in Figure 5.6.

**Theorem 8.2.** *Given a skew partition $\lambda/\mu$ and a total order on alphabets $B, B'$, we have*

$$Z\left(\widetilde{\mathcal{B}}_{\lambda/\mu}(X/Y)\right) = \mathcal{G}_{\lambda/\mu}^{(n)}(X/Y; q).$$

The proof of Theorem 8.2 is entirely analogous to that of Theorem 5.7, viewing particles in the system as travelling up and right as opposed to up and left. It is helpful to think of this alternate model as removing $n$ ribbon strips from $\lambda$ (on the top boundary) to reach $\mu$ (on the bottom boundary), whereas our original model added $n$ ribbon strips to $\mu$ (on the top boundary) to obtain $\lambda$ (on the bottom boundary). In this way, the alternate model corresponds to the polynomial $G_{\lambda/\mu}$ from Section 3, whereas the original model corresponds to $F_{\lambda/\mu}$.

This interpretation allows us to stack our original model atop or below this new one and evaluate the results to give the sum sides of the generic Cauchy and Pieri identities for super LLT polynomials. However, since horizontal ribbon strips may be arbitrarily long (i.e., their skew partitions have arbitrarily large parts) and vertical ribbon strips may be arbitrarily tall (i.e., their skew partitions have arbitrarily many parts), we need to modify the model slightly to account for all possible partitions that could appear on the boundary between original and alternate models. We do this by taking inspiration from the Fock space: given infinitely many columns indexed in $\mathbb{Z}$ and a partition $\mu$, place particles (up arrows) on parts of $\mu + \rho$ as well as on non-positive columns. This method produces an infinite sea of particles to the left of the columns involved in $\mu + \rho$ and an infinite void of holes to the right, mimicking the particle-hole interaction of Hamiltonian operators on the Fock space. Note: if the original model portion is all horizontal strip rows and the alternate all vertical strip rows (or vice versa), the interaction between horizontal and vertical ribbon strips will restrict the partitions so that only finitely many columns are necessary.

**Definition 8.3.** Consider four ordered alphabets $A, A', B, B'$ together with total orderings $\prec_A$ on $A \cup A'$ and $\prec_B$ on $B \cup B'$ that respect the orderings on $A, A'$ and $B, B'$ respectively. A *overarching total order $I$* is a total ordering on $A, A', B, B'$ that respects the individual orderings on each alphabet as well as the intermediate total orderings $\prec_A$ and $\prec_B$.

For example, let $I_A$ be the overarching total order where $a_i < b_j$ for any $a_i \in A \cup A'$, $b_j \in B \cup B'$, and $I_B$ be the overarching total order where $a_i > b_j$ for all $a_i \in A \cup A', b_j \in B \cup B'$.

**Definition 8.4.** Consider two pairs of totally ordered alphabets $A/A'$ and $B/B'$, with spectral parameters $X/Y$ and $W/Z$, respectively. Let $\mu, \nu$ be partitions, padded with zero parts to have
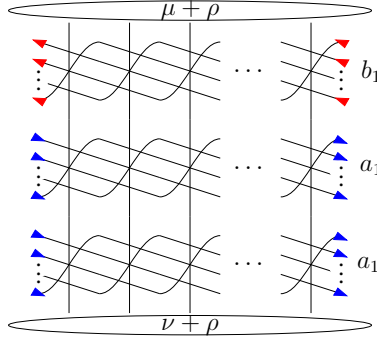
Figure 8.3: An example of boundary conditions for the Cauchy lattice model, under the overarching order $b_1 < a_1 < a_{1'}$ for $A = \{a_1\}$, $A' = \{a_{i'}\}$, $B = \{b_1\}$.

the same length, and define an overarching total order $I$ on $A$, $A'$, $B$, $B'$ that respects all existing orderings. We define the *Cauchy system* $\mathcal{C}_{\mu,\nu,I}(X/Y, W/Z)$ to have the following boundary conditions and spectral parameters on a semi-infinite lattice:

- columns are labelled left to right with indices in $\mathbb{Z}$, and designating a fixed column 1,

- edges on the top boundary are labelled by $\mu + \rho$ in the manner described above,

- edges on the bottom boundary are labelled by $\nu + \rho$,

- rows are labelled in increasing order from top to bottom according to the overarching total order $I$,

- horizontal strip rows labelled by $A$ (respectively $B$) take weights $wt_H$ (respectively $wt_{\widetilde{H}}$) and spectral parameters $x_i$ (respectively $w_i$), and

- vertical strip rows labelled by $A'$ (respectively $B'$) take weights $wt_V$ (respectively $wt_{\widetilde{V}}$) and spectral parameters $y_{i'}$ (respectively $z_{i'}$).

- side boundary edges on $A/A'$ rows are labelled $>$ and those on $B/B'$ rows are labelled $<$.

Here, we consider the spectral parameters as formal variables in order to remove issues of convergence of the partition function, since the semi-infinite model will produce infinitely many states for some choices of boundary conditions on the Cauchy model.

**Proposition 8.5.** *If we choose the overarching order $I_A$, then*

$$Z(\mathcal{C}_{\mu,\nu,I_A}(X/Y, W/Z)) = \sum_\lambda \mathcal{G}_{\lambda/\mu}(X/Y; q) \cdot \mathcal{G}_{\lambda/\nu}(W/Z; q),$$

*where the sum runs over all partitions $\lambda$. If instead we choose the order $I_B$, then*

$$Z(\mathcal{C}_{\mu,\nu,I_B}(X/Y, W/Z)) = \sum_\lambda \mathcal{G}_{\mu/\lambda}(X/Y; q) \cdot \mathcal{G}_{\nu/\lambda}(W/Z; q),$$

*where the sum runs over all $\lambda$.*

*Proof.* This proof is similar to that of Proposition 7.1. For the first case, the overarching order places the $A$ and $A'$ rows above the $B$ and $B'$ rows. Slicing along the lattice model in between these two pieces, the cut edge is labelled by $\lambda + \rho$ for some partition $\lambda$. Note: it is possible that $\lambda$ has more parts than $\mu$ or $\nu$, in which case some of the particles travelling to fill parts of $\lambda + \rho$ will be coming from the $-\infty, \dots, 0$ columns. Since the partition function does not depend on the column numbers at all, shifting the designated "1" column left to accommodate all travelling particles results in the same partition function and is equivalent to padding $\mu, \nu$ with zero parts to have the same number of parts as $\lambda$. The top half then has boundary conditions $B_{\lambda/\mu}(X/Y)$ and gives $\mathcal{G}_{\lambda/\mu}(X/Y; q)$, whereas the bottom half has boundary conditions $\widetilde{B}_{\lambda/\nu}(W/Z)$ and gives $\mathcal{G}_{\lambda/\nu}(W/Z)$. The second case follows similarly, with the additional note that any partition $\lambda$ such that $\mu/\lambda$ and $\nu/\lambda$ are both skew shapes will have at most the same number of parts as $\mu, \nu$. If $\lambda \not\subset \mu$ or $\lambda \not\subset \nu$ then one of the super LLT polynomials in the sum will be zero, i.e. there will be no non-zero filling of the lattice model, so we may sum over all partitions $\lambda$. $\qquad\square$

**Theorem 8.6.** *Along with the $R$-vertex weights in Figure 8.4 and Appendix C, the $H, \widetilde{H}, V$ and $\widetilde{V}$ weights satisfy mixed YBEs in all combinations.*

As in the Yang–Baxter equation for the original superribbon model, factors of $q$ appear on many of these vertices, adjusting for the fact that there is one vertex with a different spin than the others in each of models. These factors will be similar to the $\sigma, \sigma'$, and $\sigma''$ appearing in Section 6. For an $R^{(1)}$ vertex $r_k$ in a $R^{(n)}$ vertex, set

$$\tau = \#SS\downarrow + \#NN\uparrow \qquad \text{and} \qquad \kappa = \#SS\uparrow + \#NN\downarrow.$$

*Proof.* These proofs are analogous to those of Theorems 6.4, 6.7, and 6.8 so we omit them for length. The weights of type $\widetilde{A}B$, which cross an alternate model $\widetilde{A}$ row past a original model $B$ row, are given in Figures 8.4 and 8.6. As we will not explicitly use the weights of type $A\widetilde{B}$ in this paper, they are displayed in Appendix C.

*However*, one new phenomenon arises in these weights: rather than the standard combinatorial fusion of multiplying all $n$ of the $R^{(1)}$ weights together to obtain the $R^{(n)}$ weight, as described in Section 6, a curious "pre-fusion" of pairs of vertices occurs. For Cauchy model YBEs of type $\widetilde{A}B$, if the $R^{(n)}$ vertex contains $t$ pairs of a type $NN$ vertex appearing above a type $SS$ vertex, these paired $NN$ and $SS$ vertices combine into a "type NN/SS" $R^{(2t)}$-vertex, whose weight is not merely the product of the individual $2t$ vertices. Any remaining unpaired NN and SS vertices retain their usual weight; for the sake of convention, we choose pairs from the outside in, i.e., the first $t$ NN vertices and the last $t$ SS vertices, unpaired vertices "inside" the pre-fusing vertices contribute change the weight of the prefusion vertex, whereas those "outside" do not. The weight of a prefused vertex is prescribed by two cooperating recursive conditions arising from the YBE, one that removes an NN vertex from the top and one that removes a SS vertex from the bottom. Fusion into the $R^{(n)}$ vertex weight then proceeds by multiplication using this new vertex. If this condition does not occur, for instance if all type SS vertices appear above any type NN vertex, the normal fusion process occurs.

As prefusion vertices will not occur in our proofs in Section 9, we will postpone further discussion of this structure to future papers, but Figure 8.6 gives the defining recursive conditions

for type $\widetilde{A}B$ vertices and Figure 8.5 the resulting weights for prefused vertices with a single NN/SS pair. The reader may also wish to consult Example 8.7 for some further examples of the distinction between prefused and non-prefused vertices, or Example 8.8 for an example of how this prefusion arises from the YBE. $\qquad\square$

**Example 8.7.** For instance, if $n = 4$, consider $R^{(4)}$ vertices of type $\widetilde{H}V$ swapping rows $w$ and $y$, displayed with their weight factors coming from each $R^{(1)}$ vertex below. In the five examples below, we see that a single pair pre-fusion arises in the first three cases, no pre-fusion occurs in the fourth, and that the entire vertex pre-fuses into a two pair pre-fusion vertex in the last case. The pre-fusing vertices and their weights are displayed in green in each case.



$$
\begin{array}{cccc}
1 & q^4 - q^6 yw - 1 & q^6 - q^6 yw - 1 & q \\
q^2 - q^4 yw - 1 & q^3(1 - yw) & -q^2 yw & -yw \\
q(1 - yw) & -q^2 yw & -qyw & 1 \\
& & & q
\end{array}
$$



$$
\mathrm{wt}\left(\begin{array}{c}\vbox{\hbox{NN}\hbox{NN}\hbox{SS}\hbox{SS}}\end{array}\right) = \mathrm{wt}\left(q^{2-1}\begin{pmatrix} E \\ NN \\ NN \\ SS \end{pmatrix}\right) - \mathrm{wt}\left(q^{0+1+0}\begin{pmatrix} NN \\ NN \\ SS \\ S \end{pmatrix}\right)
$$

$$
= q^4(1 - yw)(q^4 - q^6 yw - 1) - q^2(q^4 - q^4 yw - 1)
$$
$$
= q^{10}(yw)^2 - (q^{10} + q^8 - q^6 - q^4)yw + (q^8 - q^6 - q^4 + q^2)
$$

Similarly, for Cauchy model YBEs of type $A\widetilde{B}$, this "pre-fusion" combines the first instance of a type SS vertex with the first NN vertex below it to create "type SS/NN" vertex.

**Example 8.8.** For example, considering the Yang–Baxter equation mixing rows of types $\widetilde{H}$ (with parameter $w$) and $V$ (with parameter $y$) with the following boundary conditions, we see that the pre-fusion weight is necessary to make the model solvable. If we replaced $\mathrm{wt}(\frac{NN}{SS}) = q^2 - q^2 yw - 1$ with $\mathrm{wt}(NN) \cdot \mathrm{wt}(SS) = -yw$ here, the partition functions on each side would not match.



$$
\begin{array}{cccc}
-q^2 yw & -q^2 yw(-yw) & -yw & -yw(q^2 - q^2 yw - 1)
\end{array}
$$

| Label | $r_k$ | $\mathrm{wt}_{\widetilde{H}H}$ | $\mathrm{wt}_{\widetilde{V}H}$ | $\mathrm{wt}_{\widetilde{H}V}$ | $\mathrm{wt}_{\widetilde{V}V}$ |
|---|---|---|---|---|---|
| W | | $q^\tau$ | $(1-q^{2W\uparrow}x_jz_i)q^\tau$ | $q^\tau$ | $q^\tau$ |
| S | | $(1-q^{2n-2-2S\downarrow}x_jw_i)$ | 1 | 1 | 1 |
| SS* | | $q^{W+SS+NN+N+E}x_jw_i$ | $-x_jz_i\cdot q^{SS\downarrow+W}$ | $-y_jw_i\cdot q^{SS\downarrow+E}$ | $y_jz_j$ |
| NN* | | $q^{W+SS+NN+N+E}$ | $q^{NN\uparrow+W}$ | $q^{NN\uparrow+E}$ | 1 |
| N | | $-q^{W+SS+NN+N+E}$ | $-q^{n-1-S}$ | $-q^{n-1-S}$ | $(y_jz_i-q^{2N\downarrow})q^{W+E}$ |
| E | | $q^{\tau-W}$ | $q^{\tau+W}$ | $(1-q^{2E\downarrow}y_jw_i)q^{\tau+W}$ | $q^{\tau+W}$ |

Figure 8.4: The weights for a normal (i.e. not "pre-fused") $R^{(1)}$-vertex in an $R^{(n)}$-vertex swapping rows $i,j$, where strand $i$ is in the alternate model and strand $j$ is in the original model. For the sake of space, we have eliminated the $\#$ in these weights, so $W := \#W$ and so on.

| Type | Single NN/SS Pair Prefusion Weight |
|---|---|
| $\mathrm{wt}_{\widetilde{H}H}$ | $q^{2\iota+2} - 1 + q^{2n-2-2\#S}xw$ |
| $\mathrm{wt}_{\widetilde{V}H}$ | $q^{2\iota+2} - q^{2\iota+2\#W+2}xz - 1$ |
| $\mathrm{wt}_{\widetilde{H}V}$ | $q^{2\iota+2} - q^{2\iota+2\#E+2}yw - 1$ |
| $\mathrm{wt}_{\widetilde{V}V}$ | $yz + q^{2\iota+2\#N+2} - q^{2\#N}$ |

Figure 8.5: The weights for a pre-fused NN/SS vertex with a single matched NN/SS pair in an $R^{(n)}$ vertex of type $\widetilde{A}B$ swapping rows $i,j$. Let $\iota :=$ the number of unpaired NN or SS vertices *inside* the prefusing pair (See Example 8.7.) Note: these unpaired vertices will be all of one kind (NN or SS), else there would be two matched pairs.

*Remark* 8.9. Unlike the R-vertices considered in Section 6, whose weights arise from a diagrammatic generalization of fusion of tensor products of quantum group modules, this set of mixed R-vertices is fascinating because *there is no currently known quantum algebraic object that corresponds to this sort of "pre-fusion" procedure.*

# 9. SuperLLT Pieri and Cauchy identities on the lattice model

In the spirit of previous lattice model proofs for Cauchy identities, such as that for factorial Schur functions given in [BMN14], we use the Cauchy model to prove the Pieri and Cauchy identities of Section 4 diagrammatically by attaching a braid of $R^{(n)}$ vertices to one side of the semi-infinite Cauchy model (see Figure 9.8), and using the Yang–Baxter equation to pass this braid through to the other side. Although the models in this section are semi-infinite, we may ensure this computation is well-defined by identifying infinite swathes of columns with weight one, thereby matching the truncation to a finite model with the same partition function. It would

| Type | Top NN Recursion | Bottom SS Recursion |
|------|------------------|---------------------|
| $\mathrm{wt}_{\widetilde{H}H}$ | $\binom{NN}{*} = \frac{1}{xw}\left(q^{SS-2NN-E-N}\binom{*}{W}+\binom{S}{*}\right)$ | $\binom{*}{SS} = q^{NN-2SS-E-N}\binom{W}{*}-\binom{*}{S}$ |
| $\mathrm{wt}_{\widetilde{V}H}$ | $\binom{NN}{*} = \frac{-1}{xz}\left(q^{SS-NN}\binom{*}{W}+q^{N+NN+E}\binom{S}{*}\right)$ | $\binom{*}{SS} = q^{NN-SS}\binom{W}{*}-q^{N+SS+E}\binom{*}{S}$ |
| $\mathrm{wt}_{\widetilde{H}V}$ | $\binom{NN}{*} = \frac{-1}{yw}\left(q^{SS-NN}\binom{*}{E}-q^{N+NN+W}\binom{S}{*}\right)$ | $\binom{*}{SS} = q^{NN-SS}\binom{E}{*}+q^{N+SS+W}\binom{*}{S}$ |
| $\mathrm{wt}_{\widetilde{V}V}$ | $\binom{NN}{*} = \frac{1}{yz}\left(\binom{*}{N}+q^{2N+2SS+E+W}\binom{*}{NN}\right)$ | $\binom{*}{SS} = \binom{N}{*}+q^{2N+2NN+E+W}\binom{SS}{*}$ |

Figure 8.6: The weights for a "pre-fused" NN/SS vertex in an $R^{(n)}$ vertex swapping rows $i, j$, where row $i$ is in the alternate model and row $j$ is in the original model. Note that the coefficients refer to the number of vertices of a given type in their attached term, not the original vertex.

also be interesting to consider whether there is a generalization of the "infinite source" style of model used in [BMP21] that would apply to superLLT polynomials, in effect collecting all non-positive columns into one column on which infinitely many particles are allowed to travel.

In the case of the Pieri identity, the pre-fusion vertices discussed in Section 8 will not occur, so it is an ideal starting point; in contrast, using this method to prove the Cauchy identity, pre-fusion vertices arise and require careful bookkeeping of all possible combinations of vertex types on each copy of the braid. For this section, the particle interpretation of the lattice model more clearly illustrates the comparisons between the original and alternate models.

**Lemma 9.1.** *Thinking of a Cauchy braid of mixed YBEs of either type $\widetilde{A}B$ or type $A\widetilde{B}$ in terms of particles passing through the lattice from bottom to top, the number of particles in the system must remain constant. That is, for the $n$-strand Cauchy braid given by*
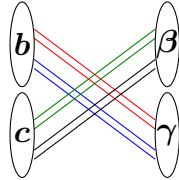


*we must have $\#\{particles\ entering\ from\ \boldsymbol{c}\ or\ \boldsymbol{\gamma}\} = \#\{particles\ exiting\ from\ \boldsymbol{b}\ or\ \boldsymbol{\beta}\}$.*

*Proof.* Considering first the case of type $\widetilde{A}B$, color the six diagonal vertices with particles according to the rule established for rows: particles travel along left arrows on original model (i.e., $B$) strands and right arrows on alternate model (i.e., $\widetilde{A}$) strands. Recall that the first label (i.e., $\widetilde{A}$ here) is the SW-NE strand and the second (i.e., $B$ here) is the SE-NW strand.
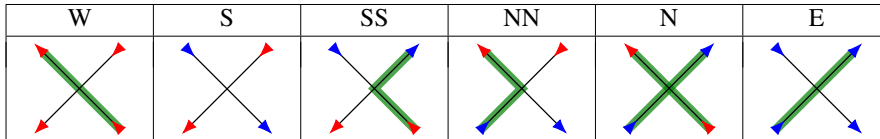


Since each individual vertex obeys the desired condition, a Cauchy braid built out of admissible vertices will as well. The statement follows similarly for type $A\widetilde{B}$.                          □
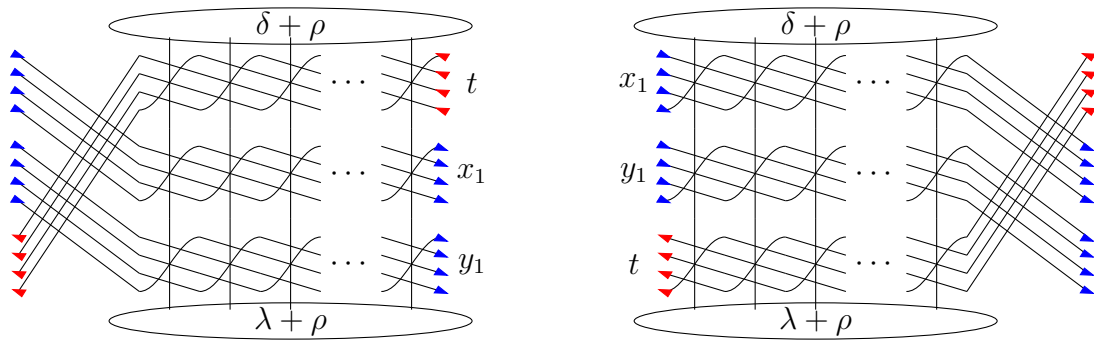
Figure 9.2: The Cauchy models used in Proposition 9.2 for $X = \{x_1\}, Y = \{y_1\}$ and $n = 4$.

For ease of reference during the following proofs, we reprint the ribbon lattice vertices with their particle interpretations for each type of model.



| Label | SW | NS | SE | NW | EW | NE |
|---|---|---|---|---|---|---|
| Original Model Particle Version | | | $wt_H = 0$ | | | $wt_V = 0$ |
| Alternate Model Particle Version | $wt_{\widetilde{H}} = 0$ | | | $wt_{\widetilde{V}} = 0$ | | |

Figure 9.1: The six ribbon vertices with their particle interpretations for original model and alternate model rows, noting which vertices vanish for each type of row.

Recall the generating functions $H(t)$ and $E(t)$ from Section 4: for $a_i = 1 + q^{2i} + \cdots + q^{2i(n-1)}$,

$$H(X;t) = \prod_i \prod_{k=0}^{n-1} \frac{1}{(1 - q^{2k}x_i t)} \text{ and } E(X;t) = \prod_i \prod_{k=0}^{n-1} (1 - q^{2k}x_i t).$$

**Theorem 9.2.** *(Lattice Model Pieri Rule) Given alphabets $X, Y$, we have that*

$$E(Y;t) \cdot Z\left(\mathcal{C}_{\delta,\lambda,I_B}(X/Y, \{t\}/\varnothing)\right) = \left(H(X;t)\right)^{-1} Z\left(\mathcal{C}_{\delta,\lambda,I_A}(X/Y, \{t\}/\varnothing)\right),$$
$$E(X;-t)H(Y;-t) \cdot Z\left(\mathcal{C}_{\delta,\lambda,I_B}(X/Y, \varnothing/\{t\})\right) = Z\left(\mathcal{C}_{\delta,\lambda,I_A}(X/Y, \varnothing/\{t\})\right).$$

Combining Theorem 9.2 with Proposition 8.5 gives a lattice model proof of Corollary 4.10.

*Proof.* For the right hand side of the first statement, consider the lattice model formed by attaching a Cauchy braid onto the right side of $\mathcal{C}_{\delta,\lambda,I_A}(X/Y, \{t\}/\varnothing)$, which has a single alternate model horizontal strip row (with parameter $t$) below standard model rows for $X/Y$. (See the right hand

side of Figure 9.2 for reference.) Consider all the possible states of this system: first, if no particles pass through the braid, all of its vertices are of type $S$, giving weight $\prod_{i,j} \prod_{k=0}^{n-1} (1 - q^{2k} x_i t) = (H(X;t))^{-1}$, and the rest of the lattice has boundary conditions $\mathcal{C}_{\delta,\lambda,I_A}$.

For the remaining states, since particles only travel up and right on alternate model rows, the only way for particles to enter the Cauchy braid and create vertices other than type S is by travelling rightwards through the infinite void of particles on row $t$. States in this case will garner an infinite number of factors of $t$ and may be thought of as a $\mu \to \infty$ term in the sum. Since we will take the $t^k$-th coefficient at the end to recover the Pieri rule, this term will not contribute to that identity for any finite $k$.

On the other hand, by Theorem 8.6, the partition function of this system is equal to that obtained by passing the Cauchy braid through to the left side, i.e. that with a Cauchy braid attached on the left hand side of $\mathcal{C}_{\delta,\lambda,I_B}(X/Y, \{t\}/\varnothing)$, where the alternate model horizontal strip row is the top row. In this case, particles cannot travel infinitely on row $t$, since type SW vertices have weight $0$ and such a particle would cross one of the particles in the infinite sea on the left hand side. Thus, particles travelling into the braid can only come from the leftmost $n$ columns of the infinite sea of particles. The particles in these first $n$ columns will travel either straight up or up through the braid, so chopping off these first $n$ columns leaves boundary conditions $\mathcal{C}_{\delta,\lambda,I_B}$ for the remaining portion of the state.

It remains to show that the partition function of these first $n$ columns and the Cauchy braid equals $E(Y;t)$. (The reader may wish to refer to Appendix D, which gives all states for this subsystem for $n = 2, X = \{x_1\}, Y = \{y_1\}$.) In the remainder of the proof, we will renumber these columns as $1, \ldots, n$ for simplicity; we hope that this will not cause any confusion with the original indexing. We will also generally split calculations into to the weight of the braid and the weight of the columns, for readability.

We proceed by induction on the number of variables in $Y$, since $E(Y;t)$ is multiplicative. For each step of this process, group states of the lattice model by the number of particles in the braid and their entry rows. The boundary conditions restrict vertices in the braid to types S (no particle), SS (particle turns northeast), and E (particle travels straight northeast), so once a particle encounters a type SS vertex, it must travel straight out of the braid. Notably, there cannot be any type NN vertices, so the pre-fusion phenomena will not occur. There are thus $(|X| + |Y| + 1)^n$ possible fillings of this portion of the lattice model, as each particle can enter the braid on any horizontal strip row in $X$, any vertical strip row in $Y$, or eschew the braid and travel straight up. By Proposition 2.10, assume without loss of generality that the alphabet $A/A'$ is ordered $1 < 2 < \cdots < 1' < 2' < \ldots$, so the spectral parameters read $t, x_1, x_2, \ldots, y_1, y_2, \ldots$ from top to bottom.

For the base case, let $Y = \{y_1\}$. If all $n$ particles enter the braid on row $y_1$ (see Figure 9.3), the $R$-matrix $R_{y_1,t}$ where this row crosses the $\widetilde{H}$ row will be all type $SS$ vertices, all other $R$-vertices will be all type E, and the particles must all travel out across row $t$, garnering powers of $q$ along the way. Thus, this state of the subsystem has weight $\prod_{k=0}^{n-1} \left( q^{2k}(-y_1 t) \right)$, as

$$\mathrm{wt}(braid) = q^{n-1}(-y_1 t) \cdots q^1(-y_1 t) \cdot (-y_1 t)$$
$$\mathrm{wt}(columns) = q^{n-1} \cdots q^1 \cdot 1.$$

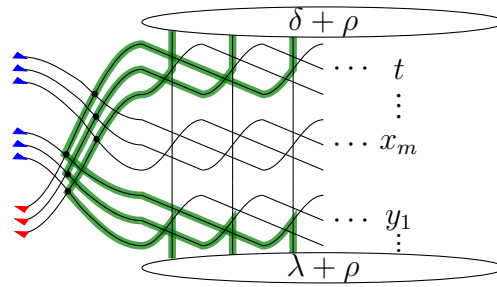This state gives the highest order term of $E(\{y_1\}; t)$. Accordingly, states in which $m$ particles

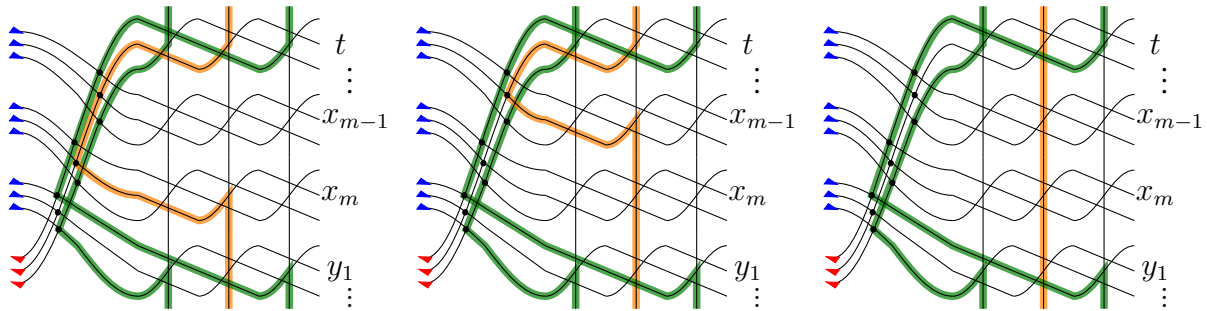Figure 9.3: The state for the $n = 3$ base case in which all particles travel through the braid.



Figure 9.4: For $n = 3$, the process of moving particle $i = 2$ up through the braid. The weight of the leftmost state swaps a power of $(-y_1 t)$ from that of Figure 9.3 for a factor of $q^4 x_m t$. The second then swaps the $q^4 x_m t$ for a factor of $q^4(1 - q^6 x_m t)x_{m-1}t$. However, the last term swaps this for $q^4(1 - x_m t - x_{m-1}t + q^6 x_m t x_{m-1}t)$, so adding them together, only the $q^4 \cdot 1$ term survives.

enter the braid on row $y_1$ will give the terms with powers $(y_1 t)^{n-m}$: starting with the $n-1$-st powers, we examine the effect of moving the entry point of one particle up row by row. If the $i$-th particle (i.e., the one entering on column $i$) exits through the last horizontal strip row $x_m$ instead (see the leftmost state in Figure 9.4), the the $R$-matrix $R_{y_1,t}$ will have a type S vertex in the $(n-i)$-th position instead, and $R_{x_m,t}$ will have a type SS vertex in the $(n-i)$-th place, which interacts with the type E vertices in the rest of $R_{x_m,t}$. Also, the $i$-th vertex in row $y_1$ changes to type SW, so the result is

$$\mathrm{wt}(braid) = \left(q^{n-2}(-y_1 t) \cdots q^1(-y_1 t) \cdot (-y_1 t)\right) \cdot \left(q^{n-1}(x_m t) \cdot q^{n-i}\right)$$
$$\mathrm{wt}(columns) = q^{n-i} \cdot \left(q^{n-1} \cdots q^1 \cdot 1\right)$$

That is, this change trades one factor of $(-y_1 t)$ for $q^{2n-2i}x_m t$.

Continuing to move the braid entry point for this particle up row by row, referencing Figure 9.4, we swap a factor of $x_k t$ for one of $(1 - q^{2n-2}x_k t)x_{k-1}t$ at each step until we eventually reach the

state in which the $i$-th particle travels straight up. The weight of this state is

$$q^{2n-2i}\left(\prod_j(1-q^{2n-2}x_jt)\right)\prod_{k=0}^{n-2}q^{2k}(-y_1t)$$

which cancels off all the $x_jt$ terms obtained by previous states in which particle $i$ moved. Thus, the collective terms give a factor of $q^{2n-2i}\cdot\prod_{k=0}^{n-2}\left(q^{2k}(-y_1t)\right)$. Ranging over all $i$ accounts for each of the $(n-1)$-st powers of $(-y_1t)$ appearing in $E(\{y_1\};t)$.

Similarly, states in which $m$ particles move up from row $y_1$ at a time give all of the $(n-m)$-th powers of $(-y_1t)$ appearing in $E(\{y_1\},t)$. Summing over $m$ proves the base case.

Suppose that the inductive hypothesis holds for all $i < k$ for some $k$. To incorporate an extra row $y_{k+1}$, move particles up from row $k+1$ using the same careful bookkeeping as in the base case. Referring to Figure 9.5, we see that the weight of the case where all $n$ particles enter on row $k+1$, is

$$\left(\prod_{k=0}^{n-1}\left(q^{2k}(-y_{k+1}t)\right)\right)\cdot E(\{y_1,\ldots,y_k\};t),$$

where the factor of $E(\{y_k\},t)$ arises from the $R_{y_k,t}$ vertex. As in the base case, moving $m$ particles up at a time gives us all the terms with a factor of $(y_kt)^{n-m}$, each multiplied by $E(\{y_1,\ldots,y_k\};t)$. Summing over all possible states then completes the proof of the first statement.
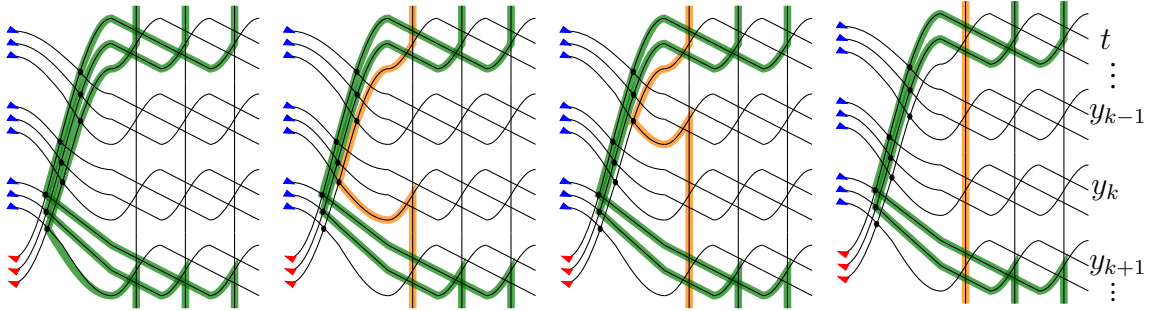


Figure 9.5: For $n = 3$, the leftmost figure gives the case where all particles travel into the braid on row $y_{k+1}$. The remaining three states illustrate the process of moving particle $i = 1$ up from row $y_{k+1}$, and their weights sum to $q^{2n-2}(y_{k+1}t)^{n-1}E(\{y_1,\ldots,y_k\};t)$. The spectral parameters for all four figures are listed at the far right.

For the second statement, consider the similar lattice model in which the alternate model row is a vertical strip row with parameter $t$. As this process is entirely analogous to the proof above, we note the main differences and leave the details for the reader.

Starting with the braid on the right as before, there cannot be any particles travelling through the infinite void into the braid, since NW vertices have weight $0$ for type $\widetilde{V}$ vertices, so the partition function of this side is $\mathcal{C}_{\delta,\lambda,I_A}(X/Y,\varnothing/\{t\})$.

Swapping the braid to the left hand side using the Yang–Baxter equation, the particle version of the lattice admits the following interpretation: particles from the infinite sea can travel either straight up or first left, then up, then right on the $\widetilde{V}$ row back to their starting column. (There are other ways of interpreting the particle movement in each state, but we prefer this one for clarity as to which factors arise in the weight.) However, since $\widetilde{V}$ allows the SW vertex, particles can now travel arbitrarily far on $V$ rows from within the sea to reach the braid, and once a particle has moved into the braid, it leaves room for subsequent particles to move left onto the column it has vacated. In any state, at most finitely many particles can move; cutting off the portion of the infinite sea with moving particles leaves behind the boundary conditions for $\mathcal{C}_{\delta,\lambda,I_B}(X/Y, \varnothing/\{t\})$ in each case.

In these states, a particle that moves left one step of $n$ columns (where travelling up through the braid counts as a complete step) from a vertical strip row $y_j$ gains a factor of $-q^{2k}y_jt$, where $k$ is the number of columns with unmoving particles that have been passed by this particle. Carefully indexing all possibilities of moving particles and the number of unmoving columns between them, it is possible to obtain any weight of the form $(-y_jt)^{m_1}(-q^2y_jt)^{m_2}\cdots(-q^{2n-2}y_jt)^{m_{n-1}}$, so states in which particles only move on this row collect to give weight $\prod_{k=0}^{n-1}(1+q^{2k}y_jt)^{-1} = H(y_j; -t)$. (See Figure 9.6 for examples when $n = 3$.)
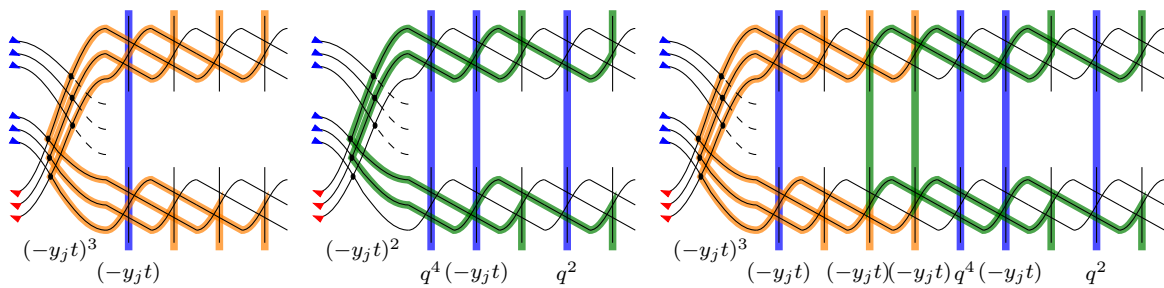


Figure 9.6: For $n = 3$, the states with weights $(-y_jt)^4$, $(-q^2y_jt)^3$, and $(-y_jt)^4(-q^2y_jt)^3$, respectively, where only rows $y_j$ and $t$ are shown and the label below each column is its weight.

Starting from the bottom row $y_\ell$, note that a particle on a higher row can only move left if the particle $n$ steps to its left has already moved. (See Figure 9.7 for an example of how the weights interact in this case for two original model vertical strip rows.) Furthermore, since the SW vertex has weight $0$ on original horizontal strip rows, at most $n$ particles can move one step of $n$ columns each on these rows, gaining at most one factor each of $x_it, q^2x_it, \ldots, q^{2(n-1)}x_it$. Applying a similar induction to that of the first statement, using the fact that both $E(X; -t)$ and $H(Y; t)$ are multiplicative, gives that the partition function on this side is

$$H(Y; -t)E(X; -t) \cdot \mathcal{C}_{\delta,\lambda,I_B}(X/Y, \varnothing/\{t\}). \qquad \square$$

*Remark* 9.3. This setup for the Cauchy model gives the Pieri identities of Theorem 4.8 for the function $F_{\lambda/\delta}$, but an analogous proof passing a single standard model strip through an alternate model would give the analogous identities for $G_{\lambda/\delta}$. Similarly, for the Cauchy identity, the braid may be attached to either side: here, we find it more convenient to attach on the other side.
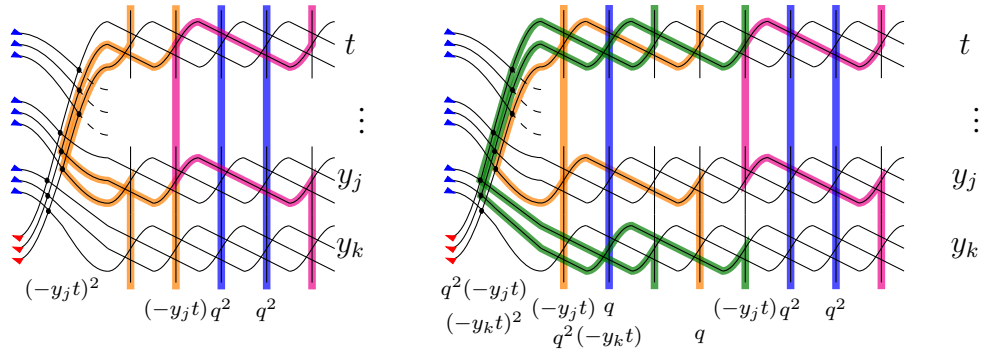
Figure 9.7: For $n = 3$, the states with weight $(-y_k t)^2(-q^4 y_k t)$ (left hand side) and $(-q^2 y_k t)^3(-y_j t)^2(-q^4 y_j t)$ (right hand side), assuming $j < k$. Note that the row $y_k$ moving particles on the right hand state are garnering the same factors as those in the middle state on Figure 9.6.
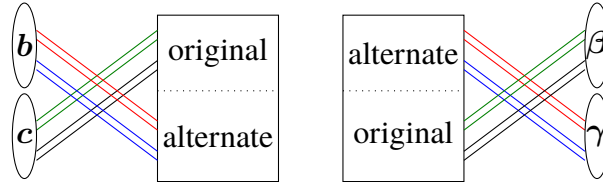


Figure 9.8: Braids of $R^{(n)}$ vertices attached to the Cauchy model; note the change of side from Figure 9.2 and that both types of models may have more than one row.

**Conjecture 9.4.** [Lattice Model Cauchy Identity] Set $\boldsymbol{\beta}, \boldsymbol{c} = $ "right" and $\boldsymbol{\gamma}, \boldsymbol{b} = $ "left". Attaching a braid to the right hand side of the Cauchy model with ordering $I_B$ (as in the right hand figure of Figure 9.8) gives the partition function

$$\prod_{i,j,k,\ell} \prod_{t=0}^{n-1} \frac{(1 - q^{2t} x_i z_\ell)(1 - q^{2t} y_j w_k)}{(1 - q^{2t} x_i w_k)} \cdot \sum_\lambda \mathcal{G}_{\mu/\lambda}(X/Y; q) \cdot \mathcal{G}_{\nu/\lambda}(W/Z; q),$$

By the Yang–Baxter equation, this expression is equal to the partition function of the system obtained by attaching a braid to the left hand side of the Cauchy model with ordering $I_A$, which is

$$\left( \prod_{j,\ell} \prod_{t=0}^{n-1} (1 - q^{2t} y_j z_\ell) \cdot \sum_\lambda \mathcal{G}_{\lambda/\mu}(X/Y; q) \cdot \mathcal{G}_{\lambda/\nu}(W/Z; q) \right) + O(Y^\infty)O(Z^\infty),$$

where $O(Y^\infty)$ denotes terms with infinitely many powers of spectral parameters in the set $Y$.

*Remark* 9.5. Setting $\boldsymbol{\beta}, \boldsymbol{c} = $ "left" and $\boldsymbol{\gamma}, \boldsymbol{b} = $ "right" and swapping sides on which the braid attaches (i.e. using type $\widetilde{A}B$ instead of type $A\widetilde{B}$ R-vertices) produces an analogous result.

The latter statement is actually fairly straightforward to prove, using the same techniques as in the proof of Theorem 9.2. If no particles pass through the Cauchy braid, it has weight

$\prod_{j,\ell} \prod_{t=0}^{n-1}(1 - q^{2t}y_j z_\ell)$ and the remaining piece of the lattice has boundary conditions $\mathcal{C}_{\mu,\nu,I_B}$, so Proposition 8.5 gives the first term. On the other hand, suppose a particle enters the Cauchy braid: since all of the particles entering from boundary conditions are required to travel away from the braid in the alternate model, we may think of this particle as originating in the braid, travelling right through the alternate model, up into the original model, then back left into the braid to create a loop. Since columns can only carry one particle at a time, this particle must travel through the infinite sea in the alternate model before it can travel up into the original and back through; however, horizontal strip rows have weight 0 on SE (respectively SW) vertices in the original (respectively alternate) model, so infinitely many of these steps must occur in vertical strip rows, amassing a weight of $O(Y^\infty)$ in the alternate model sea and $O(Z^\infty)$ in the original model sea.

The first claim is the more difficult, because it requires precisely evaluating the Cauchy braid in circumstances where there may be particles travelling through the braid (in fact, looping as in the previous case), taking into account the phenomena of pre-fusion as well. As the bookkeeping for this process becomes significantly more complicated than that of Theorem 9.2, we will not embark on that journey in this paper. However, note that excepting this difficulty, cutting off the loops in any state on this side and considering the remaining boundary conditions, we see that they match $\mathcal{C}_{\mu,\nu,I_A}$, so again Proposition 8.5 gives the desired sum over superLLT polynomials.

*Remark* 9.6. For circumstances where we only need finitely many columns to represent all possible $\lambda$ that could appear on the boundary (i.e., when the original model is all horizontal strip rows and the alternate all vertical strip rows, or vice versa), the error terms in $O(Y^\infty)O(Z^\infty)$ appearing in Conjecture 9.4 are zero, so truncating this Cauchy lattice model recovers precisely the dual Cauchy identity for LLT polynomials discussed in Section 4.

## Acknowledgements

## References

[ABPW21]    Amol Aggarwal, Alexei Borodin, Leonid Petrov, and Michael Wheeler. Free Fermion Six Vertex Model: Symmetric Functions and Random Domino Tilings. 2021. `arXiv:2109.06718`.

[ABW21]    Amol Aggarwal, Alexei Borodin, and Michael Wheeler. Colored Fermionic Vertex Models and Symmetric Functions. 2021. `arXiv:2101.01605`.

[Bax89]    Rodney Baxter. *Exactly Solved Models in Statistical Mechanics*. Academic Press Limited, 1989.

[BBBG19]   Ben Brubaker, Valentin Buciumas, Daniel Bump, and Nathan Gray. A Yang–Baxter equation for metaplectic ice. *Commun. Number Theory Phys.*, 13(1):101–148, 2019. `doi:10.4310/CNTP.2019.v13.n1.a4`.

[BBBG20]   Ben Brubaker, Valentin Buciumas, Daniel Bump, and Henrik P. A. Gustafsson. Metaplectic Iwahori Whittaker functions and supersymmetric lattice models. 2020. `arXiv:2012.15778`.

[BBBG21a]  Ben Brubaker, Valentin Buciumas, Daniel Bump, and Henrik P. A. Gustafsson. Colored five-vertex models and Demazure atoms. *J. Combin. Theory Ser. A*, 178:Paper No. 105354, 48, 2021. `doi:10.1016/j.jcta.2020.105354`.

[BBBG21b]  Ben Brubaker, Valentin Buciumas, Daniel Bump, and Henrik P. A. Gustafsson. Iwahori-metaplectic duality. 2021. `arXiv:2112.14670`.

[BBF11]    Ben Brubaker, Daniel Bump, and Solomon Friedberg. Schur polynomials and the Yang-Baxter equation. *Comm. Math. Phys.*, 308(2):281–301, 2011. `doi: 10.1007/s00220-011-1345-3`.

[BMN14]    Daniel Bump, Peter J. McNamara, and Maki Nakasuji. Factorial Schur functions and the Yang-Baxter equation. *Comment. Math. Univ. St. Pauli*, 63(1-2):23–45, 2014.

[BMP21]    Alexey Bufetov, Matteo Mucciconi, and Leonid Petrov. Yang-Baxter random fields and stochastic vertex models. *Adv. Math.*, 388:Paper No. 107865, 94, 2021. `doi:10.1016/j.aim.2021.107865`.

[BW20]     Alexei Borodin and Michael Wheeler. Observables of coloured stochastic vertex models and their polymer limits. *Probab. Math. Phys.*, 1(1):205–265, 2020. `doi: 10.2140/pmp.2020.1.205`.

[CGKM22]   Sylvie Corteel, Andrew Gitlin, David Keating, and Jeremy Meza. A vertex model for LLT polynomials. *Int. Math. Res. Not. IMRN*, (20):15869–15931, 2022. `doi: 10.1093/imrn/rnab165`.

[CW12]     Shun-Jen Cheng and Weiqiang Wang. *Dualities and representations of Lie superalgebras*, volume 144 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2012. `doi:10.1090/gsm/144`.

[CYWZZ19]  Michael Curran, Calvin Yost-Wolff, Sylvester Zhang, and Valerie Zhang. Ribbon lattices and ribbon function identities. REU report, 2019.

[Fre20]    Claire Frechette. Yang-Baxter Equations for General Metaplectic Ice. 2020. `arXiv:2009.13669`.

[GK21]     Andrew Gitlin and David Keating. A Vertex Model for Supersymmetric LLT Polynomials. 2021. `arXiv:2110.10273`.

[Gra17]    Nathan Tyler Gray. *Metaplectic Ice for Cartan Type C*. PhD thesis, University of Minnesota, 2017. URL: `https://hdl.handle.net/11299/190563`.

[Har21]    Andrew Hardt. Lattice Models, Hamiltonian Operators, and Symmetric Functions. 2021. `arXiv:2109.14597`.

[KRS81]   P. P. Kulish, N. Yu. Reshetikhin, and E. K. Sklyanin. Yang-Baxter equations and representation theory. I. *Lett. Math. Phys.*, 5(5):393–403, 1981. `doi:10.1007/BF02285311`.

[Lam05]   Thomas Lam. Ribbon tableaux and the Heisenberg algebra. *Math. Z.*, 250(3):685–710, 2005. `doi:10.1007/s00209-005-0771-3`.

[Lam06]   Thomas Lam. A combinatorial generalization of the boson-fermion correspondence. *Math. Res. Lett.*, 13(2-3):377–392, 2006. `doi:10.4310/MRL.2006.v13.n3.a4`.

[LLT97]   Alain Lascoux, Bernard Leclerc, and Jean-Yves Thibon. Ribbon tableaux, Hall-Littlewood functions, quantum affine algebras, and unipotent varieties. *J. Math. Phys.*, 38(2):1041–1068, 1997. `doi:10.1063/1.531807`.

[SW85]    Dennis W. Stanton and Dennis E. White. A Schensted algorithm for rim hook tableaux. *J. Combin. Theory Ser. A*, 40(2):211–247, 1985. `doi:10.1016/0097-3165(85)90088-3`.

# A. Reduced Yang–Baxter equation

After eliminating the $0$-weighted terms, the Yang–Baxter Equation can be simplified to the following system of equations.



(A.1)          (A.2)          (A.3)

(A.4)          (A.5)          (A.6)

(A.7)          (A.8)          (A.9)
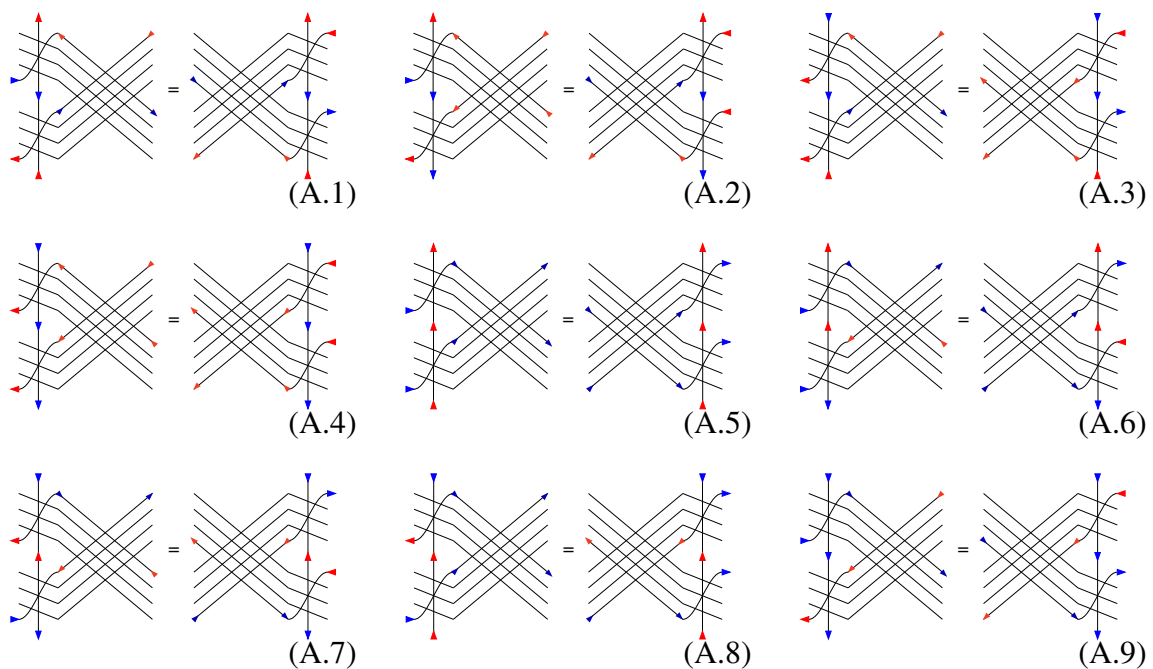
$$(\text{A.10})$$



$$(\text{A.11}) \qquad\qquad (\text{A.12})$$



$$(\text{A.13}) \qquad\qquad (\text{A.14})$$

## B. Reduced Yang–Baxter equation for column weights

When considering the YBE for the vertical strip weights, boundary conditions (A.2), (A.3), (A.4), (A.9), (A.12), and (A.14) each contain a NE vertex and thus have weight $0$ on both sides. They are replaced by (B.1), (B.2), (B.3), (B.4), (B.5), (B.7), which each contain a SE vertex and had weight $0$ for the horizontal strip weights. Also, conditions (A.6) and (A.8) acquire an extra state in this case (see (B.6) and (B.8), respectively).



$$(\text{B.1}) \qquad\qquad (\text{B.2}) \qquad\qquad (\text{B.3})$$



$$(\text{B.4})$$

$$(\text{B.5})$$



$$(\text{B.6})$$



$$(\text{B.7})$$



$$(\text{B.8})$$

## C. $R^{(1)}$ vertex weights for Cauchy mixed YBES of type $A\widetilde{B}$

Let $\kappa = \#NN\downarrow$ below $+ \#SS$ above, and let $\tau = \#SS$ below $+ \#NN$ above.

| Label | $\text{wt}_{H\widetilde{H}}$ | $\text{wt}_{V\widetilde{H}}$ | $\text{wt}_{H\widetilde{V}}$ | $\text{wt}_{V\widetilde{V}}$ |
|---|---|---|---|---|
| W | $q^{-\tau}$ | $(1-q^{2n-2-2\#W\downarrow}xy)q^{-\tau}$ | $q^{\kappa-\#E}$ | $q^{-\tau}$ |
| S | $-q^{1-n+2\kappa}(q^{2n-2+2\#S\downarrow}xy-1)$ | $-q^{1-n+2\kappa+\#N}$ | $-q^{1-n+2\kappa+\#N}$ | $-q^{1-n+\kappa+\#N-\tau}$ |
| SS* | $q^{n-1-\#NN\uparrow}$ | $q^{\#SS\uparrow}$ | $q^{\#SS\uparrow}$ | $q^{-\#NN\uparrow}$ |
| NN* | $xy \cdot q^{n-1-\#SS\downarrow}$ | $-xy \cdot q^{\#NN\downarrow}$ | $-xy \cdot q^{\#NN\downarrow}$ | $xy \cdot q^{-\#SS\downarrow}$ |
| N | $q^{\kappa-\tau+\#S}$ | $q^{2\kappa}$ | $q^{2\kappa}$ | $(1-q^{2\#N\downarrow}xy)q^{2\kappa}$ |
| E | $q^{-\tau-\#W}$ | $q^{\kappa-\#W}$ | $(1-q^{2n-2-2\#E\downarrow}xy)q^{-\tau}$ | $q^{\tau-\#W}$ |

| Type | Top SS Recursion | Bottom NN Recursion |
|---|---|---|
| $\text{wt}_{H\widetilde{H}}$ | $\binom{SS}{*} = \frac{1}{xw}\left(q^{-2SS-2S-E-W}\binom{*}{SS}+q^{2NN-2SS}\binom{*}{S}\right)$ | $\binom{*}{NN} = \frac{q^{-2NN-2S-E-W}}{xw}\binom{NN}{*}+q^{2SS-2NN}\binom{S}{*}$ |
| $\text{wt}_{V\widetilde{H}}$ | $\binom{SS}{*} = \frac{-1}{wy}\left(q^{NN-SS}\binom{*}{W}-q^{-SS-E-S}\binom{N}{*}\right)$ | $\binom{*}{NN}q^{SS-NN}\binom{W}{*}-q^{-NN-S-E}\binom{*}{N}$ |
| $\text{wt}_{H\widetilde{V}}$ | $\binom{SS}{*} = \frac{-1}{xz}\left(q^{NN-SS}\binom{*}{E}-q^{-SS-W-S}\binom{N}{*}\right)$ | $\binom{*}{NN} = q^{SS-NN}\binom{E}{*}-q^{-NN-S-W}\binom{*}{N}$ |
| $\text{wt}_{V\widetilde{V}}$ | $\binom{SS}{*} = \frac{1}{yz}\left(q^{E+NN+S}\binom{*}{W}-\binom{N}{*}\right)$ | $\binom{*}{NN} = -\binom{*}{N}+q^{SS+W+S}\binom{E}{*}$ |

## D. Fillings of the left-hand Pieri Cauchy braid and first $n$ columns for $n = 2$, $X = \{x_1\}, Y = \{y_1\}$, each with their weights.



weight = $(1 - xt)(1 - q^2xt)$



$(xt)(1 - q^2xt)$



$q^2(xt)(1 - q^2xt)$



$q^4(xt)^2$



$(1 - q^2xt)(-yt)$



$q^2(xt)(-yt)$



$q^2(1 - q^2xt)(-yt)$



$q^4(xt)(-yt)$



$q^2(yt)^2$