MULTILINE QUEUES WITH SPECTRAL PARAMETERS [DETAILED VERSION]

ERIK AAS, DARIJ GRINBERG, AND TRAVIS SCRIMSHAW

Abstract. Using the description of multiline queues as functions on words, we introduce the notion of a spectral weight of a word by defining a new weighting on multiline queues. We show that the spectral weight of a word is invariant under a natural action of the symmetric group, giving a proof of the commutativity conjecture of Arita, Ayyer, Mallick, and Prolhac. We give a determinant formula for the spectral weight of a word, which gives a proof of a conjecture of the first author and Linusson.

Contents

Date: October 18, 2018.

2010 *Mathematics Subject Classification.* 60C05, 05A19, 16T25, 05E05.

Key words and phrases. multiline queue, TASEP, R-matrix, symmetric function.

TS was partially supported by the Australian Research Council DP170102648 and the National Science Foundation RTG grant DMS-1148634.

1. Introduction

One of the fundamental models of particles moving in a 1-dimensional lattice is the asymmetric simple exclusion process (ASEP), and it has received broad attention in many different variations. The earliest known publication of the ASEP was done to model the dynamics of ribosomes along RNA [MGP68]. For statistical mechanics, it is a model for gas particles in a lattice with an induced current, where the exclusion mimics the short-range interactions among the particles. Despite admitting very simple descriptions of the particle dynamics, the ASEP has very rich macroscopic behaviors, such as

- boundary-induced phase transitions [Kru91],
- spontaneous symmetry breaking with possibly multiple broken symmetry phases [AHR98, AHR99, CEM01, EFGM95, EPSZ05, GLE+95, PK07],
- describing the formations of shocks [DJLS93, Fer92, FF94a, FF94b, Lig76], and
- phase separation and condensation [EKKM98, JNH+09, KLM+02, RSS00].

We also refer the reader to [PEM09, Sch01, SZ95, TJHJ17] and references therein.

The term exclusion process was coined by Spitzer [Spi70], where he was focused on an application with Brownian motion with hard-core interactions. Moreover, it was [Spi70] that initiated the investigation of exclusion processes using probability theory. However, the applications of the ASEP (and its variations) has since spread to other areas, such as

- transportation processes in capillary vessels [Lev73] or proteins within the cells along actin filaments [KNL05],
- anistropic conductors known as solid electrolytes [CL99],
- discrete models of traffic flow [Sch00],
- partition growth processes [Lam15],
- (non)symmetric Macdonald polynomials, Koornwinder polynomials, and (deformed) Knizhnik–Zamolodchikov (KZ) equations [CdGW15, KT07],
- random matrix theory [Joh00, TW09], and
- moments of Askey–Wilson polynomials [CW11, USW04].

If we prohibit the particles from moving backwards, we obtain the totally asymmetric exclusion process (TASEP), a non-equilibrium stochastic process that has its own vast literature. For example, we refer the reader to [AL18, AAMP11, BE07, BP14, DEHP93, KMO15, KMO16b, Lig99] and references therein. In this paper, we consider the TASEP on a ring with n sites and ℓ species of particles. Thus, we will consider the states to be words u in the alphabet $\{1, \dots, \ell\}$ of length n, where we take the indices to be $\mathbb{Z}/n\mathbb{Z}$. We will also consider our process to be discrete in time, where our transition map interchanges a pair $u_i u_{i+1}$ with $u_i > u_{i+1}$ to $u_{i+1} u_i$ and is done at a uniform rate.

The steady state of the TASEP on a ring is known in terms of another process using ordinary multiline queues (MLQs) and applying the Ferrari–Martin (FM) algorithm [FM06, FM07]. This is a generalization of 2-line queues used by Angel [Ang06] and the work of Ferrari, Fontes, and Kohayakawa [FFK94].

In [KMO15, KMO16b], the FM algorithm was reformulated in terms of the combinatorial R-matrix [NY97, Shi02] and using type $A_{n-1}^{(1)}$ Kirillov–Reshetikhin crystals [KKM+92]. This interpretation gives a connection with five-vertex models, corner transfer matrices [Bax89], 3D integrable lattice models, and the tetrahedron equation [Zam80], yielding a matrix product formula for the steady state distribution different than [CdGW15, EFM09, PEM09].

In this paper, we introduce a new weighting of MLQs, which is the weight of the MLQ considered as a tensor product of Kirillov–Reshetikhin crystals. We also interpret MLQs as functions on words of a fixed length n following [AAMP11], where it was referred to as the generalized FM algorithm. This allows us to define the spectral weight or amplitude of a word u to be the sum over all the weight of all ordinary MLQs $\mathbf{q}$ such that $u = \mathbf{q}(1^n)$, where 1^n is the word $1 \cdots 1$. Furthermore, we introduce the notion of a σ-twisted MLQ, where σ is a permutation, although this is implicitly considered in [AAMP11]. Our main result (Theorem 3.1) is that for a fixed permutation σ, the sum of the weights of all σ-twisted MLQs $\mathbf{q}_\sigma$ such that $u = \mathbf{q}_\sigma(1^n)$ equals the spectral weight of u. To this end, we construct an action of the symmetric group on MLQs that corresponds, under the usual FM algorithm, to the natural action by letters on words. We show that does not change the MLQ as a function on words. This action has previously appeared in a number of different guises, such as in Danilov and Koshevoy [DK05] (see also [Gor17, Ch. 4]), van Leeuwen [vL06, Lemma 2.3], and Lothaire [Lot02, Ch. 5, (5.6.3)]. In the context of Kirillov–Reshetikhin crystals, it can be described as applying a combinatorial R-matrix to an MLQ, where the weight remaining invariant is a condition of being a crystal isomorphism.

As a consequence of this action and specializing our weight parameters to 1, we obtain a proof of the commutativity conjecture of [AAMP11]. However, we note that the interlacing property of [AAMP11] does not generalize to our weighting of MLQs. Furthermore, we give a determinant expression for the spectral weight of decreasing words by using the Lindström–Gessel–Viennot Lemma [GV85, Lin73]. By combining these results, we obtain a proof of [AL18, Conj. 3.10], which in turn proves a number of other conjectures in [AL18].

We note that our weighting scheme can be extended to multiline process used to determine the steady state distribution of the totally asymmetric zero range process (TARZP) on a ring, where multiple particles can occupy the same site. This comes from the fact that the TARZP steady state distribution can also be computed using a tensor product of Kirillov–Reshetikhin crystals (under rank-level duality) using combinatorial R-matrices with analogous connections to corner transfer matrices and the tetrahedron equation [KMO16c, KMO16d]. Thus, we expect that a similar description of σ-twisted multiline process can be defined such that the weighting is invariant under the action of the combinatorial R-matrix. Yet it seems unlikely that our weighting is related to the steady state distribution for the inhomogeneous TASEP [AM13, AL14] or TARZP [KMO16a].

This paper is organized as follows. In Section 2, we give the necessary background and definitions of MLQs and spectral weight. In Section 3, we give our main results, and use them to prove some of the conjectures in [AL18]. In Section 4, we describe the connection between MLQs and the TASEP. In Section 5, we give a proof of our Jacobi-Trudi-type formula (Theorem 3.9). In Section 6, we give a proof of our main theorem (Theorem 3.1). In Section 7, we give some additional remarks about our results.

1.1. **Acknowledgements.** We thank Atsuo Kuniba for explaining the results in his papers [KMO15, KMO16a, KMO16b, KMO16c, KMO16d]. We thank Olya Mandelshtam for useful discussions on the inhomogeneous TASEP. We thank Jae-Hoon Kwon for pointing out that the $\mathfrak{S}_n$-action on MLQs comes from an $(\mathfrak{sl}_m \oplus \mathfrak{sl}_n)$-action. This work benefited from computations using SageMath [Sag18, SCc08].

2. Background and definitions

Fix a positive integer n. For a nonnegative integer k, let $[k]$ denote the set $\{1, 2, \dots, k\}$, and so $[0] = \emptyset$. Let $\mathfrak{S}_k$ denote the symmetric group on $[k]$, and let $s_i \in \mathfrak{S}_k$ be the simple transposition of i and $i+1$. Let $w_0 \in \mathfrak{S}_k$ be the longest element: the permutation $k(k-1)\cdots 321$ (written in one line notation) that reverses the order of all elements.

We shall refer to the elements $1, 2, \dots, n \in \mathbb{Z}/n\mathbb{Z}$ as *sites*. We visualize them as points on a line that "wraps around" cyclically; thus, for example, the sites weakly to the right of a site i are $i, i+1, \dots, n-1, n, 1, 2, 3, \dots$ (in this order).

2.1. Words and queues.

Let $\mathcal{W}_n$ be the set of words $u = u_1 \cdots u_n$ in the ordered alphabet $\mathcal{A} := \{1 < 2 < 3 < \cdots\}$. We consider the indices of letters in a word to be taken modulo n (that is, $u_{k+n} = u_k$ for all k). Thus, if u is a word and $i \in \mathbb{Z}/n\mathbb{Z}$ is a site, then the i-th letter u_i of u is well-defined. We sometimes refer to a letter $u_i = t$ as a *particle at site i of class t*.

The *type* of a word u is the vector $\mathbf{m} = (m_1, m_2, \dots)$, where m_i is the number of occurrences of i in u. Let $\ell = \max\{i \mid m_i \neq 0\}$, which we say is the number of *classes* in u or $\mathbf{m}$. A word u or type $\mathbf{m}$ with ℓ classes is *packed* if $m_i \neq 0$ for all $1 \leq i \leq \ell$. A word w of type $\mathbf{m}$ is *standard* if $m_i \leq 1$ for all i. We will write $1^n = 1 \cdots 1$ for the (unique) word of type $(n, 0, \dots)$.

We *merge* two adjacent classes $i, i+1$ in a word u to obtain a new word by replacing all occurrences of j by $j-1$ in u for each $j = i+1, i+2, \dots$ in that order. We denote the merging of i and $i+1$ in u by $\vee_i u$. Note that $\vee_i u$ is packed whenever u is packed. For $T = \{t_1 < \cdots < t_k\} \subseteq [\ell - 1]$, we set $\bigvee_T u := \vee_{t_1} \cdots \vee_{t_k} u$. Similarly, the merging of $i, i+1$ in a type $\mathbf{m} = (m_1, m_2, \dots)$ is $\vee_i(\mathbf{m}) = (m_1, \dots, m_{i-1}, m_i + m_{i+1}, m_{i+2}, \dots)$. Likewise, we define $\bigvee_T \mathbf{m}$ for a type $\mathbf{m}$. These operations interact as one would hope: If the type of a word u is $\mathbf{m}$, then the type of $\vee_i u$ is $\vee_i(\mathbf{m})$.

Fix a word $u \in \mathcal{W}_n$, and let $\mathbf{m} = (m_1, m_2, \dots)$ be the type of u. For each $i \geq 0$, set

$$p_i(\mathbf{m}) := m_1 + m_2 + \cdots + m_i. \tag{2.1}$$

When $\mathbf{m}$ is clear, we simply write p_i for this. (Thus, $p_0 = 0$ and $p_i = n$ for sufficiently large i.)

We define a *queue* to be any set of sites. A queue of size r will be called an *r-queue*.

We shall now define an action of queues on words. Namely, if q is any queue and $u \in \mathcal{W}_n$ is a word, then a new word $v = q(u) \in \mathcal{W}_n$ is defined by the following algorithm from [AAMP11, Sec. 4.5]: In the beginning, no letter of $v \in \mathcal{W}_n$ is set. Choose a permutation $(i_1, i_2, \dots, i_n)$ of $(1, 2, \dots, n)$ such that $u_{i_1} \leq$

$u_{i_2} \leq \cdots \leq u_{i_n}$. (The result of the algorithm will not depend on this choice, as we will show in Lemma 2.2 below.)

Phase I: For $i = i_n, i_{n-1}, \ldots, i_{|q|+1}$, do the following. Find the first site j weakly to the left (cyclically) of i such that $j \notin q$ and v_j is not set. Then set $v_j = u_i + 1$.

Phase II: For $i = i_1, i_2, \ldots, i_{|q|}$, do the following. Find the first site j weakly to the right (cyclically) of i such that $j \in q$ and v_j is not set. Then set $v_j = u_i$.

Remark 2.1. Consider the above algorithm. Notice that Phase I sets v_j for all $j \notin q$ (because it contains $n - |q|$ steps, and sets one such v_j per step), whereas Phase II sets v_j for all $j \in q$ (for similar reasons). Hence, at the end of the algorithm, all letters of v are set, and we never run out of j's in either phase.

Since Phase I only deals with $j \notin q$, and Phase II only with $j \in q$, the two phases can be arbitrarily interleaved (*i.e.*, we can perform the steps of the algorithm in any order as long as the steps of Phase I (resp. Phase II) are processed in the order $i = i_n, i_{n-1}, \ldots, i_{|q|+1}$ (resp. $i = i_1, i_2, \ldots, i_{|q|}$).

Lemma 2.2. *The resulting word $v = q(u)$ does not depend on the choice of permutation $(i_1, i_2, \ldots, i_n)$ (as long as $u_{i_1} \leq u_{i_2} \leq \cdots \leq u_{i_n}$ holds).*

Proof. Consider some $k \in [n-1]$ such that $u_{i_k} = u_{i_{k+1}}$. If we switch the two adjacent entries i_k and i_{k+1} of the permutation $(i_1, i_2, \ldots, i_n)$, then the resulting word v is unchanged. Indeed, if we set $h = u_{i_k} = u_{i_{k+1}}$, then:

- If $k < |q|$, then this switch interchanges two consecutive steps in Phase II, causing the corresponding letters v_j to get set in a possibly different order; but this does not change v because these two letters are set to the same value (namely, to $h + 1$).
- If $k > |q|$, then a similar argument works (using Phase I instead).
- If $k = |q|$, then recall from Remark 2.1 that the two phases can be arbitrarily interleaved. In particular, we can first perform all but the last step of Phase I, then perform all but the last step of Phase II, and finally perform the remaining two steps. The switch only affects these final two steps. However, the effect of these two steps is simply that the unique remaining unset v_j with $j \in q$ gets set to h, and the unique remaining unset v_j with $j \notin q$ gets set to $h + 1$. The switch clearly does not change this behavior, since it does not depend on i_k and i_{k+1}.

□

Lemma 2.2 states that the order between sites i with equal u_i does not matter.

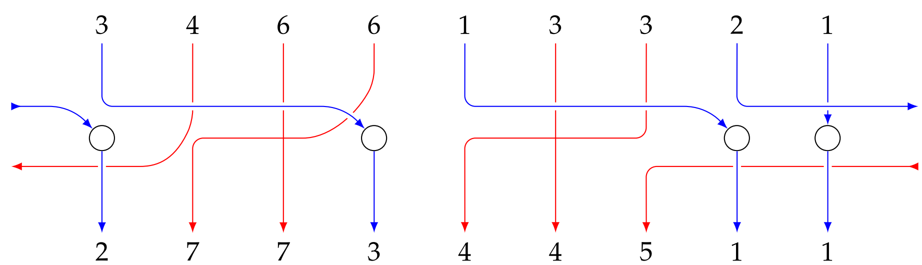
Remark 2.3. Let u and q be as before. Let $r = |q|$. There exists a $t \in [\ell]$ such that $p_{t-1} \leq r \leq p_t$. The word $v = q(u)$ then has type

$$(m_1, \ldots, m_{t-1}, r - p_{t-1}, p_t - r, m_{t+1}, m_{t+2}, \ldots). \tag{2.2}$$

Note that $p_t - r = m_t + (p_{t-1} - r)$. We think of this as splitting the class t into two new classes t and $t + 1$.

For all i processed in Phase I (resp. Phase II) of the algorithm, we have $u_i \geq t$ (resp. $u_i \leq t$). The queue q can be reconstructed from v and t as the set of all $j \in [n]$ satisfying $v_j \leq t$.

Example 2.4. We consider the 4-queue $q = \{1, 4, 8, 9\}$, and let $u = 346613321$. Thus, the type of u is $\mathbf{m} = (2, 1, 3, 1, 0, 2, 0, \ldots)$ with $p_2 = 3$ and $p_3 = 6$. Thus, the t in Remark 2.3 equals 3. To compute $q(u)$, draw the following diagram (whose upper row shows u, whose lower row shows $q(u)$, and whose middle row represents the set q by balls in the positions of its elements):



where the paths in red correspond to Phase I and those in blue are from Phase II (and where we have picked the permutation $(i_1, i_2, \ldots, i_n) = (5, 9, 8, 1, 7, 6, 2, 4, 3)$ out of a total of $2! \cdot 3! \cdot 2! = 24$ permutations satisfying $u_{i_1} \leq u_{i_2} \leq \cdots \leq u_{i_n}$; but any other among them would lead to the same result). Hence, we have $q(346613321) = 277344511$, which has type $(2, 1, 1, 2, 1, 0, 2, \ldots)$.

We illustrate the situation $v = q(u)$ with a $2 \times n$ array, where the first row is the word u and the second row has a circle labeled v_j for $j \in q$ or a square labeled v_j for $j \notin q$ in position j. Using this convention, we can write Example 2.4 as

$$\begin{array}{ccccccccc} 3 & 4 & 6 & 6 & 1 & 3 & 3 & 2 & 1 \\ 2 & \boxed{7} & \boxed{7} & 3 & \boxed{4} & \boxed{4} & \boxed{5} & 1 & 1 \end{array} \tag{2.3}$$

We call this the *graveyard diagram* of q and u.

There is a natural duality in the algorithm above. For each queue q, let q^* be the *contragredient dual queue*, defined by $(i \in q^*) \iff (n + 1 - i \notin q)$. Similarly, for each word u with ℓ classes, let u^* be the *contragredient dual word* defined by $u_i^* = \ell + 1 - u_{n+1-i}$. For a fixed $k \in [\ell]$, we call $\ell + 1 - k$ the *contragredient dual letter* of k. Note that if q is an r-queue, then q^* is the $(n - r)$-queue obtained by reflecting $[n] \setminus q$ through the middle of $[n]$. Similarly, u^* is obtained by reversing the word u and taking the contragredient dual letters.

Lemma 2.5 (Contragredient duality)**.** *Let q be a queue and u be a word with ℓ classes. Then we have $\left(q(u)\right)^* = q^*(u^*)$, i.e., we have $q(u)_i = \ell + 2 - q^*(u^*)_{n+1-i}$ for all i.*

Here, we treat $q(u)$ as a word with $\ell + 1$ classes, even if it may have only ℓ classes (in the degenerate case when $q = [n]$).

Proof. Phase I (resp. II) in the construction of $q(u)$ corresponds to Phase II (resp. I) in the construction of $q^*(u^*)$ when the word is reversed and the letters are replaced by their contragredient duals. Hence the claim follows. $\square$

We note that our construction is a minor variation of the construction given by [AAMP11, §3.1], which is a reformulation of that of [FM07]. We more precisely describe the relationship with [FM07] in Appendix A, where we also discuss a small variant that has appeared in [AS18].

2.2. **Multiline queues.** We now give our main definition of a multiline queue and spectral weight.

Definition 2.6. For $\sigma \in \mathfrak{S}_{\ell-1}$, a *$\sigma$-twisted multiline queue (MLQ) of type* $\mathbf{m} = (m_1, m_2, \ldots, m_\ell, 0, 0, \ldots)$, with ℓ classes, is a sequence of queues $\mathbf{q} = (q_1, \ldots, q_{\ell-1})$ such that q_i is a $p_{\sigma(i)}(\mathbf{m})$-queue and $m_\ell = n - p_{\ell-1}(\mathbf{m})$ (that is, $p_\ell(\mathbf{m}) = n$). When σ is the identity permutation, we simply call $\mathbf{q}$ an *(ordinary) MLQ of type* $\mathbf{m}$. We also consider $\mathbf{q}$ as a function on words by

$$\mathbf{q}(u) := q_{\ell-1}(\cdots q_2(q_1(u)) \cdots).$$

Remark 2.7. Our notion of an (ordinary) MLQ is equivalent to what is called a "discrete MLQ" in [AL18, §2.2], where we can recover the labeling of level k from the word $q_k(\cdots q_1(1^n) \cdots)$. See Appendix A for more details. We omit the word "discrete" as these are the only MLQs in this note.

We shall now introduce generating functions for queues.

Definition 2.8. Let $\mathbf{x} := \{x_1, x_2, \ldots, x_n\}$ be commuting indeterminates indexed by elements of $\mathbb{Z}/n\mathbb{Z}$. (Thus, $x_{n+k} = x_k$ for all $k \in \mathbb{Z}$.) The *weight* of a queue q is

$$\mathrm{wt}(q) := \prod_{i \in q} x_i.$$

The *weight* of a σ-twisted MLQ $\mathbf{q} = (q_1, \ldots, q_{\ell-1})$ is

$$\mathrm{wt}(\mathbf{q}) := \prod_{i=1}^{\ell-1} \mathrm{wt}(q_i).$$

Definition 2.9. For $\sigma \in \mathfrak{S}_{\ell-1}$ and a packed word u of type $\mathbf{m}$ with ℓ classes, we define the *σ-spectral weight* or *σ-amplitude* as

$$\langle u \rangle_\sigma := \sum_{\mathbf{q}} \mathrm{wt}(\mathbf{q}),$$

where the sum is over all σ-twisted MLQs $\mathbf{q}$ of type $\mathbf{m}$ satisfying $u = \mathbf{q}(1^n)$. (Recall that 1^n denotes the word in $\mathcal{W}_n$ whose all letters are 1.) When $\sigma = \mathrm{id}$ is the identity permutation, we simply call this the *spectral weight* or *amplitude* and denote it by $\langle u \rangle := \langle u \rangle_{\mathrm{id}}$.

Lemma 2.10. *Let $\ell \geq 1$ and $\sigma \in \mathfrak{S}_{\ell-1}$. Let $\mathbf{q}$ be a σ-twisted MLQ. Then the type of the word $\mathbf{q}(1^n)$ is the type of $\mathbf{q}$.*

Proof. Write the σ-twisted MLQ $\mathbf{q}$ as $(q_1, \ldots, q_{\ell-1})$. Let $\mathbf{m} = (m_1, m_2, \ldots, m_\ell, 0, 0, \ldots)$ be the type of $\mathbf{q}$. Thus, $m_1 + m_2 + \cdots + m_\ell = p_\ell(\mathbf{m}) = n$. Now, the word 1^n has type $(n, 0, 0, \ldots) = (m_1 + m_2 + \cdots + m_\ell, 0, 0, \ldots)$ (since $n = m_1 + m_2 + \cdots + m_\ell$). Every time we apply one of the queues q_i to this word, the type changes in a simple way (because of (2.2)): Namely, the plus sign between $m_{\sigma(i)}$ and $m_{\sigma(i)+1}$ turns into a comma (so, for example, the application of q_1 transforms it into $\left(m_1 + m_2 + \cdots + m_{\sigma(1)}, m_{\sigma(1)+1} + m_{\sigma(1)+2} + \cdots + m_\ell, 0, 0, \ldots\right)$). Hence, the action of $\mathbf{q}$ transforms 1^n into a word whose type has all the plus signs replaced by commas – *i.e.*, whose type is $(m_1, m_2, \ldots, m_\ell, 0, 0, \ldots) = \mathbf{m}$. In other words, the type of the word $\mathbf{q}(1^n)$ is $\mathbf{m}$. This proves Lemma 2.10. $\square$

Alternative proof of Lemma 2.10. Here is an alternative way of presenting essentially the same argument, using a different point of view: Since $(q_1, \ldots, q_{\ell-1}) = \mathbf{q}$ is a σ-twisted MLQ of type $\mathbf{m}$, we know that the sizes $|q_1|, |q_2|, \ldots, |q_{\ell-1}|$ of its queues are a permutation of $p_1(\mathbf{m}), p_2(\mathbf{m}), \ldots, p_{\ell-1}(\mathbf{m})$. Thus, $p_1(\mathbf{m}), p_2(\mathbf{m}), \ldots, p_{\ell-1}(\mathbf{m})$ is the weakly increasing permutation of the sequence $|q_1|, |q_2|, \ldots, |q_{\ell-1}|$. Now, define the *p-sequence* of a type $\mathbf{n}$ to be the weakly increasing infinite sequence $(p_0(\mathbf{n}), p_1(\mathbf{n}), p_2(\mathbf{n}), \ldots)$ of integers (which uniquely determines $\mathbf{n}$). Furthermore, if u is a word of type $\mathbf{n}$, then we define the p-sequence of u to be the p-sequence of $\mathbf{n}$. Then, (2.2) shows the following: If q is a queue and u is a word, then the p-sequence of $q(u)$ is obtained from the p-sequence of u by inserting $|q|$ into the p-sequence at the appropriate position (appropriate in the sense that the resulting sequence is still weakly increasing). Hence, the p-sequence of the word $\mathbf{q}(1^n)$ is obtained from the p-sequence $(0, n, n, n, \ldots)$ of the word 1^n by inserting $|q_1|, |q_2|, \ldots, |q_{\ell-1}|$ at the appropriate positions. In other words, the p-sequence of the word $\mathbf{q}(1^n)$ is $(0, p_1(\mathbf{m}), p_2(\mathbf{m}), \ldots, p_{\ell-1}(\mathbf{m}), n, n, \ldots, n)$ (since $p_1(\mathbf{m}), p_2(\mathbf{m}), \ldots, p_{\ell-1}(\mathbf{m})$ is the weakly increasing permutation of the sequence $|q_1|, |q_2|, \ldots, |q_{\ell-1}|$). In other words, this p-sequence is $(p_0(\mathbf{m}), p_1(\mathbf{m}), p_2(\mathbf{m}), \ldots)$. Hence, the type of the word $\mathbf{q}(1^n)$ is $\mathbf{m}$. $\square$

Example 2.11. Let $\ell \geq 1$. Let u be a packed word with ℓ classes and type $\mathbf{m}$. For each $k \in [\ell - 1]$, let q_k be the set of all sites i such that $u_i \leq k$. It is then easy to check that $\mathbf{q} := (q_1, q_2, \ldots, q_{\ell-1})$ is an MLQ of type $\mathbf{m}$ satisfying $\mathbf{q}(1^n) = u$ and has weight $\operatorname{wt}(\mathbf{q}) = \prod_{j \in \mathbb{Z}/n\mathbb{Z}} x_j^{\ell - u_j}$. Hence, $\langle u \rangle \neq 0$ (as a polynomial over $\mathbb{Z}$).

2.3. **Symmetric polynomials.** We also need the *elementary symmetric polynomials* and *complete homogeneous symmetric polynomials*. Recall that they are defined for each $N \in \mathbb{N}$ and $k \geq 0$ and any N indeterminates $y_1, y_2, \ldots, y_N$ by

$$e_k(y_1, y_2, \ldots, y_N) = \sum_{1 \leq i_1 < \cdots < i_k \leq N} y_{i_1} \cdots y_{i_k},$$
$$h_k(y_1, y_2, \ldots, y_N) = \sum_{1 \leq i_1 \leq \cdots \leq i_k \leq N} y_{i_1} \cdots y_{i_k},$$

respectively. We define $e_k(y_1, \ldots, y_N) = 0$ and $h_k(y_1, \ldots, y_N) = 0$ for $k < 0$. For more details on symmetric polynomials, we refer the reader to [Sta99, Ch. 7].

3. Main results

In this section, we state our main results and use them to prove the commutativity conjecture of [AAMP11] and [AL18, Conj. 3.10].

3.1. σ-independence of the spectral weight.

Theorem 3.1. *Let u be a packed word of type $\mathbf{m}$ with ℓ classes. For any $\sigma \in \mathfrak{S}_{\ell-1}$, we have*

$$\langle u \rangle = \langle u \rangle_\sigma .$$

We will give the proof of Theorem 3.1 in Section 6.

3.2. **Queue action and merges.** We shall next study the interplay between the action of queues on words and the merging of adjacent classes.

If w is a word of type $\mathbf{m}$, and if $j \geq 0$, then $p_j(\mathbf{m})$ is the number of letters of w that are at most j.

For a word w and a nonnegative integer k, we let $\vee^{(k)}w$ be the word obtained from w by decrementing (by 1) all but the k smallest letters of w. This is only well-defined if these k smallest letters are determined uniquely and the remaining $n-k$ letters are >1. In other words, this is only well-defined if $k \in \{p_j(\mathbf{m}) \mid j \geq 1\}$, where $\mathbf{m}$ is the type of w. Note that $\vee^{(k)}w = \vee_j w$, where j is such that $k = p_j(\mathbf{m})$.

Lemma 3.2. *Let $u \in \mathcal{W}_n$ be a word of type $\mathbf{m}$. Let $k \in \{p_j(\mathbf{m}) \mid j \geq 1\}$. If q is a queue, then*
$$\vee^{(k)}q(u) = q(\vee^{(k)}u).$$
In particular, both $\vee^{(k)}q(u)$ and $\vee^{(k)}u$ are well-defined.

Proof. The word $\vee^{(k)}u$ is well-defined since $k = p_j(\mathbf{m})$ for some $j \geq 1$; furthermore, $\vee^{(k)}q(u)$ is well-defined since the type $\mathbf{n}$ of $q(u)$ satisfies
$$k \in \{p_j(\mathbf{m}) \mid j \geq 1\} \subseteq \{p_j(\mathbf{m}) \mid j \geq 1\} \cup \{|q|\} = \{p_j(\mathbf{n}) \mid j \geq 1\}.$$
Thus, it remains to prove $\vee^{(k)}q(u) = q(\vee^{(k)}u)$. The permutation $(i_1, i_2, \ldots, i_n)$ in the construction of $q(u)$ also works for the construction of $q(\vee^{(k)}u)$, since $(\vee^{(k)}u)_a \leq (\vee^{(k)}u)_b$ whenever $u_a \leq u_b$. Consequently, the construction of $q(\vee^{(k)}u)$ proceeds exactly like the construction of $q(u)$ (with the same entries being set in the same order), except that all but the k smallest letters are now smaller by 1. Hence, $q(\vee^{(k)}u)$ is obtained from $q(u)$ by decrementing (by 1) all but the k smallest letters of $q(u)$. However, the word $\vee^{(k)}q(u)$ is obtained from $q(u)$ in exactly the same way. Therefore, we have $\vee^{(k)}q(u) = q(\vee^{(k)}u)$ as claimed. ∎

Example 3.3. Consider $q = \{1,4,5,9,10\}$ and $v = 3455313321$:
$$\begin{array}{cccccccccc} 3 & 4 & 5 & 5 & 3 & 1 & 3 & 3 & 2 & 1 \\ 2 & \boxed{6} & \boxed{6} & 3 & 3 & \boxed{4} & \boxed{4} & \boxed{5} & 1 & 1 \end{array}.$$
Let $u = 3566413321$, and note that $v = \vee_3 u = \vee^{(6)}u$ and
$$\begin{array}{cccccccccc} 3 & 5 & 6 & 6 & 4 & 1 & 3 & 3 & 2 & 1 \\ 2 & \boxed{7} & \boxed{7} & 3 & 3 & \boxed{4} & \boxed{5} & \boxed{6} & 1 & 1 \end{array},$$
where we have $2663344511 = \vee_4 2773345611 = \vee^{(6)}2773345611$. Similarly, let $u' = 4566413421$, and note that $v = \vee_3 u'$ and
$$\begin{array}{cccccccccc} 4 & 5 & 6 & 6 & 4 & 1 & 3 & 4 & 2 & 1 \\ 2 & \boxed{7} & \boxed{7} & 3 & 4 & \boxed{5} & \boxed{5} & \boxed{6} & 1 & 1 \end{array},$$
where we have $2663344511 = \vee_4 2773455611$.

Lemma 3.4. *Let q' be a $p_i(\mathbf{m})$-queue for some type $\mathbf{m}$ and some $i \geq 1$. Let the type $\mathbf{m}$ have ℓ classes, and let $\sigma \in \mathfrak{S}_{\ell-1}$. Let $\mathbf{q}$ be a σ-twisted MLQ of type $\vee_i \mathbf{m}$. For the word $u = \mathbf{q}(q'(1^n))$, we have*
$$\mathbf{q}(1^n) = \vee_i u.$$

Proof. Write the σ-twisted MLQ $\mathbf{q}$ as $(q_1, \ldots, q_{\ell-1})$. It has type $\vee_i \mathbf{m}$.

Set $\mathbf{q}' := (q', q_1, \ldots, q_{\ell-1})$; this is easily seen to be a ζ-twisted MLQ of type $\mathbf{m}$, for some $\zeta \in \mathfrak{S}_\ell$ (since the multiset of the $p_k(\vee_i \mathbf{m})$ for $k \geq 1$ is precisely the multiset of the $p_k(\mathbf{m})$ for $k \geq 1$ with one copy of $p_i(\mathbf{m}) = |q'|$ removed). Furthermore, the definition of u becomes $u = \mathbf{q}'(1^n)$. Hence, the type of u is $\mathbf{m}$ (by Lemma 2.10).

Set $k = p_i(\mathbf{m})$. Then, q' is a k-queue, so that the type $\mathbf{n}$ of the word $q'(1^n)$ satisfies $p_1(\mathbf{n}) = k$. Hence, any word obtained by actions of queues on $q'(1^n)$ will have its type $\mathbf{n}$ satisfy $k \in \{p_j(\mathbf{n}) \mid j \geq 1\}$.

Since the type of u is $\mathbf{m}$, and since $k = p_i(\mathbf{m})$, we have

$$\vee_i u = \vee^{(k)}(u) = \vee^{(k)} \mathbf{q}\big(q'(1^n)\big) = \mathbf{q}\big(\vee^{(k)} q'(1^n)\big) \tag{3.1}$$

by repeated use of Lemma 3.2 (since any word obtained by actions of queues on $q'(1^n)$ will have its type $\mathbf{n}$ satisfy $k \in \{p_j(\mathbf{n}) \mid j \geq 1\}$). It is clear that $\vee^{(k)} q'(1^n) = 1^n$, and so (3.1) becomes $\mathbf{q}(1^n) = \vee_i u$. $\square$

3.3. **Spectral weights of merged words.** The preceding lemmas will help us establish a rule for products of spectral weights with elementary symmetric polynomials (somewhat similar to the dual Pieri rule, *e.g.*, [Sta99, Section 7.15]):

Theorem 3.5. *Let $\mathbf{m}$ be a type with $m_i \neq 0$ and $m_{i+1} \neq 0$. Let v be a packed word of type $\vee_i \mathbf{m}$. Then,*

$$\langle v \rangle \, e_{p_i(\mathbf{m})}(x_1, \ldots, x_n) = \sum_u \langle u \rangle \,,$$

where we sum over all u of type $\mathbf{m}$ such that $v = \vee_i u$.

Proof. The type $\mathbf{m}$ is packed (since $m_i \neq 0$ and $m_{i+1} \neq 0$, and since $\vee_i \mathbf{m}$ is packed). Let $\vee_i \mathbf{m}$ have ℓ classes; then, $\mathbf{m}$ has $\ell + 1$ classes. First note that

$$\langle v \rangle \, e_{p_i(\mathbf{m})}(x_1, \ldots, x_n) = \sum_{(\mathbf{q}, q')} \operatorname{wt}(\mathbf{q}) \operatorname{wt}(q'),$$

where we sum over all pairs $(\mathbf{q}, q')$ such that

- $\mathbf{q} = (q_1, \ldots, q_{\ell-1})$ is an MLQ of type $\vee_i \mathbf{m}$ such that $v = \mathbf{q}(1^n)$ and
- q' is a $p_i(\mathbf{m})$-queue.

Given any such pair $(\mathbf{q}, q')$, we set

$$\theta(\mathbf{q}, q') := (q', q_1, \ldots, q_{\ell-1});$$

the result is a $(s_{i-1} \cdots s_2 s_1)$-twisted MLQ of type $\mathbf{m}$ with weight $\operatorname{wt}\big(\theta(\mathbf{q}, q')\big) = \operatorname{wt}(\mathbf{q}) \operatorname{wt}(q')$. But recall that $v = \mathbf{q}(1^n)$; thus, by Lemma 3.4, we have

$$v = \mathbf{q}(1^n) = \vee_i \mathbf{q}\big(q'(1^n)\big) = \vee_i\big(\theta(\mathbf{q}, q')(1^n)\big).$$

Thus, we have defined a weight preserving bijection θ from the set of all pairs $(\mathbf{q}, q')$ as above to the set of all $(s_{i-1} \cdots s_1)$-twisted MLQs $\widetilde{\mathbf{q}}$ of type $\mathbf{m}$ satisfying $v = \vee_i \widetilde{\mathbf{q}}(1^n)$. Hence, we have

$$\sum_{(\mathbf{q}, q')} \operatorname{wt}(\mathbf{q}) \operatorname{wt}(q') = \sum_{\widetilde{\mathbf{q}}} \operatorname{wt}(\widetilde{\mathbf{q}}) = \sum_u \langle u \rangle_{s_{i-1} \cdots s_1} \,,$$

where the last sum is over all words u of type $\mathbf{m}$ satisfying $v = \vee_i u$. Finally, we have $\langle u \rangle_{s_{i-1} \cdots s_1} = \langle u \rangle$ for all such u by Theorem 3.1. Combining all the above equalities, we find $\langle v \rangle \, e_{p_i(\mathbf{m})}(x_1, \ldots, x_n) = \sum_u \langle u \rangle$. $\square$

Remark 3.6. Our proof of Theorem 3.1 is by constructing a bijection ω, and hence, we can give a bijective proof of Theorem 3.5 by the composition $\omega \circ \theta$.

Example 3.7. Suppose $n = 5$. Let $v = 13234$, and we have that $v = \vee_3 u$ if and only if $u \in \{13245, 14235\}$. By examining all possible MLQs for these words, we obtain

$$\begin{aligned}
\langle 13234 \rangle &= x_1 x_2 x_3^2 x_4 (x_1^2 + x_1 x_4 + x_1 x_5 + x_4 x_5 + x_5^2), \\
\langle 13245 \rangle &= x_1 x_2 x_3^2 x_4 (x_1^2 + x_1 x_4 + x_1 x_5 + x_4^2 + x_4 x_5 + x_5^2) \\
&\qquad \times (x_1 x_2 x_3 + x_1 x_2 x_5 + x_1 x_3 x_5 + x_2 x_3 x_5), \\
\langle 14235 \rangle &= x_1 x_2 x_3^2 x_4^2 (x_1^3 x_2 + x_1^3 x_3 + x_1^3 x_5 + x_1^2 x_2 x_3 + x_1^2 x_2 x_4 + 2 x_1^2 x_2 x_5 \\
&\qquad\qquad + x_1^2 x_3 x_4 + 2 x_1^2 x_3 x_5 + x_1^2 x_4 x_5 + x_1^2 x_5^2 + x_1 x_2 x_3 x_5 \\
&\qquad\qquad + x_1 x_2 x_4 x_5 + 2 x_1 x_2 x_5^2 + x_1 x_3 x_4 x_5 + 2 x_1 x_3 x_5^2 + x_1 x_4 x_5^2 \\
&\qquad\qquad + x_1 x_5^3 + x_2 x_3 x_5^2 + x_2 x_4 x_5^2 + x_2 x_5^3 + x_3 x_4 x_5^2 + x_3 x_5^3).
\end{aligned}$$

(We have factored the expressions for readability only.) We verify Theorem 3.5 in this case by computing $\langle 13234 \rangle \, e_3(x_1, x_2, x_3, x_4, x_5) = \langle 13245 \rangle + \langle 14235 \rangle$.

By repeated applications of Theorem 3.5, we obtain the following:

Corollary 3.8. *Let T be a finite set of positive integers. Let $\mathbf{m}$ be a type such that every $t \in T$ satisfies $m_t \neq 0$ and $m_{t+1} \neq 0$. Let v be a packed word of type $\vee_T \mathbf{m}$. Then,*

$$\langle v \rangle \prod_{t \in T} e_{p_t(\mathbf{m})}(x_1, \ldots, x_n) = \sum_u \langle u \rangle \,,$$

where we sum over all u of type $\mathbf{m}$ such that $v = \bigvee_T u$.

3.4. **A Jacobi-Trudi-like formula for special u.** Throughout this subsection, we shall regard the sites $1, 2, \ldots, n$ as elements of $\{1, 2, \ldots, n\}$ rather than of $\mathbb{Z}/n\mathbb{Z}$. In particular, they are totally ordered by $1 < 2 < \cdots < n$.

Let ℓ and r be positive integers. Let $B = \{b_1 < b_2 < \cdots < b_r\} \subseteq [n]$. Let $v = (v_1, v_2, \ldots, v_r)$ be an r-tuple of elements of $[\ell - 1]$. Define a word $u(v) \in \mathcal{W}_n$ with ℓ classes by

$$\big(u(v)\big)_i = \begin{cases} v_j & \text{if } i = b_j \text{ for some } j, \\ \ell & \text{otherwise.} \end{cases} \tag{3.2}$$

We shall now state a determinantal formula for the special case of $\langle u(v) \rangle$ when v is a weakly decreasing r-tuple whose entries cover $[\ell - 1]$:

Theorem 3.9. *Let ℓ be a positive integer. Let $B = \{b_1 < b_2 < \cdots < b_r\} \subseteq [n]$. Let $(v_1 \geq v_2 \geq \cdots \geq v_r)$ be a weakly decreasing r-tuple of integers such that $\{v_1, v_2, \ldots, v_r\} = [\ell - 1]$. For each $j \in [r]$, set $\gamma_j = \ell - v_j$. (The value γ_j can also be described as the number of distinct letters among $v_1, v_2, \ldots, v_j$.) Then*

$$\langle u(v) \rangle = \left(\prod_{b \in B} x_b \right) \det \left(h_{\gamma_j + i - j - 1}(x_1, \ldots, x_{b_j}) \right)_{i,j \in [r]} .$$

The proof of this theorem is given in Section 5.

3.5. **Conclusions for Aas–Linusson MLQs.** We can use Theorem 3.9 to settle two conjectures from [AL18].

Fix a set of sites $B = \{b_1 < b_2 < \cdots < b_r\}$. We say that a set S of integers is *lacunar* if $i \in S$ implies $i+1 \notin S$. Let $S \subseteq [r-1]$ be lacunar, and define the permutation $\sigma_S := \left(\prod_{i\in S} s_i\right) w_0$, where $s_i, w_0 \in \mathfrak{S}_r$. Note that the elements $\{s_i \mid i \in S\}$ all commute, so their product, and hence σ_S, is well-defined. In one-line notation, σ_S is the list of all elements of $[r]$ in decreasing order, except that for each $i \in S$, the pair $(i, i+1)$ is sorted back into increasing order.

For each permutation $\tau \in \mathfrak{S}_r$, we let v_τ be the r-tuple $(\tau_1, \tau_2, \ldots, \tau_r)$ (that is, the one-line notation of τ). To ease notation, we shall identify any $\tau \in \mathfrak{S}_r$ with the corresponding word $u(v_\tau) \in \mathcal{W}_n$ defined by (3.2), where we set $\ell = r+1$.

In [AL18, Conj. 3.10], a formula for the spectral weight $\langle \sigma_S \rangle$ at $x_1 = \cdots = x_n = 1$ is conjectured. We now prove a version of this conjecture for general x_i.

Corollary 3.10. *Let $B = \{b_1 < b_2 < \cdots < b_r\} \subseteq [n]$. Let $T \subseteq [r-1]$ be lacunar, and let $\psi(T) = \sum_{S\subseteq T} \langle \sigma_S \rangle$. Then we have*

$$\psi(T) = \left(\prod_{t\in T} e_t(x_1, \ldots, x_n)\right) \left\langle \bigvee_T w_0 \right\rangle$$
$$= \left(\prod_{b\in B} x_b\right) \left(\prod_{t\in T} e_t(x_1, \ldots, x_n)\right) \det\big(h_{\gamma_j+i-j-1}(x_1, \ldots, x_{b_j})\big)_{i,j\in[r]},$$

where $\gamma_j = j - |\{t \in T \mid t > r-j\}|$.

Proof. Recall that we identify the permutation $w_0 \in \mathfrak{S}_r$ with the word $u(v_{w_0}) \in \mathcal{W}_n$; the latter word has type $\mathbf{m} := (1, 1, \ldots, 1, n-r, 0, 0, \ldots)$ (starting with r ones). Hence, the word $\bigvee_T w_0$ has type $\bigvee_T \mathbf{m}$. By Corollary 3.8 (applied to $v = \bigvee_T w_0$), we thus have

$$\left(\prod_{t\in T} e_t(x_1, \ldots, x_n)\right) \left\langle \bigvee_T w_0 \right\rangle = \sum \langle u' \rangle,$$

where the sum ranges over all words $u' \in \mathcal{W}_n$ of type $\mathbf{m}$ satisfying $\bigvee_T u' = \bigvee_T w_0$. But the words $u' \in \mathcal{W}_n$ of type $\mathbf{m}$ satisfying $\bigvee_T u' = \bigvee_T w_0$ are precisely the words of the form σ_S (recall that this stands for $u(v_{\sigma_S})$) for $S \subseteq T$. Hence, the $\sum \langle u' \rangle$ in the above equality rewrites as $\psi(T)$. This proves the first equality of the corollary.

To prove the second equality, we consider $\ell = r - |T| + 1$ and the r-tuple

$$v = \big(v_j = \ell - j + |\{t \in T \mid t > r-j\}| = \ell - \gamma_j\big)_{j=1}^r.$$

Note that $v_1 = r - |T| = \ell - 1$ and $v_r = 1$ and

$$v_{i+1} = v_i - \begin{cases} 0 & \text{if } r-i \in T, \\ 1 & \text{if } r-i \notin T, \end{cases} \qquad \text{for each } i \in [r-1].$$

Thus, v is weakly decreasing and $\{v_1, v_2, \ldots, v_r\} = [\ell - 1]$ and $u(v) = \bigvee_T w_0$. Hence, the second equality follows from Theorem 3.9. □

Example 3.11. Consider $n = 8$, $B = \{1, 2, 5, 6, 8\}$, and $T = \{1, 4\}$. Then we have $\bigvee_T w_0 = 33442141$, and

$$\psi(T) = \langle 54663261 \rangle + \langle 45663261 \rangle + \langle 54663162 \rangle + \langle 45663162 \rangle.$$

Applying Theorem 3.5 to $t = 4$ twice and then to $t = 1$, we obtain

$$\begin{aligned}\psi(T) &= e_4(x_1,\dots,x_8)\left(\langle 44553251\rangle + \langle 44553152\rangle\right)\\ &= e_1(x_1,\dots,x_8)\, e_4(x_1,\dots,x_8)\,\langle 33442141\rangle\,.\end{aligned}$$

Using Theorem 3.9, this further becomes

$$\begin{aligned}\psi(T) &= x_1x_2x_5x_6x_8\, e_1(x_1,\dots,x_8)\, e_4(x_1,\dots,x_8)\\ &\quad\times \det\begin{pmatrix} 1 & 0 & 0 & 0 & 0\\ h_1[1] & 1 & 1 & 1 & 0\\ h_2[1] & h_1[2] & h_1[5] & h_1[6] & 1\\ h_3[1] & h_2[2] & h_2[5] & h_2[6] & h_1[8]\\ h_4[1] & h_3[2] & h_3[5] & h_3[6] & h_2[8]\end{pmatrix},\end{aligned}$$

where $h_i[k] := h_i(x_1,\dots,x_k)$.

Note that the well-known formula for Möbius inversion on the boolean lattice yields

$$\langle \sigma_S\rangle = \sum_{T\subseteq S}(-1)^{|S|-|T|}\psi(T)$$

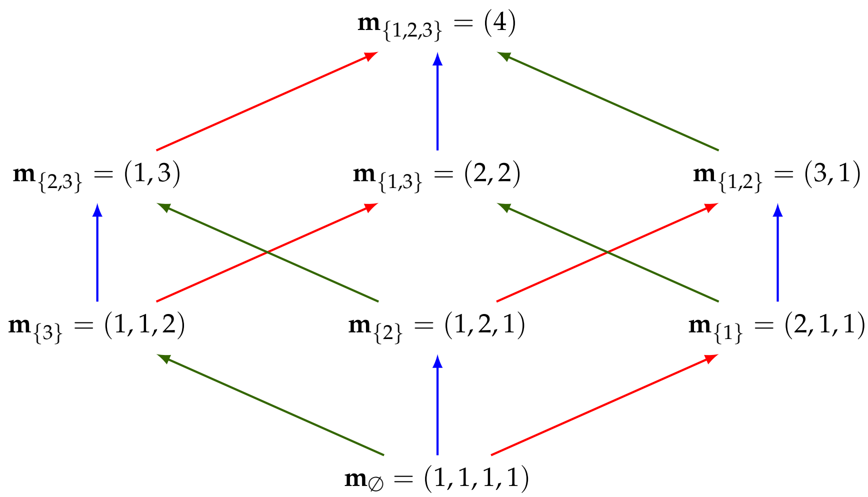
for any $S \subseteq [r-1]$. Therefore, by specializing Corollary 3.10 at $x_1 = \cdots = x_n = 1$, we obtain [AL18, Conj. 3.10] (which is a generalization of [AL18, Conj. 3.9]). Additionally, this proves [AL18, Conj. 3.6] (which is a generalization of [AL18, Conj. 3.4]).

4. The TASEP connection

We now explain how our proof of Theorem 3.1 gives a proof of the commutativity conjecture of [AAMP11].

There are 2^{n-1} packed types for words of length n, as they are the compositions of n (see [Sta12, Section 1.2]). We let the subset $S \subseteq [n-1]$ correspond to the type of the word obtained by merging i and $i+1$ in $12\cdots n$ for each $i \in S$. Denote this type by $\mathbf{m}_S$. Note that $\{p_1(\mathbf{m}_S),\dots,p_{\ell-1}(\mathbf{m}_S)\} = [n-1]\setminus S$, where $\mathbf{m}_S$ has ℓ classes; this is the complement of the usual bijection between subsets of $[n-1]$ and compositions of n. Let $\mathcal{W}_S$ denote the set of words of type $\mathbf{m}_S$. Let V_S be the vector space over $\mathbb{R}$ with basis $\{\epsilon_w \mid w \in \mathcal{W}_S\}$.

Example 4.1. For $n = 4$, we have



where we have drawn an arrow $\mathbf{m}_S \to \mathbf{m}_{S\cup\{i\}}$ for each $S \subseteq [n-1]$ and each $i \in [n-1] \setminus S$ (this is the Hasse diagram of the Boolean lattice of subsets of $[n-1]$).

The *totally asymmetric simple exclusion process* (TASEP) is a Markov chain on $\mathcal{W}_S$, where $S \subseteq [n-1]$, as follows. For a state $u \in \mathcal{W}_S$, we move to a new state by picking a random $i \in [n]$ and either

- if $u_i > u_{i+1}$, swap the positions u_i and u_{i+1}, or
- do nothing (*i.e.* stay at u).

Let $M_S \colon V_S \to V_S$ be the transition matrix of this Markov chain. Note that these moves preserve the type of the words; thus we could consider this as a Markov chain on $\mathcal{W}_n$, where $\mathcal{W}_S$ becomes an irreducible component. For $i \notin S$, we have the merging map $\Phi_i \colon \mathcal{W}_S \to \mathcal{W}_{S\cup\{i\}}$ given by $\Phi_i(\epsilon_u) = \epsilon_{\vee^{(i)}u}$. It is easy to see that $\Phi_i M_S = M_{S\cup\{i\}}\Phi_i$.

Building on work by Ferrari and Martin [FM06, FM07], the paper [AAMP11] introduced opposite operators $\Psi_i \colon \mathcal{W}_S \to \mathcal{W}_{S\setminus\{i\}}$ given by $\Psi_i(\epsilon_u) = \sum_q \epsilon_{q(u)}$, where the sum is taken over all i-queues q, and showed that $\Psi_i M_S = M_{S\setminus\{i\}}\Psi_i$. Furthermore they proposed the *commutativity conjecture*: that $\Psi_i\Psi_j = \Psi_j\Psi_i$. By looking at the (u,v) entry of both sides of this equation, the commutativity conjecture is asking whether the number of (i,j)-configurations C such that $v = C(u)$ equals the number of (j,i)-configurations C' such that $v = C'(u)$. Thus, our proof of Theorem 3.1 shows that $\widetilde{\Psi}_i\widetilde{\Psi}_j = \widetilde{\Psi}_j\widetilde{\Psi}_i$ for the weighted operators $\widetilde{\Psi}_i$ given by

$$\widetilde{\Psi}_i(\epsilon_u) = \sum_q \operatorname{wt}(q)\epsilon_{q(u)},$$

where we also sum over all i-queues q. Note that $\widetilde{\Psi}_i = \Psi_i$ when we specialize $x_1 = \cdots = x_n = 1$, giving the connection between our MLQs and the multi-species TASEP. We note that the proof of interlacing given in [AAMP11] is significantly different from our approach.

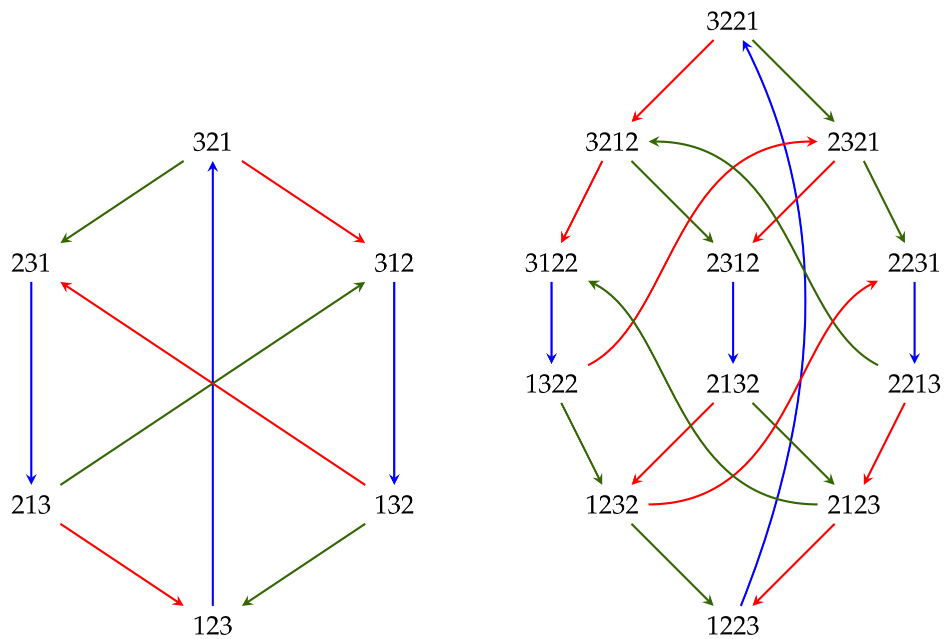


Figure 1. The states and transitions for $\mathcal{W}_{\varnothing}$ for $n = 3$ (left) and $\mathcal{W}_{\{2\}}$ for $n = 4$ (right). All probabilities of the drawn transitions are $1/n$.

We have not managed to find a process similar to the TASEP whose transition matrix $\widetilde{M}_S$ would satisfy $\widetilde{M}_S \widetilde{\Psi}_i = \widetilde{\Psi}_i \widetilde{M}_{S\setminus\{i\}}$ for our $\widetilde{\Psi}_i$ operators. Note however that queues give us *a* random process with this property: for a word $u \in \mathcal{W}_S$, a move in the chain is given by

(1) picking a random i-queue q
(2) going to the state $\vee_t q(u) \in \mathcal{W}_S$, where $t = \min\{k \mid p_k(\mathbf{m}_S) \geq i\}$.

5. Proof of Theorem 3.9

The goal of this section is to prove Theorem 3.9. Along the way, we shall derive a number of intermediate results, some of which may be of independent interest.

As in Subsection 3.4, we shall consider the sites as elements of the totally ordered set $\{1, 2, \ldots, n\}$ (ordered by $1 < 2 < \cdots < n$) throughout this section. Moreover, we shall use infinitely many distinct commuting indeterminates $\ldots, x_{-2}, x_{-1}, x_0, x_1, x_2, \ldots$ instead of those indexed by $\mathbb{Z}/n\mathbb{Z}$; thus we do not have $x_{n+k} = x_k$ in this section.

5.1. **Lattice paths and the Lindström–Gessel–Viennot theorem.** Our arguments will rely on the *Lindström–Gessel–Viennot (LGV) Lemma* [GV85, Lin73] and on a re-interpretation of MLQs as a certain kind of semistandard tableaux (of non-partition shape). This takes inspiration from the "bully paths" of [AL18] as well as from the standard proof of the Jacobi–Trudi identities for Schur functions [Sta99, First proof of Theorem 7.16.1]. We begin with basic definitions.

The *lattice* shall mean the (infinite) directed graph whose vertices are all pairs of integers (that is, its vertex set is $\mathbb{Z}^2$), and whose arcs are

$$\begin{aligned} (i,j) &\to (i,j+1) && \text{for all } (i,j) \in \mathbb{Z}^2, \qquad \text{and} \\ (i,j) &\to (i+1,j) && \text{for all } (i,j) \in \mathbb{Z}^2. \end{aligned}$$

The arcs of the first kind are called *north-steps*, whereas the arcs of the second kind are called *east-steps*. The vertices of the lattice will just be called *vertices*. We consider the lattice as the usual integer lattice in the Cartesian plane.

For each vertex $v = (i,j) \in \mathbb{Z}^2$, we set $\mathrm{x}(v) = i$ and $\mathrm{y}(v) = j$. We refer to $\mathrm{x}(v)$ (resp. $\mathrm{y}(v)$) as the *x-coordinate* (resp. *y-coordinate*) of v. The *y-coordinate* of an east-step $(i,j) \to (i+1,j)$ is defined to be j.

For each arc a of the lattice, we define the *weight* of a as the monomial

$$\mathrm{wt}(a) := \begin{cases} x_j & \text{if } a \text{ is an east-step } (i,j) \to (i+1,j), \\ 1 & \text{if } a \text{ is a north-step } (i,j) \to (i,j+1). \end{cases}$$

Thus, all north-steps have weight 1, while east-steps with y-coordinate j have weight x_j.

Fix $k \in \mathbb{N}$. A k-tuple of vertices of the lattice will be called a *k-vertex*. If $\mathbf{v} = (A_1, A_2, \ldots, A_k)$ is a k-vertex, and if $\sigma \in \mathfrak{S}_k$ is a permutation, then $\sigma(\mathbf{v})$ denotes the k-vertex $\left(A_{\sigma(1)}, A_{\sigma(2)}, \ldots, A_{\sigma(k)}\right)$. A *path* simply means a (directed) path in the lattice. The *weight* of a path p, denoted $\mathrm{wt}(p)$, is defined as the product of the weights of all arcs of this path; this weight is a monomial. If A and B are two vertices, then $N(A,B)$ shall denote the set of all paths from A to B.

It is easy to see (see, *e.g.*, [Sta12, (2.36)]) that any two vertices $A = (a,b)$ and $B = (c,d)$ satisfy

$$\sum_{p \in N(A,B)} \mathrm{wt}(p) = h_{c-a}(x_b, x_{b+1}, \ldots, x_d). \tag{5.1}$$

If $(A_1, A_2, \ldots, A_k)$ and $(B_1, B_2, \ldots, B_k)$ are two k-vertices, then a *non-intersecting lattice path tuple (NILP)* from $(A_1, A_2, \ldots, A_k)$ to $(B_1, B_2, \ldots, B_k)$ shall mean a k-tuple $(p_1, p_2, \ldots, p_k)$ of paths such that

- each p_i is a path from A_i to B_i;
- no two of the paths $p_1, p_2, \ldots, p_k$ have a vertex in common.

(Visually speaking, the paths must neither cross nor touch.)

The *weight* of a NILP $\mathbf{p} = (p_1, p_2, \ldots, p_k)$ is the monomial $\mathrm{wt}(\mathbf{p})$ defined by

$$\mathrm{wt}(\mathbf{p}) := \prod_{i=1}^{k} \mathrm{wt}(p_i).$$

See Figure 2 for an illustration.

If $\mathbf{u}$ and $\mathbf{v}$ are two k-vertices, then $N(\mathbf{u}, \mathbf{v})$ denotes the set of all NILPs from $\mathbf{u}$ to $\mathbf{v}$.

We now shall state a folklore result, which follows from the Lindström–Gessel–Viennot lemma:[1]

[1] See the proof of Corollary 4 in https://math.stackexchange.com/questions/2870640 (applied to $\omega_a = \mathrm{wt}(a)$) for a detailed derivation of Proposition 5.1.

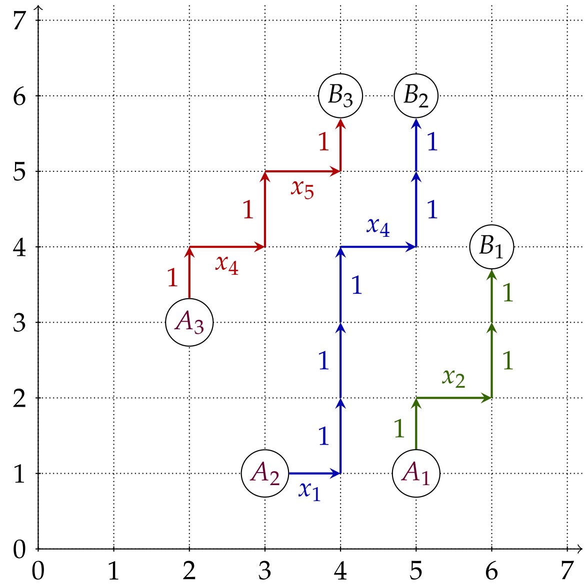


FIGURE 2. A NILP from the 3-vertex (A_1, A_2, A_3) to the 3-vertex (B_1, B_2, B_3) of weight $x_1x_2x_4^2x_5$. The weights of the edges of the paths are written next to the edges.

Proposition 5.1. *Let $k \in \mathbb{N}$. Let $\mathbf{u} = (A_1, A_2, \ldots, A_k)$ and $\mathbf{v} = (B_1, B_2, \ldots, B_k)$ be two k-vertices such that*

$$\begin{aligned}
\mathrm{x}(A_1) &\geq \mathrm{x}(A_2) \geq \cdots \geq \mathrm{x}(A_k),\\
\mathrm{y}(A_1) &\leq \mathrm{y}(A_2) \leq \cdots \leq \mathrm{y}(A_k),\\
\mathrm{x}(B_1) &\geq \mathrm{x}(B_2) \geq \cdots \geq \mathrm{x}(B_k),\\
\mathrm{y}(B_1) &\leq \mathrm{y}(B_2) \leq \cdots \leq \mathrm{y}(B_k).
\end{aligned}$$

Then,

$$\sum_{\mathbf{p} \in N(\mathbf{u},\mathbf{v})} \mathrm{wt}(\mathbf{p}) = \det \left(\sum_{p \in N(A_i, B_j)} \mathrm{wt}(p) \right)_{i,j \in [k]}.$$

The situation of this proposition is illustrated in Figure 2.

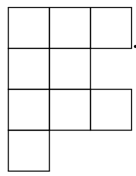
5.2. **Pseudo-partitions and tableaux.** We shall next introduce the concepts of pseudo-partitions and their corresponding semistandard tableaux; we will then express a generating function for these tableaux by a determinantal formula (Theorem 5.3) akin to the Jacobi–Trudi formula for Schur functions (and, like the latter, the proof will rely on Proposition 5.1). A pseudo-partition is similar to the concept of a partition, except it allows entries to increase by 1. The semistandard tableaux of a pseudo-partition shape are defined just like for partitions. The generating function in our determinantal formula is going to be a sum over the semistandard tableaux of a fixed pseudo-partition shape with given rightmost entries in each row. Later we will translate these tableaux into MLQs that yield a specific word when applied to 1^n.

Let us formalize these definitions. A *pseudo-partition* shall mean a k-tuple $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ of positive integers (for some $k \in \mathbb{N}$) such that

$$\text{each } i \in [k-1] \text{ satisfies } \lambda_i + 1 \geq \lambda_{i+1}.$$

For example, both $(5,3,4,2,2)$ and $(6,2,3,4,1)$ are pseudo-partitions.

The *diagram* $[\lambda]$ of a pseudo-partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ is defined as the set $\{(i,j) \in [k] \times \{1,2,3,\ldots\} \mid j \leq \lambda_i\}$. This diagram is drawn in the plane like usual Young diagrams, in English notation. For example, the diagram of the pseudo-partition $(3,2,3,1)$ is drawn as



If $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ is a pseudo-partition, then a *tableau of shape λ* is a map $T\colon [\lambda] \to \{1,2,3,\ldots\}$. For each such tableau T and each $(i,j) \in [\lambda]$, we refer to the value $T(i,j)$ as the *entry of T in cell (i,j)*. As usual, we represent a tableau T of shape λ by placing the entry $T(i,j)$ into the box corresponding to $(i,j) \in [\lambda]$. For example, the following are two tableaux of shape $(3,2,3,1)$:

$$\begin{array}{|c|c|c|}\hline 2 & 5 & 3 \\ \hline 1 & 1 & \multicolumn{1}{c}{} \\ \cline{1-2} 2 & 4 & 1 \\ \hline 7 & \multicolumn{2}{c}{} \\ \cline{1-1}\end{array} \quad \text{and} \quad \begin{array}{|c|c|c|}\hline 1 & 1 & 3 \\ \hline 2 & 3 & \multicolumn{1}{c}{} \\ \cline{1-2} 4 & 5 & 5 \\ \hline 5 & \multicolumn{2}{c}{} \\ \cline{1-1}\end{array}. \tag{5.2}$$

A tableau T of shape λ is said to be *semistandard* if and only if

- the entries of T are weakly increasing along each row (*i.e.*, we have $T(i,j_1) \leq T(i,j_2)$ whenever $(i,j_1) \in [\lambda]$ and $(i,j_2) \in [\lambda]$ satisfy $j_1 < j_2$);
- the entries of T are strictly increasing down each column (*i.e.*, we have $T(i_1,j) < T(i_2,j)$ whenever $(i_1,j) \in [\lambda]$ and $(i_2,j) \in [\lambda]$ satisfy $i_1 < i_2$).

For example, the second tableau in (5.2) is semistandard, while the first is not.

If $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ is a pseudo-partition, and if T is a tableau of shape λ, then the *surface* of T is defined to be the k-tuple $(s_1, s_2, \ldots, s_k)$, where s_i is the rightmost entry of the i-th row of T (that is, $s_i = T(i, \lambda_i)$).

If T is a tableau of shape λ whose entries belong to $[n]$, then the *weight* of T is defined as the monomial

$$\operatorname{wt}(T) := \prod_{(i,j) \in [\lambda]} x_{T(i,j)}$$

(that is, the product of x_d for d ranging over all entries of T). For example, the two tableaux in (5.2) have weights $x_1^3 x_2^2 x_3 x_4 x_5 x_7$ and $x_1^2 x_2 x_3^2 x_4 x_5^3$, respectively (assuming that $n \geq 7$). Let $\operatorname{SST}(\lambda, s)$ denote the set of all semistandard tableaux of shape λ and surface s.

Example 5.2. The diagram of the pseudo-partition $(3, 2, 3, 1)$ is drawn as

$$\begin{array}{|c|c|c|}\hline & & \\ \hline & & \multicolumn{1}{c}{} \\ \hline & & \\ \hline & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array}.$$

The following are two tableaux of shape $(3, 2, 3, 1)$:

$$\begin{array}{|c|c|c|}\hline 1 & 2 & 5 \\ \hline 2 & 3 & \multicolumn{1}{c}{} \\ \hline 3 & 4 & 5 \\ \hline 7 & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array} \qquad \text{and} \qquad \begin{array}{|c|c|c|}\hline 1 & 1 & 4 \\ \hline 2 & 3 & \multicolumn{1}{c}{} \\ \hline 4 & 5 & 5 \\ \hline 5 & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array}.$$

The right tableau is semistandard, while the left one is not (the two 5's in the rightmost column). These tableaux have surfaces $(5, 3, 5, 7)$ and $(4, 3, 5, 5)$, respectively, and weights $x_1 x_2^2 x_3^2 x_4 x_5^2 x_7$ and $x_1^2 x_2 x_3 x_4^2 x_5^3$, respectively.

We now state the main result of this subsection.

Theorem 5.3. *Let $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ be a pseudo-partition. Let $s = (s_1, s_2, \ldots, s_k)$ be a strictly increasing sequence of elements of $[n]$. Then,*

$$\sum_{T \in \mathrm{SST}(\lambda, s)} \mathrm{wt}(T) = \left(\prod_{i=1}^{k} x_{s_i} \right) \det \left(h_{\lambda_j - j + i - 1}(x_1, x_2, \ldots, x_{s_j}) \right)_{i,j \in [k]}.$$

We remark that the determinant in Theorem 5.3 is an instance of a *multi-Schur function* as defined in [LLPT18, (SCHUR.2.2)].

Proof of Theorem 5.3. Let us define two k-vertices $\mathbf{u} = (A_1, A_2, \ldots, A_k)$ and $\mathbf{v} = (B_1, B_2, \ldots, B_k)$ by

$$A_i = (k - i, 1) \qquad \text{and} \qquad B_i = (\lambda_i + k - i - 1, s_i).$$

The conditions of Proposition 5.1 are satisfied (since λ is a pseudo-partition and s is weakly increasing). Thus, Proposition 5.1 yields

$$\begin{aligned} \sum_{\mathbf{p} \in N(\mathbf{u}, \mathbf{v})} \mathrm{wt}(\mathbf{p}) &= \det \left(\sum_{p \in N(A_i, B_j)} \mathrm{wt}(p) \right)_{i,j \in [k]} \\ &= \det \left(h_{\lambda_j - j + i - 1}(x_1, x_2, \ldots, x_{s_j}) \right)_{i,j \in [k]}, \end{aligned} \tag{5.3}$$

where the second equality follows from the fact each $i, j \in [k]$ satisfy

$$\sum_{p \in N(A_i, B_j)} \mathrm{wt}(p) = h_{\lambda_j - j + i - 1}(x_1, x_2, \ldots, x_{s_j})$$

(by (5.1), applied to $A = A_i$ and $B = B_j$).

Next, let us define a bijection

$$\Phi : N(\mathbf{u}, \mathbf{v}) \to \mathrm{SST}(\lambda, s).$$

Indeed, let $(p_1, p_2, \ldots, p_k) \in N(\mathbf{u}, \mathbf{v})$ be arbitrary. Thus, each p_i is a path from A_i to B_i, and no two of the paths $p_1, p_2, \ldots, p_k$ have a vertex in common.

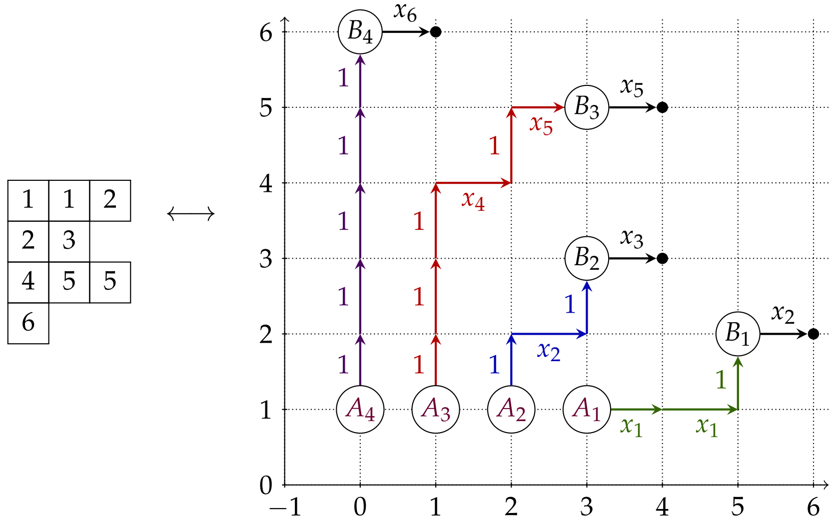


FIGURE 3. A semistandard tableau of surface $(2,3,5,6)$ (left) and the corresponding NILP (right). The black horizontal steps don't belong to the paths of the NILP; they stand for the last entries of the rows of the tableau.

For each $i \in [k]$, the path p_i is a path from $A_i = (k-i, 1)$ to $B_i = (\lambda_i + k - i - 1, s_i)$; thus, it must contain exactly $(\lambda_i + k - i - 1) - (k-i) = \lambda_i - 1$ east-steps.

Let $p_{i,1}, p_{i,2}, \ldots, p_{i,\lambda_i - 1}$ be the y-coordinates of these east-steps (from first to last). Also, set $p_{i,\lambda_i} = s_i$. Thus,

$$1 \leq p_{i,1} \leq p_{i,2} \leq \cdots \leq p_{i,\lambda_i - 1} \leq p_{i,\lambda_i} = s_i \tag{5.4}$$

for each $i \in [k]$. Note that for each $i \in [k]$,

$$\text{the path } p_i \text{ contains the vertices } (k-i+j-1, p_{i,j}) \text{ for all } j \in [\lambda_i] \tag{5.5a}$$

$$\text{and the vertices } (k-i+j, p_{i,j}) \text{ for all } j \in [\lambda_i - 1]\,. \tag{5.5b}$$

Now, let T be the tableau of shape λ that sends each $(i,j) \in [\lambda]$ to $p_{i,j}$. Then, the entries of T are weakly increasing along each row by (5.4). We shall next show that the entries of T are strictly increasing down each column.

First, we claim that if $i \in [k-1]$, then, for all $j \in [\lambda_i] \cap [\lambda_{i+1}]$, we have

$$p_{i,j} < p_{i+1,j}. \tag{5.6}$$

Proof. We will show (5.6) holds. Let $i \in [k-1]$. We must prove (5.6) for each $j \in [\lambda_i] \cap [\lambda_{i+1}]$. To do so, we assume the contrary, and pick the *smallest* j for which (5.6) fails. Thus, $p_{i,j} \geq p_{i+1,j}$. Assume that $j > 1$ (the argument for the case $j = 1$ is similar). Thus, the minimality of j forces $p_{i,j-1} < p_{i+1,j-1}$. Finally, (5.4) yields $p_{i+1,j-1} \leq p_{i+1,j}$. Hence, $p_{i,j-1} < p_{i+1,j-1} \leq p_{i+1,j} \leq p_{i,j}$. No two of the paths $p_1, p_2, \ldots, p_k$ have a vertex in common. Hence, p_i and p_{i+1} have no vertex in common. Note that (5.5a) yields that p_i contains the vertex $(k-i+j-1, p_{i,j})$. Also, (5.5b) (applied to $j-1$ instead of j) yields that p_i contains the vertex $(k-i+j-1, p_{i,j-1})$. However, the vertex $(k-i+j-1, p_{i+1,j})$

lies on the vertical line segment connecting the two vertices $\left(k-i+j-1, p_{i,j}\right)$ and $\left(k-i+j-1, p_{i,j-1}\right)$ (since $p_{i,j-1} \leq p_{i+1,j} \leq p_{i,j}$). Hence, p_i must contain the former vertex (since p_i contains the latter two vertices).

From (5.4), we obtain $p_{i,j} \leq s_i < s_{i+1}$ (since s is strictly increasing). If we had $j = \lambda_{i+1}$, then we would have $p_{i+1,j} = p_{i+1,\lambda_{i+1}} = s_{i+1}$ (by the definition of $p_{i+1,\lambda_{i+1}}$), which would contradict $p_{i+1,j} \leq p_{i,j} < s_{i+1}$. Hence, $j \neq \lambda_{i+1}$. Thus, $j \in [\lambda_{i+1} - 1]$ (since $j \in [\lambda_{i+1}]$). Therefore, (5.5b) (applied to $i+1$ instead of i) shows that p_{i+1} contains the vertex $\left(k-(i+1)+j, p_{i+1,j}\right) = \left(k-i+j-1, p_{i+1,j}\right)$. So we have shown that both p_i and p_{i+1} contain the vertex $\left(k-i+j-1, p_{i+1,j}\right)$. This contradicts the fact that p_i and p_{i+1} have no vertex in common.

We have assumed that $j > 1$; but the same argument works for $j = 1$ with minor changes (we now need to use $(k-i+j-1, 1)$ instead of $\left(k-i+j-1, p_{i,j-1}\right)$). Thus, we always obtain a contradiction. This contradiction shows that our assumption was false; hence, (5.6) is proven. ∎

Next, we claim that the entries of T are strictly increasing down each column.

Proof. Let $i_1, i_2 \in [k]$ be such that $i_1 < i_2$. Let $j \in [\lambda_{i_1}] \cap [\lambda_{i_2}]$. We must show that $T(i_1, j) < T(i_2, j)$. In other words, we must prove that $p_{i_1,j} < p_{i_2,j}$ (since $T(i,j) = p_{i,j}$ for all i).

If the column j of $[\lambda]$ has no gaps between row i_1 and row i_2 (that is, if we have $(i,j) \in [\lambda]$ for each $i \in \{i_1, i_1+1, \ldots, i_2\}$), then this follows from (5.6). Hence, without loss of generality, we assume that column j of $[\lambda]$ has at least one gap between row i_1 and row i_2. In other words, there exists some $q \in \{i_1, i_1+1, \ldots, i_2\}$ such that $(q,j) \notin [\lambda]$. Consider the *largest* such q. Clearly, $i_1 < q < i_2$. Also, $\lambda_q < j$ (since $(q,j) \notin [\lambda]$).

The maximality of q shows that column j of $[\lambda]$ has no gaps between row q and row i_2. Hence, (5.6) shows that $p_{q,j} < p_{i_2,j}$ (since $q < i_2$).

From (5.4), we obtain $p_{i_1,j} \leq s_{i_1} \leq s_q$ (since $i_1 < q$, but the sequence s is weakly increasing). Also, $p_{q,\lambda_q} = s_q$ (by the definition of p_{q,λ_q}), whence $s_q = p_{q,\lambda_q} \leq p_{q,j}$ (by (5.4), since $\lambda_q < j$). Hence, $p_{i_1,j} \leq s_q \leq p_{q,j} < p_{i_2,j}$. This proves our claim. ∎

Altogether, we thus know that T is a semistandard tableau of shape λ and surface s (since $T(i, \lambda_i) = p_{i,\lambda_i} = s_i$ for all $i \in [k]$). That is, $T \in \mathrm{SST}(\lambda, s)$. So let us set

$$\Phi(p_1, p_2, \ldots, p_k) = T.$$

Thus, the map Φ is defined.

It is easy to see that this map Φ is injective (indeed, $(p_1, p_2, \ldots, p_k)$ can be reconstructed from T, since the i-th row of T encodes the y-coordinates of all the east-steps of p_i) and surjective (indeed, the reverse of the above construction turns every $T \in \mathrm{SST}(\lambda, s)$ into a k-tuple $(p_1, p_2, \ldots, p_k)$ of paths p_i from A_i to B_i; furthermore, the strict increase of the entries of T down columns forces these paths $p_1, p_2, \ldots, p_k$ to have no vertices in common). Thus, Φ is a bijection. Moreover, it is easy to see that

$$\mathrm{wt}\big(\Phi(\mathbf{p})\big) = \left(\prod_{i=1}^{k} x_{s_i}\right) \mathrm{wt}(\mathbf{p})$$

for each $\mathbf{p} \in N(\mathbf{u}, \mathbf{v})$. Hence,

$$\sum_{T \in \mathrm{SST}(\lambda, s)} \mathrm{wt}(T) = \sum_{\mathbf{p} \in N(\mathbf{u}, \mathbf{v})} \left(\prod_{i=1}^{k} x_{s_i} \right) \mathrm{wt}(\mathbf{p}) = \left(\prod_{i=1}^{k} x_{s_i} \right) \sum_{\mathbf{p} \in N(\mathbf{u}, \mathbf{v})} \mathrm{wt}(\mathbf{p})$$
$$\overset{(5.3)}{=} \left(\prod_{i=1}^{k} x_{s_i} \right) \det \left(h_{\lambda_j - j + i - 1}(x_1, x_2, \ldots, x_{s_j}) \right)_{i,j \in [k]}.$$

□

We note that during the construction of the bijection Φ, the fixed surface s condition on the semistandard tableaux translates into requiring each path p_i to having a final arc $B_i \to (\lambda_i + k - i, s_i)$. Otherwise the rest of the proof is similar to the usual Jacobi–Trudi bijection relating semistandard tableaux and NILPs.

See Figure 3 for an example of the bijection Φ used in the proof of Theorem 5.3.

5.3. **Interlacing MLQs.** Let us introduce some further notations now.

First, we define two binary relations $\succeq$ and $\gg$ on the powerset of $[n]$:

- Given two subsets $A = \{a_1 > \cdots > a_\alpha\}$ and $B = \{b_1 > \cdots > b_\beta\}$ of $[n]$, we say that $A \succeq B$ if and only if $\alpha = \beta$ and every $1 \leq k \leq \alpha$ satisfies $a_k \geq b_k$.
- Given two subsets A and B of $[n]$, we say that $A \gg B$ if and only if every $a \in A$ and $b \in B$ satisfy $a > b$. (For nonempty A and B, this is equivalent to $\min A > \max B$).

For example, $\{10, 8\} \gg \{4, 6, 7\} \succeq \{4, 5, 7\} \succeq \{2, 5, 6\} \gg \{1\}$. Note that $A \gg \varnothing$ and $\varnothing \gg A$ for any subset A of $[n]$ (for vacuous reasons), so that $\gg$ is not a partial order (but it becomes a partial order if we forbid $\varnothing$).

The following criterion (proof left to the reader) will be useful:

Lemma 5.4. *Let $A, B \subseteq [n]$. We have $A \succeq B$ if and only if there exists a bijection $\phi \colon B \to A$ satisfying $\phi(b) \geq b$ for each $b \in B$.*

Fix some $\ell \in \mathbb{N}$ and a type $\mathbf{m} = (m_1, m_2, \ldots, m_\ell, 0, 0, \ldots)$ with $\leq \ell$ classes. Assume that $m_i > 0$ for all $i \in [\ell - 1]$ (but m_ℓ may be 0).

Our next few definitions concern MLQs. Consider an (ordinary) MLQ $\mathbf{q} = (q_1, \ldots, q_{\ell-1})$ of type $\mathbf{m}$. Thus, for each $i \in [\ell - 1]$, we have $|q_i| = p_i(\mathbf{m}) = m_1 + m_2 + \cdots + m_i$. Hence, for each $i \in [\ell - 1]$, we can subdivide the set q_i into i blocks: the block containing the largest m_1 elements; the block containing the next-largest m_2 elements; and so on, until the block containing the smallest m_i elements. Denote these i blocks by $q_i^{(1)}, q_i^{(2)}, \ldots, q_i^{(i)}$, respectively. Pictorially, we can thus write q_i as

$$\{\underbrace{a_1 > \cdots > a_{p_1(\mathbf{m})}}_{= q_i^{(1)}} > \underbrace{a_{p_1(\mathbf{m})+1} > \cdots > a_{p_2(\mathbf{m})}}_{= q_i^{(2)}} > \cdots > \underbrace{a_{p_{i-1}(\mathbf{m})+1} > \cdots > a_{p_i(\mathbf{m})}}_{= q_i^{(i)}}\}.$$

Thus, $q_i^{(1)}, q_i^{(2)}, \ldots, q_i^{(i)}$ are pairwise disjoint nonempty subsets of $[n]$ satisfying

$$q_i = q_i^{(1)} \cup q_i^{(2)} \cup \cdots \cup q_i^{(i)} \qquad \text{and} \tag{5.7a}$$

$$q_i^{(1)} \gg q_i^{(2)} \gg \cdots \gg q_i^{(i)} \qquad \text{and} \tag{5.7b}$$

$$\left| q_i^{(j)} \right| = m_j \text{ for all } j \in [i] . \tag{5.7c}$$

Thus, we have defined nonempty queues $q_i^{(j)} \subseteq [n]$ for all $i \in [\ell - 1]$ and $j \in [i]$ whenever $\mathbf{q} = (q_1, \ldots, q_{\ell-1})$ is an MLQ of type $\mathbf{m}$.

Example 5.5. Consider $n = 15$. Let $\mathbf{m} = (4, 2, 2)$ and $\mathbf{q} = (q_1, q_2, q_3)$, where

$$q_1 = \{2, 4, 9, 12\}, \qquad q_2 = \{1, 5, 6, 8, 12, 15\}, \qquad q_3 = \{1, 2, 4, 5, 8, 9, 13, 14\}.$$

Therefore, we have

$$\begin{array}{ccc} & q_1^{(1)} = \{2,4,9,12\}, & \\ q_2^{(2)} = \{1,5\}, & & q_2^{(1)} = \{6,8,12,15\}, \\ q_3^{(3)} = \{1,2\}, & q_3^{(2)} = \{4,5\}, & q_3^{(1)} = \{8,9,13,14\}. \end{array}$$

Similar to the graveyard diagram (2.3), we define the graveyard diagram of an MLQ $\mathbf{q} = (q_1, q_2, \ldots, q_{\ell-1})$ as a matrix with $\ell - 1$ rows, whose entries are circles and squares; its row i has each element $p \in q_i$ represented as a circle at position p labeled by the letter $\left(q_i(\cdots q_1(1^n)\cdots)\right)_p$, and each element of $[n] \setminus q_i$ represented as an unlabeled square (*i.e.*, we suppress filling the squares with an $i+1$). The $\mathbf{q}$ in this example is represented by the following graveyard diagram:

$$\begin{array}{ccccccccccccccc} \square & 1 & \square & 1 & \square & \square & \square & \square & 1 & \square & \square & 1 & \square & \square & \square \\ 2 & \square & \square & \square & 1 & 1 & \square & 2 & \square & \square & \square & 1 & \square & \square & 1 \\ 1 & 2 & \square & 3 & 1 & \square & \square & 1 & 2 & \square & \square & \square & 1 & 3 & \square \end{array} .$$

Definition 5.6. We say that the MLQ $\mathbf{q} = (q_1, \ldots, q_{\ell-1})$ is *interlacing* if each $i \in \{2, 3, \ldots, \ell - 1\}$ satisfies

$$q_i^{(1)} \succeq q_{i-1}^{(1)} \gg q_i^{(2)} \succeq q_{i-1}^{(2)} \gg q_i^{(3)} \succeq q_{i-1}^{(3)} \gg \cdots \gg q_i^{(i)} \tag{5.8}$$

(that is, $q_i^{(j)} \succeq q_{i-1}^{(j)} \gg q_i^{(j+1)}$ for all $j \in [i-1]$).

Example 5.7. Consider $n = 15$. Let $\mathbf{m} = (3, 2, 4, 0, \ldots)$ and $\mathbf{q} = (q_1, q_2, q_3)$, where

$$q_1 = \{9, 12, 13\}, \qquad q_2 = \{7, 8, 11, 12, 14\}, \qquad q_3 = \{1, 3, 5, 6, 8, 10, 11, 14, 15\}.$$

Therefore, we have

$$\begin{array}{ccc} & q_1^{(1)} = \{9,12,13\}, & \\ q_2^{(2)} = \{7,8\}, & & q_2^{(1)} = \{11,12,14\}, \\ q_3^{(3)} = \{1,3,5,6\}, & q_3^{(2)} = \{8,10\}, & q_3^{(1)} = \{11,14,15\}. \end{array}$$

Thus $\mathbf{q}$ is interlacing. In this case, the elements of $q_i^{(j)}$ will be labeled by j (as we shall see more generally in the proof of Lemma 5.13 below), and so the graveyard diagram of $\mathbf{q}$ is

$$\begin{array}{ccccccccccccccc}
\square & \square & \square & \square & \square & \square & \square & \square & 1 & 1 & \square & \square & 1 & \square & \square \\
\square & \square & \square & \square & \square & \square & 2 & 2 & \square & \square & 1 & 1 & \square & 1 & \square \\
3 & \square & 3 & \square & 3 & 3 & \square & 2 & \square & 2 & 1 & \square & \square & 1 & 1
\end{array}.$$

Define a pseudo-partition

$$\lambda^{\mathbf{m}} := (\underbrace{1, 1, \ldots, 1}_{m_{\ell-1} \text{ times}}, \underbrace{2, 2, \ldots, 2}_{m_{\ell-2} \text{ times}}, \ldots, \underbrace{\ell-1, \ell-1, \ldots, \ell-1}_{m_1 \text{ times}}). \tag{5.9}$$

Thus, the columns of the diagram $[\lambda^{\mathbf{m}}]$ are aligned to the bottom and have lengths $p_{\ell-1}(\mathbf{m}), p_{\ell-2}(\mathbf{m}), \ldots, p_1(\mathbf{m})$ (from left to right).

For the following lemma, note that $\mathbf{m}$ is fixed but $\mathbf{q}$ is not.

Lemma 5.8. *Let ℓ, $\mathbf{m}$, and m_i be as above. Then, there is an injection*

$$P\colon \{\text{MLQs of type } \mathbf{m}\} \to \{\text{tableaux of shape } \lambda^{\mathbf{m}}\}$$

with the following properties:

(a) *We have $\operatorname{wt}(P(\mathbf{q})) = \operatorname{wt}(\mathbf{q})$ for each MLQ $\mathbf{q}$ of type $\mathbf{m}$.*
(b) *If $\mathbf{q} = (q_1, q_2, \ldots, q_{\ell-1})$ is an MLQ of type $\mathbf{m}$, then the surface of $P(\mathbf{q})$ is the list of all elements of $q_{\ell-1}$ in increasing order.*
(c) *The map P restricts to a bijection*

$$\overline{P}\colon \{\text{interlacing MLQs of type } \mathbf{m}\} \to \{\text{semistandard tableaux of shape } \lambda^{\mathbf{m}}\}.$$

(In the border case $\ell = 1$, we interpret q_0 as the empty set whenever $\mathbf{q} = (q_1, q_2, \ldots, q_{\ell-1})$ is an MLQ.)

Proof. We define a map P as follows. Let $\mathbf{q} = (q_1, q_2, \ldots, q_{\ell-1})$ be any MLQ of type $\mathbf{m}$. Let $\widetilde{q}_i^{(j)}$ denote the entries of $q_i^{(j)}$ written vertically in a column, strictly increasing from top to bottom. For example, if $q_i^{(j)} = \{2, 5, 6\}$, then

$$\widetilde{q}_i^{(j)} = \begin{array}{|c|} \hline 2 \\ \hline 5 \\ \hline 6 \\ \hline \end{array}.$$

We construct $P(\mathbf{q})$ as the following tableau of shape $\lambda^{\mathbf{m}}$:

$$
\begin{array}{r|c|c|c|c|c|c|}
\cline{2-2}
m_{\ell-1} \text{ rows } \left\{ \vphantom{\widetilde{q}^{(\ell-1)}_{\ell-1}} \right. & \widetilde{q}^{(\ell-1)}_{\ell-1} \\
\cline{2-3}
m_{\ell-2} \text{ rows } \left\{ \vphantom{\widetilde{q}^{(\ell-1)}_{\ell-1}} \right. & \widetilde{q}^{(\ell-2)}_{\ell-2} & \widetilde{q}^{(\ell-2)}_{\ell-1} \\
\cline{2-4}
m_{\ell-3} \text{ rows } \left\{ \vphantom{\widetilde{q}^{(\ell-1)}_{\ell-1}} \right. & \widetilde{q}^{(\ell-3)}_{\ell-3} & \widetilde{q}^{(\ell-3)}_{\ell-2} & \widetilde{q}^{(\ell-3)}_{\ell-1} \\
\cline{2-5}
\vdots & \vdots & \vdots & \vdots & \ddots \\
\cline{2-6}
m_2 \text{ rows } \left\{ \vphantom{\widetilde{q}^{(\ell-1)}_{\ell-1}} \right. & \widetilde{q}^{(2)}_{2} & \widetilde{q}^{(2)}_{3} & \widetilde{q}^{(2)}_{4} & \cdots & \widetilde{q}^{(2)}_{\ell-1} \\
\cline{2-7}
m_1 \text{ rows } \left\{ \vphantom{\widetilde{q}^{(\ell-1)}_{\ell-1}} \right. & \widetilde{q}^{(1)}_{1} & \widetilde{q}^{(1)}_{2} & \widetilde{q}^{(1)}_{3} & \cdots & \widetilde{q}^{(1)}_{\ell-2} & \widetilde{q}^{(1)}_{\ell-1} \\
\cline{2-7}
\end{array}
\tag{5.10}
$$

Formally speaking, this is the tableau of shape $\lambda^{\mathbf{m}}$ whose $\ell-1$ columns are given as follows: For each $j \in [\ell-1]$, the j-th column consists of the columns $\widetilde{q}^{(\ell-j)}_{\ell-1}, \widetilde{q}^{(\ell-j-1)}_{\ell-2}, \ldots, \widetilde{q}^{(1)}_{j}$ stacked atop each other (with $\widetilde{q}^{(\ell-j)}_{\ell-1}$ at the very top, $\widetilde{q}^{(\ell-j-1)}_{\ell-2}$ coming next under it, and so on). Note that for each $i \in [\ell-1]$, the parts $\widetilde{q}^{(i)}_{i}, \widetilde{q}^{(i)}_{i+1}, \ldots, \widetilde{q}^{(i)}_{\ell-1}$ of columns $1, 2, \ldots, \ell-i$ align with each other horizontally (and together form the topmost m_i among the bottommost $m_1 + m_2 + \cdots + m_i$ rows of $P(\mathbf{q})$).

Thus, the map $P \colon \{\text{MLQs of type } \mathbf{m}\} \to \{\text{tableaux of shape } \lambda^{\mathbf{m}}\}$ is defined. This map P is an injection, because the MLQ $\mathbf{q}$ can be recovered from the tableau $P(\mathbf{q})$ (indeed, each of the elements of each of the queues of $\mathbf{q}$ lands in some predictable cell of $P(\mathbf{q})$).

We shall now prove the three properties we claimed about this injection P to complete the proof.

Property (a) is clear, since the entries of $q_1, q_2, \ldots, q_{\ell-1}$ are in 1-to-1 correspondence with the entries of the tableau $P(\mathbf{q})$.

Furthermore, the surface of $P(\mathbf{q})$ is simply

$$q^{(\ell-1)}_{\ell-1} \cup q^{(\ell-2)}_{\ell-1} \cup \cdots \cup q^{(2)}_{\ell-1} \cup q^{(1)}_{\ell-1} = q_{\ell-1}$$

written in increasing order from top to bottom because of (5.7a) and (5.7b). In other words, the surface of $P(\mathbf{q})$ is the list of all elements of $q_{\ell-1}$ in increasing order. This proves (b).

Let $\mathbf{q}$ be an MLQ of type $\mathbf{m}$. Recall that the columns $\widetilde{q}^{(j)}_{i}$ are strictly increasing. Hence, from (5.10), we see the following:

- The entries of the tableau $P(\mathbf{q})$ are weakly increasing along each row if and only if all $i \in [\ell-1]$ and $j \in [i-1]$ satisfy $q^{(j)}_{i} \succeq q^{(j)}_{i-1}$.
- The entries of the tableau $P(\mathbf{q})$ are strictly increasing down each column if and only if all $i \in [\ell-1]$ and $j \in [i-1]$ satisfy $q^{(j)}_{i-1} \gg q^{(j+1)}_{i}$. (Here, we are also tacitly using the fact that the sets $q^{(j)}_{i}$ are nonempty. This ensures

that the southern neighbor of a cell in $\widetilde{q}_i^{(j)}$ belongs either to $\widetilde{q}_i^{(j)}$ again or to $\widetilde{q}_{i-1}^{(j-1)}$, rather than (say) to $\widetilde{q}_{i-2}^{(j-2)}$.)

Combining these two observations, we conclude that the tableau $P(\mathbf{q})$ is semistandard if and only if the sets $q_i^{(j)}$ satisfy (5.8). In other words, the tableau $P(\mathbf{q})$ is semistandard if and only if $\mathbf{q}$ is interlacing. Moreover, it is clear that given any semistandard tableau T of shape $\lambda^{\mathbf{m}}$, we can construct an interlacing MLQ $\mathbf{q}$ of type $\mathbf{m}$ satisfying $P(\mathbf{q}) = T$. (Namely, we can construct this $\mathbf{q}$ by recovering the sets $q_i^{(j)}$ from the appropriate cells of T in (5.10) and combining them to obtain queues q_i and an MLQ $\mathbf{q}$.) Hence, the map P restricted to interlacing MLQs is surjective onto the set of semistandard tableaux of shape $\lambda^{\mathbf{m}}$, and since P is injective, we have a bijection. This proves (c). $\square$

Corollary 5.9. *Let ℓ, $\mathbf{m}$, and m_i be as above. Let $k \in \mathbb{N}$ and $\lambda_1, \lambda_2, \ldots, \lambda_k$ be such that $\lambda^{\mathbf{m}} = (\lambda_1, \lambda_2, \ldots, \lambda_k)$. Let $S = \{s_1 < s_2 < \cdots < s_k\}$ be a k-queue. Then,*

$$\sum_{\substack{\mathbf{q}=(q_1,q_2,\ldots,q_{\ell-1}) \text{ is an interlacing} \\ \text{MLQ of type } \mathbf{m} \text{ with } q_{\ell-1}=S}} \operatorname{wt}(\mathbf{q}) = \left(\prod_{i=1}^{k} x_{s_i}\right) \det\left(h_{\lambda_j - j + i - 1}(x_1, x_2, \ldots, x_{s_j})\right)_{i,j\in[k]}.$$

Proof. Consider the bijection $\overline{P}$ from Lemma 5.8(c), and recall the properties (a) and (b). Hence, we can substitute $\overline{P}(\mathbf{q}) = P(\mathbf{q})$ for T in the sum on the left hand side of Theorem 5.3 (applied to $\lambda = \lambda^{\mathbf{m}}$). We thus obtain

$$\begin{aligned}
\sum_{T\in\mathrm{SST}(\lambda^{\mathbf{m}},s)} \operatorname{wt}(T) &= \sum_{\substack{\mathbf{q}=(q_1,q_2,\ldots,q_{\ell-1}) \text{ is an interlacing} \\ \text{MLQ of type } \mathbf{m} \text{ such that} \\ \text{the surface of } \overline{P}(\mathbf{q}) \text{ is } s}} \operatorname{wt}(P(\mathbf{q})) \\
&= \sum_{\substack{\mathbf{q}=(q_1,q_2,\ldots,q_{\ell-1}) \text{ is an interlacing} \\ \text{MLQ of type } \mathbf{m} \text{ such that} \\ \text{the list of all elements of } q_{\ell-1} \text{ in} \\ \text{increasing order is } s}} \operatorname{wt}(\mathbf{q}) \\
&= \sum_{\substack{\mathbf{q}=(q_1,q_2,\ldots,q_{\ell-1}) \text{ is an interlacing} \\ \text{MLQ of type } \mathbf{m} \text{ with } q_{\ell-1}=S}} \operatorname{wt}(\mathbf{q})
\end{aligned}$$

where the second equality follows from Property (a) and Property (b) and the last equality is since the list of all elements of $q_{\ell-1}$ in increasing order is s if and only if $q_{\ell-1} = S$. Therefore, the claim of Corollary 5.9 follows from Theorem 5.3. $\square$

Next, we shall connect the interlacingness of an MLQ with its action on the word 1^n. First, we need another notion: If $u \in \mathcal{W}_n$ and $t \in \mathbb{N}$, then we say that u is *weakly decreasing up to level t* if and only if every two sites $i < j$ in $[n]$ satisfying $u_j \leq t$ must satisfy $u_i \geq u_j$. Equivalently, u is weakly decreasing up to level t if and only if u becomes weakly decreasing when all letters larger than t are removed. For example, the word 5455433252215 is weakly decreasing up to level 4 (and up to any level ≤ 4). We also need the following auxiliary lemma about what queues can yield weakly decreasing words.

Lemma 5.10. *Let $u \in \mathcal{W}_n$ and $t > 0$. Let q be a queue. Assume that the word $q(u)$ has at least one letter equal to t, is weakly decreasing up to level t, and has exactly $|q|$ letters that are at most t. Then:*

(a) *The word u is weakly decreasing up to level $t-1$.*
(b) *For each $h \in [t-1]$, we have*

$$\left\{p \in [n] \mid (q(u))_p = h\right\} \succeq \{p \in [n] \mid u_p = h\} \gg \left\{p \in [n] \mid (q(u))_p = h+1\right\}.$$

Example 5.11.

- Let $n = 9$, $u = 322131133$, $t = 3$ and $q = \{1,2,3,5,6,7\}$. Then, $q(u) = 322411144$ satisfies all assumptions of Lemma 5.10. Thus, part (a) of the lemma says that u is weakly decreasing up to level 2 (which is evident). Part (b) of the lemma, applied to $h = 1$, says that
$$\{5,6,7\} \succeq \{4,6,7\} \gg \{2,3\}.$$
- The assumption that $q(u)$ has at least one letter equal to t cannot be removed from Lemma 5.10. Indeed, if $n = 4$, $u = 3312$, $t = 3$ and $q = \{1,3\}$, then $q(u) = 2414$ satisfies all assumptions except for this one, but the claim of Lemma 5.10(a) does not hold.
- The assumption that $q(u)$ is weakly decreasing up to level t cannot be removed from Lemma 5.10. Indeed, if $n = 4$, $u = 1233$, $t = 3$ and $q = \{1,2,3\}$, then $q(u) = 1234$ satisfies all assumptions except for this one, but the claim of Lemma 5.10(a) does not hold.
- The assumption that $q(u)$ has exactly $|q|$ letters that are at most t cannot be removed from Lemma 5.10. Indeed, if $n = 5$, $u = 32112$, $t = 3$ and $q = \{2,3,5\}$, then $q(u) = 32141$ satisfies all assumptions except for this one, but the claim of Lemma 5.10(a) does not hold.

Proof of Lemma 5.10. Fix a permutation $(i_1, i_2, \ldots, i_n)$ of $(1, 2, \ldots, n)$ such that $u_{i_1} \leq u_{i_2} \leq \cdots \leq u_{i_n}$. From the definition of $q(u)$ and since $q(u)$ has precisely $|q|$ letters at most t, we see that all letters set during Phase II of the algorithm computing $q(u)$ are at most t, while all letters set during Phase I are strictly greater than t (see also Remark 2.3). Therefore, it is sufficient to consider the letters set during Phase II. We first show that no "wrapping around the cycle" can occur during Phase II. Let j_κ denote the site set in the κ-th step of Phase II (that is, the site j found at the time when $i = i_\kappa$). Hence, $\left\{j_1, j_2, \ldots, j_{|q|}\right\} = q$.

Claim 5.12. *We have $j_\kappa \geq i_\kappa$ for all $\kappa \leq |q|$ satisfying $u_{i_\kappa} \leq t-1$.*

Proof. Assume the contrary. Thus, there exists a $\kappa \leq |q|$ satisfying $u_{i_\kappa} \leq t-1$ such that $j_\kappa < i_\kappa$ (*i.e.*, the path of u_{i_κ} "wraps around the cycle"). Without loss of generality, assume κ is minimal. Then either $j_\kappa = \min q$ or $\min q$ has already been set on a previous step of Phase II, and in either case, we have $(q(u))_{\min q} < t$. Since $q(u)$ contains at least one t (which is set during Phase II), there exists an $h > \kappa$ such that $(q(u))_{j_h} = t$ with $j_h > \min q$. However, this contradicts that $q(u)$ is weakly decreasing up to level t. ■

Next, suppose Lemma 5.10(a) does not hold. Thus, there exists a k such that $i_{k'} < i_k$ and $u_{i_{k'}} < u_{i_k} \leq t-1$ for some $k' < k$. Without loss of generality, assume k is minimal with this property. Since $q(u)_{j_\kappa} = u_{i_\kappa}$ for all $\kappa \leq |q|$, we thus have

$q(u)_{j_{k'}} < q(u)_{j_k} \leq t-1$. From our assumption that $q(u)$ is weakly decreasing up to level t, we have $j_{k'} > j_k$. Claim 5.12 shows that $j_k \geq i_k$. Hence, $i_{k'} < i_k \leq j_k \leq j_{k'}$. Thus, during Phase II, at the time when $i = i_{k'}$, the first site j weakly to the right of i such that $j \in q$ and $(q(u))_j$ is not set cannot be $j_{k'}$ (because j_k comes before it, and $(q(u))_{i_k}$ is still not set). This contradicts the definition of $j_{k'}$. This proves Lemma 5.10(a).

Now we prove Lemma 5.10(b). Let $h \in [t-1]$, and set

$$\begin{aligned} A &= \left\{ p \in [n] \mid \big(q(u)\big)_p = h \right\}, \\ B &= \left\{ p \in [n] \mid u_p = h \right\}, \\ C &= \left\{ p \in [n] \mid \big(q(u)\big)_p = h+1 \right\}. \end{aligned}$$

We first show $A \succeq B$. Define a map $\phi \colon B \to A$ by $\phi(i_k) = j_k$, where $i_k \in B$. The map ϕ is well-defined (*i.e.*, we have $j_k \in A$ for all applicable k) and bijective from the definition of Phase II (see also Remark 2.3 for why $|A| = |B|$). We have $\phi(i_k) = j_k \geq i_k$ by Claim 5.12. Hence $A \succeq B$ by Lemma 5.4.

To prove $B \gg C$, assume the contrary. Thus, there exist $b \in B$ and $c \in C$ such that $b \leq c$. Let k be minimal such that $i_k \in B$ and $i_k \leq c$. Since $q(u)$ is weakly decreasing up to level t and $h < t$, we must have $A \gg C$. Hence, we have $i_k \leq c < j_k$ since $j_k \in A$. But during Phase II, at the time when $i = i_k$, the letter $(q(u))_c$ is not set yet (since $(q(u))_c = h+1$ is larger than $u_{i_k} = h$). Thus, at this time, the first site j weakly to the right of i such that $j \in q$ and $(q(u))_j$ is not set cannot be j_k (because c comes before it, and $(q(u))_c$ is still not set). This contradicts the definition of j_k. This shows $B \gg C$. □

Now we prove our main technical lemma:

Lemma 5.13. *Let ℓ be a positive integer. Let $\mathbf{m} = (m_1, m_2, \ldots, m_\ell, 0, 0, \ldots)$ be such that $m_i > 0$ for all $1 \leq i < \ell$. Let $\mathbf{q} = (q_1, q_2, \ldots, q_{\ell-1})$ be an ordinary MLQ with $\ell - 1$ queues. Then, the following are equivalent:*

(α) *The word $\mathbf{q}(1^n)$ has type $\mathbf{m}$ and is weakly decreasing up to level $\ell - 1$.*
(β) *The MLQ $\mathbf{q}$ has type $\mathbf{m}$ and is interlacing.*

Example 5.14.

- We require that $m_i > 0$ for all $1 \leq i \leq \ell - 1$. To see why, consider $u = 2414$, which is of type $\mathbf{m} = (1, 1, 0, 2, 0, \ldots)$ and is weakly decreasing up to level 3. However, the MLQ $\mathbf{q} = (\{2\}, \{3, 4\}, \{1, 3\})$ is not interlacing but $\mathbf{q}(1111) = u$:

$$\begin{array}{cccc} \boxed{2} & 1 & \boxed{2} & \boxed{2} \\ \boxed{3} & \boxed{3} & 1 & 2 \\ 2 & \boxed{4} & 1 & \boxed{4} \end{array}$$

- The interlacing of the MLQ (equivalently the weakly decreasing up to level $\ell - 1$) is necessary as seen by Example 5.5.

Proof of Lemma 5.13. We prove the claim by induction on ℓ. The base case of $\ell = 1$ is trivial. Thus, we assume the claim holds for ℓ. Let $\mathbf{m} = (m_1, m_2, \ldots, m_{\ell+1}, 0, \ldots)$

with $m_i > 0$ for all $1 \leq i < \ell + 1$. Let $\mathbf{q} = (q_1, q_2, \ldots, q_\ell)$ be an ordinary MLQ. The word $u := \mathbf{q}(1^n)$ has type $\mathbf{m}$ if and only if $\mathbf{q}$ has type $\mathbf{m}$ by Lemma 2.10. Thus we can assume that both of these statements hold, and it is remains to show that $u = \mathbf{q}(1^n)$ is weakly decreasing up to level ℓ if and only if $\mathbf{q}$ is interlacing. We shall prove the $\Longrightarrow$ and $\Longleftarrow$ directions separately.

First, let us prepare. Let $\mathbf{q}' := (q_1, \ldots, q_{\ell-1})$ and $u' := \mathbf{q}'(1^n)$. Thus, $u = \mathbf{q}(1^n) = q_\ell(u')$. Both the word u' and the ordinary MLQ $\mathbf{q}'$ have type

$$\mathbf{m}' := (m_1, m_2, \ldots, m_{\ell-1}, n - p_{\ell-1}(\mathbf{m}), 0, \ldots)$$

since $|q_\ell| = p_\ell(\mathbf{m})$.

Recall the definition of $q_i^{(h)}$ given by (5.7).

$\Longrightarrow$: Suppose $u = \mathbf{q}(1^n)$ is weakly decreasing up level ℓ. Thus, Lemma 5.10(a) (applied to u', q_ℓ and ℓ instead of u, q and t) shows that u' is weakly decreasing up to level $\ell - 1$ (since $u = q_\ell(u')$ has type $\mathbf{m}$ with $m_\ell > 0$ and $p_\ell(\mathbf{m}) = |q_\ell|$). Hence, by our induction assumption, $\mathbf{q}'$ is interlacing. Since u' has type $\mathbf{m}'$ and is weakly decreasing up to level $\ell - 1$, we see that $q_{\ell-1}^{(h)} = \{p \mid u'_p = h\}$ for all $h \in [\ell - 1]$. Similarly, we have $q_\ell^{(h)} = \{p \mid u_p = h\}$ for all $h \in [\ell]$. Therefore, by Lemma 5.10(b), we obtain $q_\ell^{(h)} \succeq q_{\ell-1}^{(h)} \gg q_\ell^{(h+1)}$ for each $h \in [\ell - 1]$. Hence, $\mathbf{q}$ is interlacing.

$\Longleftarrow$: Suppose the MLQ $\mathbf{q}$ is interlacing. Thus, $\mathbf{q}'$ is also interlacing. Hence, by our induction assumption, the word $u' = \mathbf{q}'(1^n)$ is weakly decreasing up to level $\ell - 1$; recall that this word has type $\mathbf{m}'$. Hence, writing u' as $u'_1 u'_2 \cdots u'_n$, we have $q_{\ell-1}^{(h)} = \left\{p \mid u'_p = h\right\}$ for all $h \in [\ell - 1]$.

Now, we compute $u = q_\ell(u')$ using the algorithm constructing $q_\ell(u')$, choosing the permutation $(i_1, i_2, \ldots, i_n)$ of $(1, 2, \ldots, n)$ in such a way that $(u_{i_1}, 1) \leq (u_{i_2}, 2) \leq \cdots \leq (u_{i_n}, n)$ in lexicographic order (thus, in Phase II, we pick our sites i in order of increasing letters in u, while breaking ties by moving left to right). For each $k \in [n]$, we let j_k be the site j found at the step at which $i = i_k$.

The interlacingness relation $q_\ell^{(h)} \succeq q_{\ell-1}^{(h)} \gg q_\ell^{(h+1)}$, combined with the fact that $q_{\ell-1}^{(h)} = \{p \mid u'_p = h\}$ for all $h \in [\ell - 1]$, reveals that throughout Phase II of our algorithm, whenever the site i_k belongs to $q_{\ell-1}^{(h)}$, the corresponding site j_k will belong to $q_\ell^{(h)}$ (and, more precisely, as the i in our algorithm runs through the sites in $q_{\ell-1}^{(h)}$ from left to right, the j will run through the sites in $q_\ell^{(h)}$ from left to right); indeed, the $q_\ell^{(h)} \succeq q_{\ell-1}^{(h)}$ part guarantees that each site in $q_{\ell-1}^{(h)}$ is reachable from the corresponding site in $q_\ell^{(h)}$ by moving right, whereas the $q_{\ell-1}^{(h)} \gg q_\ell^{(h+1)}$ part ensures that the search for j will actually reach this site (rather than already stopping at some other site in q_ℓ further left of it). (Strictly speaking, this argument is a strong induction on k, as we are using that all the previous sites have been correctly dispatched.) Hence, the converse also holds: If $j_k \in q_\ell^{(h)}$, then $i_k \in q_{\ell-1}^{(h)}$ (since $q_\ell^{(h)} \succeq q_{\ell-1}^{(h)}$ yields $\left|q_\ell^{(h)}\right| = \left|q_{\ell-1}^{(h)}\right|$).

As a consequence, we have $u_p = h$ for each $p \in q_\ell^{(h)}$ for each $h \in [\ell - 1]$ (since $p = j_k$ for some $k \in [n]$ found during Phase II of our algorithm, and thus $u_p =$

$u_{j_k} = (q_\ell(u'))_{j_k} = u'_{i_k} = h$ since the corresponding i_k lies in $q_{\ell-1}^{(h)} = \{p \mid u'_p = h\}$). Hence, $q_\ell^{(h)} = \{p \mid u_p = h\}$ for each $h \in [\ell - 1]$, and thus also for $h = \ell$ (by the exclusion principle: the letters of u that are $\leq \ell$ appear in the positions $p \in q_\ell$). Therefore, u is weakly decreasing up to level ℓ. $\square$

5.4. **Proof of Theorem 3.9.** We are now ready to prove Theorem 3.9.

Proof of Theorem 3.9. Write the r-tuple $(v_1, v_2, \ldots, v_r)$ in the form

$$(v_1, v_2, \ldots, v_r) = (\underbrace{\ell-1, \ldots, \ell-1}_{m_{\ell-1} \text{ times}}, \underbrace{\ell-2, \ldots, \ell-2}_{m_{\ell-2} \text{ times}}, \ldots, \underbrace{1, \ldots, 1}_{m_1 \text{ times}}) \tag{5.11}$$

for some $m_1, m_2, \ldots, m_{\ell-1} > 0$ (we can do this, since $\{v_1, v_2, \ldots, v_r\} = [\ell - 1]$ and $v_1 \geq v_2 \geq \cdots \geq v_r$). Thus, the first $m_{\ell-1}$ of the entries of the r-tuple $(v_1 \geq v_2 \geq \cdots \geq v_r)$ equal $\ell - 1$; the next $m_{\ell-2}$ of its entries equal $\ell - 2$; the next $m_{\ell-3}$ of its entries equal $\ell - 3$; and so on. Also set $m_\ell = n - r$, and define a type $\mathbf{m} = (m_1, m_2, \ldots, m_\ell, 0, 0, \ldots)$. Then, the word $u(v)$ has type $\mathbf{m}$ (since the entries of $u(v)$ are precisely $v_1, v_2, \ldots, v_r$ and also $n - r$ entries equal to ℓ). Additionally, the word $u(v)$ is packed (since $m_1, m_2, \ldots, m_{\ell-1} > 0$). Furthermore, the definition of $\mathbf{m}$ shows that $r = m_1 + m_2 + \cdots + m_{\ell-1} = p_{\ell-1}(\mathbf{m})$. Each $i \in [\ell]$ satisfies

$$m_i = \begin{cases} \left|\{j \in [r] \mid v_j = i\}\right| & \text{if } i < \ell, \\ n - r & \text{if } i = \ell \end{cases}$$

(by (5.11) and the definition of m_ℓ).

The word $u(v)$ is weakly decreasing up to level $\ell - 1$ since $v_1 \geq v_2 \geq \cdots \geq v_r$ and $\{v_1, v_2, \ldots, v_r\} = [\ell - 1]$. Moreover, the letters $\leq \ell - 1$ of this word lie in the positions $j \in B$. Therefore, for an MLQ $\mathbf{q} = (q_1, \ldots, q_{\ell-1})$ of type $\mathbf{m}$, the following are equivalent:

(1) We have $u(v) = \mathbf{q}(1^n)$.
(2) The word $\mathbf{q}(1^n)$ has type $\mathbf{m}$ and is weakly decreasing up to level $\ell - 1$ and its letters $\leq \ell - 1$ lie in the positions $j \in B$.
(3) The word $\mathbf{q}(1^n)$ has type $\mathbf{m}$ and is weakly decreasing up to level $\ell - 1$ and we have $q_{\ell-1} = B$.
(4) The MLQ $\mathbf{q}$ has type $\mathbf{m}$ and is interlacing, and we have $q_{\ell-1} = B$.

It is straightforward to see (1) $\iff$ (2) $\iff$ (3), and we have (3) $\iff$ (4) by Lemma 5.13.

Define the pseudo-partition $\lambda^{\mathbf{m}}$ by (5.9). Comparing (5.9) with (5.11), we obtain

$$\lambda^{\mathbf{m}} = (\ell - v_1, \ell - v_2, \ldots, \ell - v_r) = (\gamma_1, \gamma_2, \ldots, \gamma_r)$$

(since $\ell - v_j = \gamma_j$ for all $j \in [r]$).

The definition of $\langle u(v)\rangle$ yields

$$\begin{aligned}
\langle u(v)\rangle &= \sum_{\substack{\mathbf{q} \text{ is an MLQ of type } \mathbf{m};\\ u(v)=\mathbf{q}(1^n)}} \operatorname{wt}(\mathbf{q}) \\
&= \sum_{\substack{\mathbf{q}=(q_1,q_2,\ldots,q_{\ell-1}) \text{ is an interlacing}\\ \text{MLQ of type } \mathbf{m} \text{ with } q_{\ell-1}=B}} \operatorname{wt}(\mathbf{q}) \qquad (\text{by } (1) \iff (4)) \\
&= \left(\prod_{i=1}^{r} x_{b_i}\right) \det\left(h_{\gamma_j-j+i-1}\left(x_1,x_2,\ldots,x_{b_j}\right)\right)_{i,j\in[r]},
\end{aligned}$$

where the last equality is from Corollary 5.9 (applied to r, γ_i, B and b_i instead of k, λ_i, S and s_i). Hence the claim follows since $\prod_{i=1}^{r} x_{b_i} = \prod_{b\in B} x_b$. □

6. Proof of Theorem 3.1

In this section, we shall prove Theorem 3.1 by relying on the braid relation for dual configurations. We need some definitions first.

An *(r_1,r_2)-configuration* shall mean a pair $C=(q_1,q_2)$, where q_1 is an r_1-queue and q_2 is an r_2-queue. As usual, we consider C as a function on words by $C(u) := q_2(q_1(u))$, and we define the weight of C by $\operatorname{wt}(C) := \operatorname{wt}(q_1)\operatorname{wt}(q_2)$. We construct the *dual*[2] (r_2,r_1)-configuration to C, which we denote by C', as follows.

We begin by constructing a sequence of parentheses as follows: For each $j = 1,2,\ldots,n$ (in that order), we write

- an opening parenthesis '(' if $j\in q_1$ and $j\notin q_2$;
- a closing parenthesis ')' if $j\notin q_1$ and $j\in q_2$;
- a matched pair of parentheses '()' if $j\in q_1$ and $j\in q_2$;
- nothing otherwise.

We say that these parentheses are *contributed* by the site j. Next, we match as many parentheses as possible following the *standard parenthesis-matching algorithm*: every time you find an opening parenthesis to the left of a closing one, with only frozen parentheses between them, you match these two parentheses and declare them frozen. Here, we understand our sequence of parentheses as being written on a circle – so the last opening parenthesis can be matched with the first closing parenthesis if all parentheses "between them" (*i.e.*, to the right of the former and to the left of the latter) are frozen.

At the end of this algorithm, there will be exactly $|r_1-r_2|$ parentheses left unmatched; these unmatched parentheses will be opening if $r_1>r_2$ and closing if $r_1<r_2$. The unmatched parentheses will not depend on the order in which we perform the matching. This can be easily seen by considering the parentheses as an infinite shifted-periodic Motzkin path (see Figure 5), where a "nothing" corresponds to a flat step. If $r_1>r_2$, then the unmatched positions correspond to down steps in the Motzkin path such that the path never returns to the same height. When $r_1<r_2$, the unmatched positions correspond to the up steps where the Motzkin path first obtains a certain height.

We define $\mathrm{SP}(q_1,q_2)$ to be the following data:

- the sequence of parentheses obtained from q_1 and q_2;

[2]This is a different duality than the contragredient duality of Lemma 2.5.

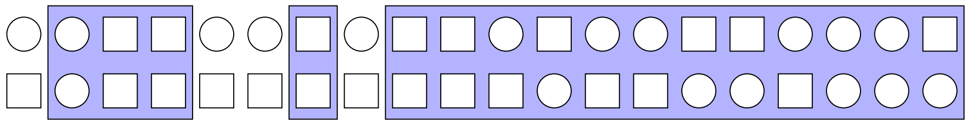


FIGURE 4. We draw a $\bigcirc$ in position i in row j corresponding to $i \in q_j$ and a $\square$ if $i \notin q_j$. The shaded boxes mark the balanced sites.

- the matching obtained by the algorithm; and
- the correspondence between sites and *closing* parentheses (*i.e.*, which site contributes which closing parenthesis).

Note that each $j \in \mathbb{Z}/n\mathbb{Z}$ that belongs to both q_1 and q_2 contributes a pair of parentheses '()', which can be immediately matched (to each other) at the beginning of the algorithm. Thus, if we disregard these pairs of parentheses, the outcome of the algorithm does not change (apart from these pairs disappearing). Hence, if we skip the sites $j \in q_1 \cap q_2$ in the definition of our sequence of parentheses, then the outcome of the algorithm (specifically, the set of unmatched parentheses) will be the same. The sequence of parentheses obtained as above, but skipping these sites j, will be called the *reduced parenthesis sequence*.

A site $j \in \mathbb{Z}/n\mathbb{Z}$ is *unbalanced* if it contributes an unmatched parenthesis; otherwise we say j is *balanced*. There are exactly $|r_1 - r_2|$ unbalanced sites.

We construct $C' = (q'_1, q'_2)$ as follows. For balanced j, we have $j \in q'_i$ if and only if $j \in q_i$ for $i = 1, 2$. For unbalanced j, we have $j \in q'_i$ if and only if $j \in q_{3-i}$ for $i = 1, 2$. Note that C and C' have the same balanced sites, since the parentheses that were matched for C remain matched in C' and the remaining parentheses are all of one kind. Hence, we have $C'' = C$. Also, we have $\mathrm{wt}(C) = \mathrm{wt}(C')$.

Example 6.1. Consider the configuration

$$C = (q_1, q_2) = (\{1, 2, 5, 6, 8, 11, 13, 14, 17, 18, 19\}, \{2, 12, 15, 16, 18, 19, 20\})$$

which is given pictorially in Figure 4. Converting C to a parenthesis sequence, we obtain

$$(\quad () \quad \cdot \quad \cdot \quad (\quad (\quad \cdot \quad (\quad \cdot \quad \cdot \quad (\quad) \quad (\quad (\quad) \quad) \quad (\quad () \quad () \quad),$$

which results in 4 unpaired ('s from positions $\{1, 5, 6, 8\}$. Therefore, we have $q'_1 = q_1 \setminus \{1, 5, 6, 8\}$ and $q'_2 = q_2 \cup \{1, 5, 6, 8\}$. Alternatively, the dual configuration C' is given by sliding all of the circles not in a shaded box from the upper level to the lower level.

Recall the notations introduced just before Lemma 2.5.

Remark 6.2. Let $C = (q_1, q_2)$ be an (r_1, r_2)-configuration. Thus, the dual configuration of the $(n - r_1, n - r_2)$-configuration $C^* = (q_1^*, q_2^*)$ is obtained from the dual configuration (q'_1, q'_2) of C by

$$(C^*)' = ((q'_1)^*, (q'_2)^*). \tag{6.1}$$

To see this, we compute the dual configurations of both C and C^* using the reduced parenthesis sequences. The sequence for C^* is obtained from that for C by reflecting both the parentheses (*i.e.*, opening become closing and vice versa)

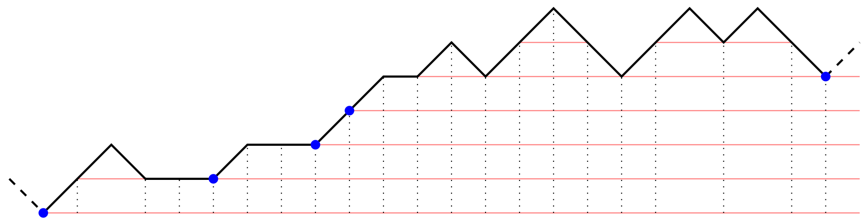


FIGURE 5. The Motzkin path corresponding to the configuration in Figure 4. The corresponding unbalanced up-steps are marked by dots.

and the sequence itself. Thus, the parenthesis matching algorithm proceeds in the same way.

In addition, applying Lemma 2.5 twice, we obtain

$$C(u) = q_2\big(q_1(u)\big) = q_2\big((q_1^*(u^*))^*\big) = \big(q_2^*(q_1^*(u^*))\big)^* = \big(C^*(u^*)\big)^*.$$

Fix $k \geq 1$. In the following, we simplify our terminology and say that an MLQ is a k-tuple of queues (without any restriction on their sizes). We want to define an action of $\mathfrak{S}_k$ on MLQs. For each $i \in [k-1]$, we define a map $\mathfrak{s}_i \colon \{\text{MLQs}\} \to \{\text{MLQs}\}$ by

$$\mathfrak{s}_i(q_1, q_2, \ldots, q_k) = (q_1, \ldots, q_{i-1}, q_i', q_{i+1}', q_{i+2}, \ldots, q_k),$$

where (q_i', q_{i+1}') is the dual configuration of (q_i, q_{i+1}). From the definition of a dual configuration, it is clear that $\mathfrak{s}_i\mathfrak{s}_i\mathbf{q} = \mathbf{q}$. It is also clear from the definition that $\mathfrak{s}_i\mathfrak{s}_j\mathbf{q} = \mathfrak{s}_j\mathfrak{s}_i\mathbf{q}$ if $|i-j| > 1$. Thus, the following proposition shows that $\mathfrak{s}_i$ defines an action of $\mathfrak{S}_k$ on the set of all MLQs.

Proposition 6.3. *We have*

$$\mathfrak{s}_i\mathfrak{s}_{i+1}\mathfrak{s}_i\mathbf{q} = \mathfrak{s}_{i+1}\mathfrak{s}_i\mathfrak{s}_{i+1}\mathbf{q}$$

for any MLQ $\mathbf{q} = (q_1, \ldots, q_k)$ *and any* $i \in \{1, 2, \ldots, k-2\}$.

Proof. We shall deduce the claim from [Lot02, Ch. 5, (5.6.3)].

Let A be the $(k+1)$-element set $\{\circ, 1, 2, \ldots, k\}$. Let A^* denote the set of all words on the alphabet A (of any finite length).

We construct a $k \times n$-matrix $M_{\mathbf{q}} \in A^{k\times n}$ from $\mathbf{q}$ by setting the (i,j)-th entry to i if $j \in q_i$ and $\circ$ otherwise. We then construct a word $\mathrm{word}(\mathbf{q}) \in A^*$ by reading $M_{\mathbf{q}}$ from top-to-bottom, left-to-right (*i.e.*, column by column). For example, if $n = 5$, $k = 3$, then

$$\mathbf{q} = (\{1,3\}, \{2\}, \{2,5\}) \longleftrightarrow M_{\mathbf{q}} = \begin{matrix} 1 & \circ & 1 & \circ & \circ \\ \circ & 2 & \circ & \circ & \circ \\ \circ & 3 & \circ & \circ & 3 \end{matrix}$$
$$\longrightarrow \mathrm{word}(\mathbf{q}) = 1\circ\circ\circ 231\circ\circ\circ\circ\circ\circ\circ 3.$$

Clearly, an MLQ $\mathbf{q}$ is uniquely determined by $\mathrm{word}(\mathbf{q})$ since n is fixed. In other words, the map $\mathrm{word} \colon \{\text{MLQs}\} \to A^*$ is injective.

Following [Lot02, §5.5], for each $i \in \{1, 2, \ldots, k-1\}$, we give an operator $\sigma_i \colon A^* \to A^*$. This operator σ_i acts on a word $p \in A^*$ as follows:

(1) Treat all letters i in p as opening parentheses '(', all letters $i+1$ as closing parentheses ')', and consider all other letters to be frozen. Now, match as many parentheses as possible according to the standard parenthesis-matching algorithm, but treating the word as a word in the usual sense (*i.e.*, not written on a circle, but having a beginning and an end). The result is independent of the choices in the algorithm, and is always a word whose non-frozen part (*i.e.*, the word obtained by removing all frozen letters) is

$$\underbrace{))\cdots)}_{a\text{ parentheses}}\ \underbrace{((\cdots(}_{b\text{ parentheses}} \tag{6.2}$$

for some integers $a, b \geq 0$.

(2) Now, replace the non-frozen part (6.2) by

$$\underbrace{))\cdots)}_{b\text{ parentheses}}\ \underbrace{((\cdots(}_{a\text{ parentheses}}$$

while keeping all frozen letters in their places. The resulting word is $\sigma_i p$.

From [Lot02, Eq. (5.6.3)], these operators σ_i satisfy

$$\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1} \tag{6.3}$$

for all $i \in \{1, 2, \ldots, k-2\}$. (See Remark 6.6 for another reference for this identity.)

Let $\zeta \colon \mathcal{W}_n \to \mathcal{W}_n$ be the cyclic shift map that sends each word $w_1 w_2 \cdots w_n$ to $w_2 w_3 \cdots w_n w_1$. We also abuse the notation ζ for the map that sends each queue q to the queue $\zeta q = \{i - 1 \mid i \in q\}$ (recall that $0 = n$ as sites). This map ζ shall act on MLQs entrywise (since an MLQ is a tuple of queues). Clearly,

$$\operatorname{word}(\zeta \mathbf{q}) = \zeta^k \operatorname{word}(\mathbf{q}) \tag{6.4}$$

for any MLQ $\mathbf{q} = (q_1, \ldots, q_k)$.

Now, we claim that

$$\operatorname{word}(\mathfrak{s}_i \mathbf{q}) = \sigma_i\big(\operatorname{word}(\mathbf{q})\big) \qquad \text{for each MLQ } \mathbf{q} \text{ and each } i. \tag{6.5}$$

Note that it is sufficient to show that $\operatorname{word}(\mathfrak{s}_1 \mathbf{q}) = \sigma_1\big(\operatorname{word}(\mathbf{q})\big)$ for $\mathbf{q} = (q_1, q_2)$ (because the definition of σ_i only relies on the letters i and $i+1$, while all other letters stay in their places and have no effect).

Thus, let $\mathbf{q} = (q_1, q_2)$. We want to show $\operatorname{word}(\mathfrak{s}_1 \mathbf{q}) = \sigma_1\big(\operatorname{word}(\mathbf{q})\big)$. If $|q_1| = |q_2|$, then $\mathfrak{s}_1 \mathbf{q} = \mathbf{q}$ by the definition of $\mathfrak{s}_1$. Moreover, we have $\sigma_1\big(\operatorname{word}(\mathbf{q})\big) = \operatorname{word}(\mathbf{q})$ in this case, since the word $\operatorname{word}(\mathbf{q})$ has as many letters 1 as it has letters 2, but the map σ_1 leaves such words unchanged. Hence, the claim holds when $|q_1| = |q_2|$. Thus, we assume that $|q_1| \neq |q_2|$. Therefore, there exists at least one unbalanced site for the configuration $\mathbf{q} = (q_1, q_2)$.

The operator $\mathfrak{s}_1$ commutes with the cyclic shift map ζ on MLQs because the standard parenthesis-matching algorithm used in the definition of dual configurations is clearly invariant under cyclic shift. The operator σ_1 commutes with the cyclic shift map ζ on words in A^* by [Lot02, Prop. 5.6.1]. Hence, and because of (6.4), we can apply ζ to $\mathbf{q}$ any number of times without loss of generality. Thus, we assume that the site 1 is unbalanced for the configuration $\mathbf{q} = (q_1, q_2)$, since at least one unbalanced site j exists and we can cyclically shift until it is 1. Thus, in the standard parenthesis-matching algorithm used in the construction of the dual configuration $\mathfrak{s}_1 \mathbf{q}$, the site 1 induces a parenthesis that stays unmatched throughout the algorithm. Hence, no two parentheses that get matched to each other during this algorithm have this parenthesis lying between them. Therefore, it does not matter whether we regard the sequence of parentheses as written on a cycle or on a straight line (as the wrapping-around is not used in the matching process). Hence, the standard parenthesis-matching algorithm used in the construction of the dual configuration $\mathfrak{s}_1\mathbf{q}$ proceeds exactly the same way as the standard parenthesis-matching algorithm used when applying σ_1 to $\operatorname{word}(\mathbf{q})$ (ignoring the letters that are neither i nor $i+1$). That is, the parentheses that get matched in the former are the same as those that get matched in the latter. Now, recall that $\mathfrak{s}_1$ merely toggles the unbalanced sites between q_1 and q_2, whereas σ_1 switches the number of unmatched ')'s with the number of unmatched '('s (which, in the case of $\operatorname{word}(\mathbf{q})$, boils down to just turning each unmatched ')' into a '(' or vice versa, because one of these numbers is 0). Since the sites of the unmatched parentheses are precisely the unbalanced sites, this shows that the two maps agree – that is, we have $\operatorname{word}(\mathfrak{s}_1\mathbf{q}) = \sigma_1(\operatorname{word}(\mathbf{q}))$. This proves (6.5).

The equality (6.5) can be rewritten as the commutative diagram

$$\begin{tikzcd} \{\text{MLQs}\} \arrow[r, "\mathfrak{s}_i"] \arrow[d, "\text{word}"'] & \{\text{MLQs}\} \arrow[d, "\text{word}"] \\ A^* \arrow[r, "\sigma_i"'] & A^* \end{tikzcd}$$

for all $i \in \{1, 2, \ldots, k-1\}$. In view of the injectivity of the map $\operatorname{word}\colon \{\text{MLQs}\} \to A^*$, this diagram allows us to translate (6.3) into $\mathfrak{s}_i\mathfrak{s}_{i+1}\mathfrak{s}_i = \mathfrak{s}_{i+1}\mathfrak{s}_i\mathfrak{s}_{i+1}$. $\square$

Remark 6.4. Our letters $1, \ldots, k$ correspond to the letters $a_k, \ldots, a_1$ in [Lot02], since the definition of σ_i in [Lot02] involves a_i rather than $i+1$ as closing parenthesis and a_{i+1} rather than i as opening one. Also, [Lot02] does not include the letter $\circ$ in the alphabet, but this makes no difference to the proof, since all letters $\circ$ are always frozen.

Remark 6.5. The operator σ_i is essentially a combination of co-plactic operators. Moreover, it corresponds to the Weyl group action on a tensor product of crystals [BS17]. Note that the bracketing rule given above is precisely the usual signature rule (see, *e.g.*, [BS17, Sec. 2.4] for a description) for computing tensor products. This arises from considering the MLQ as a binary $m \times n$ matrix and the natural $(\mathfrak{sl}_m \oplus \mathfrak{sl}_n)$-action.

Remark 6.6. The identity (6.3) can also be derived from [vL06, Lemma 2.3]. Indeed, to each word $p = p_1p_2\cdots p_d \in A^*$, we assign a $(k+1) \times d$-matrix B_p (a "binary matrix" in the parlance of [vL06]), whose j-th column has an entry 1 in its $k+1-p_j$-th position (we are treating $\circ$ as 0 here) and entries 0 everywhere else. Clearly, p is uniquely determined by B_p. Now, van Leeuwen notices in [vL06, before Proposition 1.3.5] that the "upward" and "downward" moves between two consecutive rows of a binary matrix correspond to changing unmatched parentheses in a certain parenthesis sequence. When the binary matrix is B_p and the two consecutive rows are the rows $k-i$ and $k-i+1$, this latter sequence is exactly the sequence of parentheses constructed in our definition of σ_i. Thus, applying the operator σ_i to a word $p \in A^*$ is tantamount to applying van Leeuwen's operator $\sigma^{\updownarrow}_{k-i}$ to the binary matrix B_p. Hence, (6.3) follows from the relation $\sigma^{\updownarrow}_{k-i}\sigma^{\updownarrow}_{k-(i+1)}\sigma^{\updownarrow}_{k-i} = \sigma^{\updownarrow}_{k-(i+1)}\sigma^{\updownarrow}_{k-i}\sigma^{\updownarrow}_{k-(i+1)}$ between the latter operators on binary matrices, but the latter relation is part of [vL06, Lemma 2.3].

Next, we define two queues corresponding to a word $w \in \mathcal{W}_n$ of type $\mathbf{m}$. Namely, for $k \in \{p_i(\mathbf{m}) \mid i \geq 0\}$, let $[w]_k$ denote the set of the indices $i \in [n]$ corresponding to the k smallest letters w_i of w.

Our proof of Theorem 3.1 will rely on a connection between dual configurations and the action of queues on words. We begin with a string of lemmas.

Lemma 6.7. *Let $w, w' \in \mathcal{W}_n$ be two words of the same type $\mathbf{m}$. Assume that $[w]_k = [w']_k$ for each $k \in \{p_i(\mathbf{m}) \mid i \geq 0\}$. Then, $w = w'$.*

Proof. Fix some $i \geq 1$. The sites containing the letter i in w are the elements of $[w]_{p_i(\mathbf{m})} \setminus [w]_{p_{i-1}(\mathbf{m})}$ (since w has type $\mathbf{m}$). Likewise, the sites containing the letter i in w' are the elements of $[w']_{p_i(\mathbf{m})} \setminus [w']_{p_{i-1}(\mathbf{m})}$. Since our assumption ($[w]_k = [w']_k$) yields $[w]_{p_i(\mathbf{m})} \setminus [w]_{p_{i-1}(\mathbf{m})} = [w']_{p_i(\mathbf{m})} \setminus [w']_{p_{i-1}(\mathbf{m})}$, we conclude that these are the same sites. Since this holds for all letters i, we have $w = w'$. □

Lemma 6.8. *Let (q_1, q_2) be a configuration with $|q_1| \leq |q_2|$. Let $i \in q_1$. Let j be the first site weakly to the right of i that belongs to q_2. Then, $\mathrm{SP}(q_1 \setminus \{i\}, q_2 \setminus \{j\})$ is obtained from $\mathrm{SP}(q_1, q_2)$ by removing a pair of matched parentheses.[3] Moreover, the closing parenthesis of that pair is contributed by $j \in q_2$.*

Proof. The assumption $|q_1| \leq |q_2|$ implies that all closing parentheses in $\mathrm{SP}(q_1, q_2)$ are matched. In particular, the closing parenthesis γ contributed by $j \in q_2$ is matched. Let α be the opening parenthesis contributed by $i \in q_1$. All parentheses between α and γ are opening (by the minimality of j). We know that γ is matched to an opening parenthesis β. Hence, the parenthesis sequence $\mathrm{SP}(q_1, q_2)$ between α and γ (inclusive) is

$$\underset{\alpha}{(} \quad (\quad (\quad \cdots \quad \underset{\beta}{(} \quad \underset{\gamma,}{)}$$

where α and β may be the same parenthesis. Therefore $\mathrm{SP}(q_1 \setminus \{i\}, q_2 \setminus \{j\})$ is obtained from $\mathrm{SP}(q_1, q_2)$ by removing α and γ. Yet this is tantamount to removing β and γ from $\mathrm{SP}(q_1, q_2)$, since both times we are simply shortening the string of opening parentheses before γ by 1 (and since we do not store the positions of the opening parentheses). □

Lemma 6.9. *Let $u \in \mathcal{W}_n$ be a word of type $\mathbf{m}$. Let $k = p_\alpha(\mathbf{m})$ for some α. Let q be a queue such that $k \leq |q|$. Each site $j \in [q(u)]_k$ contributes a matched closing parenthesis to $\mathrm{SP}([u]_k, q)$.*

Proof. Choose a permutation $(i_1, i_2, \ldots, i_n)$ of $(1, 2, \ldots, n)$ such that $u_{i_1} \leq u_{i_2} \leq \cdots \leq u_{i_n}$. Use this permutation to construct $q(u)$ as per the definition. For each $p \in [n]$, let j_p denote the site that is set while processing $i = i_p$ when constructing $q(u)$. Thus, $(q(u))_{j_p} = u_{i_p}$ whenever $p \leq |q|$ (since these entries of $q(u)$ are set in Phase II), and $(q(u))_{j_p} = u_{i_p} + 1$ otherwise (Phase I). Therefore, $[q(u)]_k = \{j_1, j_2, \ldots, j_k\}$ (since $k \leq |q|$). Thus, we only need to prove that each of the sites $j_1, j_2, \ldots, j_k$ contributes a matched closing parenthesis to $\mathrm{SP}([u]_k, q)$. Denote $q_1^{(p)} := q_1 \setminus \{i_1, i_2, \ldots, i_p\}$ and $q_2^{(p)} := q_2 \setminus \{j_1, j_2, \ldots, j_p\}$.

[3] The matching of all other parentheses remains the same, and so does the correspondence between sites and closing parentheses (except for the one we removed).

Claim 6.10. *Let $p \in \{0, 1, \ldots, k\}$. Then, $\operatorname{SP}(q_1^{(p)}, q_2^{(p)})$ is obtained from $\operatorname{SP}(q_1, q_2)$ by removing p pairs of matched parentheses. The closing parentheses of these p pairs are contributed by $j_1, j_2, \ldots, j_p$.*

Proof. We shall prove Claim 6.10 by induction on p. The base case ($p = 0$) is obvious. For the induction step, we assume that Claim 6.10 holds for $p - 1$. The site j_p is chosen in Phase II (since $p \leq k \leq |q|$), and therefore j_p is the first site weakly to the right of i_p that belongs to $q_2^{(p-1)}$. Hence, Lemma 6.8 (applied to $q_1^{(p-1)}$, $q_2^{(p-1)}$, i_p and j_p instead of q_1, q_2, i and j) implies that $\operatorname{SP}(q_1^{(p)}, q_2^{(p)})$ is obtained from $\operatorname{SP}(q_1^{(p-1)}, q_2^{(p-1)})$ by removing a pair of matched parentheses. Moreover, the closing parenthesis of that pair is contributed by $j_p \in q_2^{(p-1)}$. Thus, Claim 6.10 follows from the induction hypothesis. This completes the induction step. ■

The above claim applied to $p = k$ shows that $\operatorname{SP}(q_1^{(k)}, q_2^{(k)})$ is obtained from $\operatorname{SP}(q_1, q_2)$ by removing k pairs of matched parentheses, and that the closing parentheses of these k pairs are contributed by $j_1, j_2, \ldots, j_k$. Hence, each of the sites $j_1, j_2, \ldots, j_k$ contributes a matched closing parenthesis to $\operatorname{SP}([u]_k, q)$. □

Proposition 6.11. *Let $u \in \mathcal{W}_n$ be a word of type $\mathbf{m}$. Let $k = p_\alpha(\mathbf{m})$ for some α. Let q be a queue. The dual configuration of $([u]_k, q)$ has the form $\left(q^\dagger, [q(u)]_k\right)$, where $q^\dagger$ is some queue.*

Proof. Let $\left(q_1', q_2'\right)$ be the dual configuration of $([u]_k, q)$. We must then prove that $q_2' = [q(u)]_k$. Note that $|q_2'| = |[u]_k| = k = |[q(u)]_k|$.

Suppose $k \leq |q|$. Then, Lemma 6.9 shows that each site $j \in [q(u)]_k$ contributes a matched closing parenthesis to $\operatorname{SP}([u]_k, q)$. Therefore, all of these sites are balanced, and hence belong to q_2'. Thus, $q_2' \supseteq [q(u)]_k$, whence $q_2' = [q(u)]_k$ (since $|q_2'| = |[q(u)]_k|$).

Now suppose $k > |q|$. Let ℓ be the number of classes in u, and consider the contragredient duals q^* and u^* as in Lemma 2.5. Thus, $n - k < n - |q| = |q^*|$. Hence, applying the $k \leq |q|$ case (proven above) to $n - k$, u^* and q^* instead of k, u and q, we see that $([u^*]_{n-k}, q^*)' = \left(q^\dagger, [q^*(u^*)]_{n-k}\right)$ for some queue $q^\dagger$. Now,

$$\begin{aligned}((q_1')^*, (q_2')^*) &= (([u]_k)^*, q^*)' = ([u^*]_{n-k}, q^*)' \\ &= \left(q^\dagger, [q^*(u^*)]_{n-k}\right) = \left(q^\dagger, \left[(q(u))^*\right]_{n-k}\right),\end{aligned}$$

where

- the first equality follows from (6.1);
- the second equality is because $([u]_k)^* = [u^*]_{n-k}$; and
- the fourth equality follows from Lemma 2.5.

Therefore, $(q_2')^* = \left[(q(u))^*\right]_{n-k} = ([q(u)]_k)^*$, so that $q_2' = [q(u)]_k$. Hence, Proposition 6.11 is proven in the case when $k > |q|$. □

Proof of Theorem 3.1. Recall that any permutation in $\mathfrak{S}_{\ell-1}$ is a product of simple transpositions $s_1, s_2, \ldots, s_{\ell-2}$. Hence, in order to prove Theorem 3.1, it suffices to show that $\langle u \rangle_\sigma = \langle u \rangle_{\sigma s_i}$ for each $\sigma \in \mathfrak{S}_{\ell-1}$ and $i \in [\ell - 2]$. Then, Theorem 3.1 follows by induction on length, *i.e.* the minimal number of simple transpositions needed to write σ.

In order to prove $\langle u \rangle_\sigma = \langle u \rangle_{\sigma s_i}$, we need to show, for a σ-twisted MLQ $\mathbf{q}$ of type $\mathbf{m}$ satisfying $u = \mathbf{q}(1^n)$, that $\mathfrak{s}_i \mathbf{q}$ is a σs_i-twisted MLQ of type $\mathbf{m}$ satisfying $u = (\mathfrak{s}_i \mathbf{q})(1^n)$ (since this will show that $\mathfrak{s}_i$ bijects the former MLQs to the latter). The only nontrivial part is showing $u = (\mathfrak{s}_i \mathbf{q})(1^n)$. More generally, we will show that $(\mathfrak{s}_i \mathbf{q})(w) = \mathbf{q}(w)$ for any word $w \in \mathcal{W}_n$. The proof of this claim reduces to showing that for any configuration $C = (q_1, q_2)$ and any word $w \in \mathcal{W}_n$ the dual configuration $\mathfrak{s}_1 C = C' = (q_1', q_2')$ of C satisfies $C'(w) = C(w)$.

Each word w can be obtained from a standard word by a sequence of merges (each of which sends a word u to $\vee^{(k)} u$ for some $k \in \{p_j(\mathbf{m}) \mid j \geq 1\}$, where $\mathbf{m}$ is the type of u). Lemma 3.2 shows that these merges commute with the action of a queue (and thus of an MLQ). Hence, it is sufficient to consider standard words w. Thus, assume that w is standard of type $\mathbf{m}$. It is straightforward to see (using Equation (2.2) and $|q_2'| = |q_1|$ and $|q_1'| = |q_2|$) that the words $C(w) = q_2\big(q_1(w)\big)$ and $C'(w) = q_2'\big(q_1'(w)\big)$ have the same type. Let $\mathbf{n}$ be this type. We shall now show that $[C'(w)]_k = [C(w)]_k$ for all $k \in \{p_i(\mathbf{n}) \mid i \geq 0\}$. According to Lemma 6.7, this will yield $C'(w) = C(w)$, and thus our proof will be complete.

Let $k \in \{p_i(\mathbf{n}) \mid i \geq 0\}$. Thus, $k \in \{0, 1, \ldots, n\} = \{p_i(\mathbf{m}) \mid i \geq 0\}$ (since w is standard). Hence, $[w]_k$ is well-defined. Note that $q_i(w)$ and $q_i'(w)$ are also standard words. Using Proposition 6.11 to compute dual configurations, we can see how the MLQ $\mathbf{q} = ([w]_k, q_1, q_2)$ transforms under the action of $\mathfrak{s}_1\mathfrak{s}_2\mathfrak{s}_1$: Namely, we have

$$\begin{aligned} ([w]_k, q_1, q_2) &\xmapsto{\mathfrak{s}_1} (*, [q_1(w)]_k, q_2) \\ &\xmapsto{\mathfrak{s}_2} \Big(*, *, \big[q_2\big(q_1(w)\big)\big]_k\Big) \\ &\xmapsto{\mathfrak{s}_1} \Big(*, *, \big[q_2\big(q_1(w)\big)\big]_k\Big), \end{aligned}$$

where $*$ denotes some queue. Likewise, the action of $\mathfrak{s}_2\mathfrak{s}_1\mathfrak{s}_2$ is given by

$$\begin{aligned} ([w]_k, q_1, q_2) &\xmapsto{\mathfrak{s}_2} ([w]_k, q_1', q_2') \\ &\xmapsto{\mathfrak{s}_1} (*, [q_1'(w)]_k, q_2') \\ &\xmapsto{\mathfrak{s}_2} \Big(*, *, \big[q_2'\big(q_1'(w)\big)\big]_k\Big). \end{aligned}$$

(See Figure 6 for the actions depicted using crossing diagrams.) Yet, the two maps are equal by Proposition 6.3. Thus, the resulting MLQs must be identical:

$$\Big(*, *, \big[q_2'\big(q_1'(w)\big)\big]_k\Big) = \Big(*, *, \big[q_2\big(q_1(w)\big)\big]_k\Big).$$

Hence, we have

$$\big[q_2'\big(q_1'(w)\big)\big]_k = \big[q_2\big(q_1(w)\big)\big]_k.$$

In other words, $[C'(w)]_k = [C(w)]_k$. $\square$

Remark 6.12. Theorem 3.1 for the special case of $x_1 = \cdots = x_n = 1$ was proven in [AAMP11] using different techniques. We also sketch an alternative direct approach in the FPSAC extended abstract version of this work [AGS18].

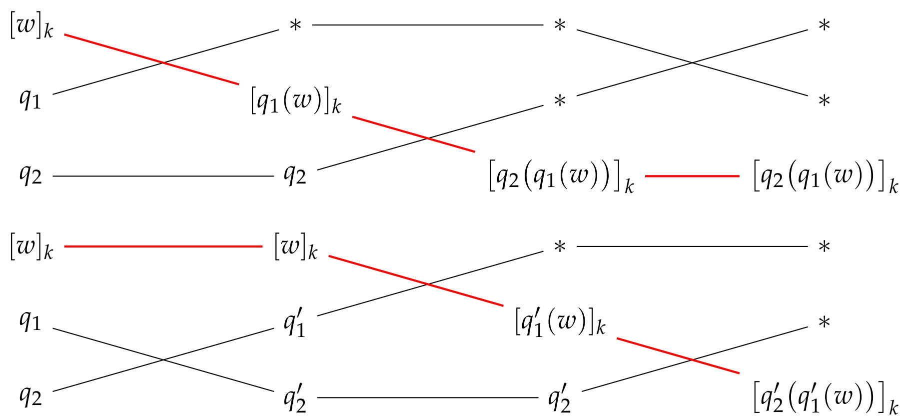


FIGURE 6. Crossing diagrams representing the action of $\mathfrak{s}_1\mathfrak{s}_2\mathfrak{s}_1$ (top) and $\mathfrak{s}_2\mathfrak{s}_1\mathfrak{s}_2$ (bottom).

7. FINAL REMARKS

We conclude by giving some additional examples, remarks, and comments about our results. We begin with an example to illustrate the proof of Theorem 3.5 in more detail.

Example 7.1. In order to compute $\langle 135452 \rangle$, we need to examine MLQs of type $(1,1,1,1,2,0,0,\ldots)$. We take a particular MLQ $\mathbf{q}$ and add the 5-queue $\{1,2,3,5,6\}$ as follows:

$$\mathbf{q} = \begin{array}{cccccc} & & & & & \\ 2 & 2 & 2 & 1 & 2 & 2 \\ 3 & 3 & 2 & 3 & 3 & 1 \\ 1 & 3 & 4 & 4 & 2 & 4 \\ 1 & 3 & 5 & 4 & 5 & 2 \end{array} \longrightarrow \begin{array}{cccccc} 1 & 1 & 1 & 2 & 1 & 1 \\ 2 & 2 & 3 & 1 & 2 & 2 \\ 3 & 4 & 2 & 3 & 3 & 1 \\ 1 & 3 & 4 & 4 & 2 & 5 \\ 1 & 3 & 5 & 4 & 6 & 2 \end{array} = \widetilde{\mathbf{q}}.$$

Thus, we obtain a $(s_4s_3s_2s_1)$-twisted MLQ $\widetilde{\mathbf{q}}$ of type $(1,1,1,1,1,1,0,\ldots)$. Furthermore, note that $135452 = \vee_5 135462$. Now, by Theorem 3.1, such MLQs $\widetilde{\mathbf{q}}$ are in bijection with ordinary MLQs contributing to, in this case, $\langle 135462 \rangle$. In more detail, let $R_i(\widetilde{\mathbf{q}})$ be the MLQ formed by taking the configuration $C = (\widetilde{q}_i, \widetilde{q}_{i+1})$ and replacing it with the dual configuration. By taking $R_4R_3R_2R_1(\widetilde{\mathbf{q}})$ to bring the top row to the bottom, we obtain the ordinary MLQ as follows:

$$\widetilde{\mathbf{q}} \xrightarrow{R_1} \begin{array}{cccccc} 2 & 2 & 1 & 2 & 2 & 2 \\ 2 & 2 & 3 & 1 & 2 & 2 \\ 3 & 4 & 2 & 3 & 3 & 1 \\ 1 & 3 & 4 & 4 & 2 & 5 \\ 1 & 3 & 5 & 4 & 6 & 2 \end{array} \xrightarrow{R_2} \begin{array}{cccccc} 2 & 2 & 1 & 2 & 2 & 2 \\ 3 & 2 & 3 & 3 & 3 & 1 \\ 3 & 4 & 2 & 3 & 3 & 1 \\ 1 & 3 & 4 & 4 & 2 & 5 \\ 1 & 3 & 5 & 4 & 6 & 2 \end{array}$$

$$\xrightarrow{R_3} \begin{array}{cccccc} 2 & 2 & 1 & 2 & 2 & 2 \\ 3 & 2 & 3 & 3 & 3 & 1 \\ 3 & 4 & 4 & 4 & 2 & 1 \\ 1 & 3 & 4 & 4 & 2 & 5 \\ 1 & 3 & 5 & 4 & 6 & 2 \end{array} \xrightarrow{R_4} \begin{array}{cccccc} 2 & 2 & 1 & 2 & 2 & 2 \\ 3 & 2 & 3 & 3 & 3 & 1 \\ 3 & 4 & 4 & 4 & 2 & 1 \\ 1 & 3 & 5 & 4 & 2 & 5 \\ 1 & 3 & 5 & 4 & 6 & 2 \end{array} = \widetilde{\mathbf{q}}',$$

which contributes to $\langle 135462 \rangle$.

There is the natural cyclic symmetry on our spectral weights.

Proposition 7.2. *Let $C_n \subseteq \mathfrak{S}_n$ denote the cyclic group of order n generated by the n-cycle $(1\ 2\ \cdots\ n)$ and $u \in \mathcal{W}_n$. We have $\langle u \rangle_\tau \sigma = \langle u\sigma \rangle_\tau$ for any $\sigma \in C_n$ and $\tau \in \mathfrak{S}_{\ell-1}$, where $\mathfrak{S}_n$ acts on $\mathcal{W}_n$ from the right by $(u_1 \cdots u_n)\sigma = u_{\sigma(1)} \cdots u_{\sigma(n)}$, that is to say permutations act on positions, and similarly on monomials in $\mathbf{x}$.*

Proof. For a queue q, define $\sigma q := \{\sigma(i) \mid i \in q\}$. It is clear from the definition of a queue that $q(u\sigma) = \big((\sigma q)(u)\big)\sigma$. Hence, for any τ-twisted MLQ $\mathbf{q}$ of type $\mathbf{m}$, we have $\mathbf{q}(u\sigma) = \big((\sigma\mathbf{q})(u)\big)\sigma$, where the action of C_n on MLQs is obtained by acting on each queue separately. Thus, in particular, for any τ-twisted MLQ $\mathbf{q}$ of type $\mathbf{m}$, we have $\mathbf{q}(1^n) = \big((\sigma\mathbf{q})(1^n)\big)\sigma$. From here, the claim follows by an obvious bijection (given by the action of σ) between the sums defining $\langle u \rangle_\tau \sigma$ and $\langle u\sigma \rangle_\tau$. $\square$

Let $B^{r,s}$, where $r \in [n-1]$ and $s \in \mathbb{Z}_{>0}$, be a *Kirillov–Reshetikhin (KR) crystal* in type $A_{n-1}^{(1)}$ [KKM+92]. Recall from [NY97, Shi02] that the combinatorial R-matrix is the unique crystal isomorphism

$$R \colon B^{r_1,s_1} \otimes B^{r_2,s_2} \to B^{r_2,s_2} \otimes B^{r_1,s_1}.$$

There is a well-known (in slightly different terminology) bijection Ξ_r relating r-queues with an element of the KR crystal $B^{r,1}$ in type $A_{n-1}^{(1)}$ by considering $q = \{b_1 < \cdots < b_r\}$ as a single-column Young tableau of height r. In [KMO15], this was extended to a bijection Ξ between multiline queues of type $\mathbf{m}$ with $\ell + 1$ classes and $B^{p_1,1} \otimes B^{p_2,1} \otimes \cdots \otimes B^{p_\ell,1}$ by

$$\Xi(\mathbf{q}) = \Xi_{p_1}(q_1) \otimes \Xi_{p_2}(q_2) \otimes \cdots \otimes \Xi_{p_\ell}(q_\ell).$$

Thus, by comparing the description of the combinatorial R-matrix for $B^{r_1,1}$ and $B^{r_2,1}$ from [NY97], we obtain that taking the dual configuration is equivalent to applying the combinatorial R-matrix under Ξ.

Proposition 7.3. *Let C be an (r_1, r_2)-configuration. Then*

$$\Xi(C') = R\big(\Xi(C)\big),$$

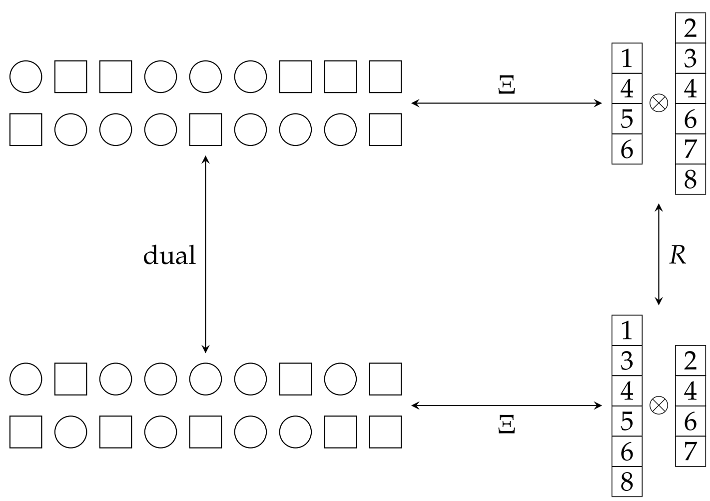
where C' is the dual configuration of C.

Note that Proposition 7.3 gives another proof of Proposition 6.3 since the combinatorial R-matrix is well-known to satisfy the Yang-Baxter equation.

Example 7.4. Suppose $n = 9$. Consider the $(4,6)$-configuration and dual $(6,4)$-configuration

$$C = (\{1,4,5,6\}, \{2,3,4,6,7,8\}), \qquad C' = (\{1,3,4,5,6,8\}, \{2,4,6,7\}).$$

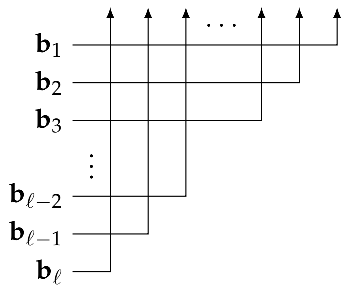
We have



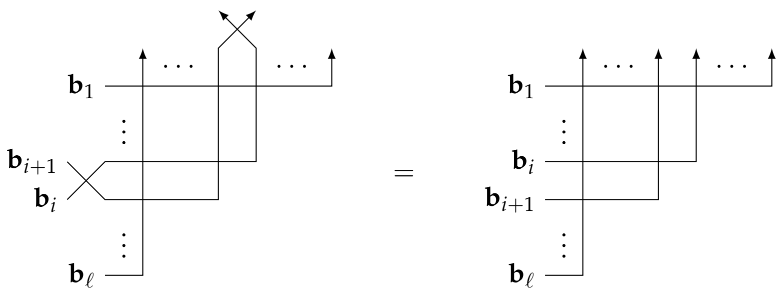
Since the combinatorial R-matrix satisfies the Yang–Baxter equation, we have an action of the symmetric group $\mathfrak{S}_\ell$ acting on MLQs with $\ell + 1$ queues. Given the description of the combinatorial R-matrix using the Robinson–Schensted–Knuth (RSK) bijection [Shi02], this $\mathfrak{S}_\ell$-action can be considered as corresponding to the one given by van Leeuwen [vL06, Lemma 2.3].[4] Furthermore, the $\mathfrak{S}_\ell$-action on MLQs has been considered by Danilov and Koshevoy [DK05] in a different context (see also [Gor17, Ch. 4]). Unlike the combinatorial R-matrix perspective, they do not have the natural interpretation of the weight since $\mathrm{wt}(\mathbf{q}) = \mathrm{wt}\big(\Xi(\mathbf{q})\big)$, where the crystal weight is the usual tableaux weight.

[4]This could also be considered as an interpretation of the Littlewood–Richardson rule, along with the fact that $s_\lambda s_\mu$, where s_λ and s_μ are Schur functions corresponding to rectangles, is multiplicity free [Ste01], and using Howe duality [BS17, Ch. 9,App. B].

Next, we describe how to interpret our action from looking at the corner transfer matrix described in [KMO15], which can be given diagrammatically by



where every crossing is a combinatorial R-matrix and $\mathbf{b}_i \in B^{p_i,1}$. Our action effectively does the following on the corner transfer matrix:



where the equality comes from applying the Yang–Baxter equation and $R^2 = \mathrm{id}$.

It could be interesting to see if our proof has any implications for the work of Danilov and Koshevoy [DK05] or van Leeuwen [vL06].

Appendix A. Connections with other constructions

We describe how our definition of $q(u)$ (when q is a queue and u a word) is connected to the construction of [FM07], which was coined bully paths in [AL18], and the tableaux combinatorics of [AS18].

To clarify the connection, we recall our visualization of a queue q in (2.3) by a row of circles and squares. We modify this visualization slightly: Namely, we still represent each $i \in q$ by a circle as in Example 2.4, but we no longer represent the $i \notin q$ by squares. We refer to the circles as "boxes".

A.1. **Connection with Ferrari–Martin and Aas–Linusson MLQs.**

We first connect our MLQs with those from [FM07]. We will use the language of [AL18], where we are only considering "discrete MLQs" as we consider our ring to have a finite number of sites.

Consider a MLQ $\mathbf{q} = (q_1, q_2, \ldots, q_\ell)$. A *labeling* of $\mathbf{q}$ is a sequence of maps $\mathbf{f} = (f_1, \ldots, f_\ell)$, where $f_i \colon q_i \to [i]$. We represent this by placing an $f_i(j)$, which we call the *label*, inside of the circle corresponding to $j \in q_i$. The *canonical labeling* $\mathbf{f}_\mathbf{q}$ of $\mathbf{q}$ is the labeling $(f_1, \ldots, f_\ell)$ defined by

$$f_k(j) = \big(q_k(\cdots q_1(1^n)\cdots)\big)_j. \tag{A.1}$$

For example, the labeling of each MLQ in Example 7.1 is precisely the circled values. When $\mathbf{q}$ is an ordinary MLQ, we can also construct this canonical labeling recursively as follows:

(1) Set $q_0 = \varnothing$, and let $f_0 : q_0 \to [0]$ be the trivial map.
(2) For each $k = 0, 1, \ldots, \ell - 1$:
   - Suppose $f_k \colon q_k \to [k]$ is already defined. Let $(j_1, j_2, \ldots, j_r)$ be a list of all elements of q_k in the order of increasing label in f_k; that is, $f_k(j_1) \leq f_k(j_2) \leq \cdots \leq f_k(j_r)$. (The relative order between elements with equal label is immaterial.)
   - For $i = i_1, i_2, \ldots, i_r$, do the following: Find the first site j weakly to the right (cyclically) of i such that $j \in q_{k+1}$ and $f_{k+1}(j)$ is not set; then set $f_{k+1}(j) = f_k(i)$.
   - For all sites $j \in q_{k+1}$ for which $f_{k+1}(j)$ is not set yet, set $f_{k+1}(j) = k + 1$.

   This defines $f_{k+1} \colon q_{k+1} \to [k+1]$.

In order to see that this recursive construction yields the same labeling $\mathbf{f_q}$ as the equality (A.1), we can proceed as follows:

If r is a queue and I is a set, then a *labeling* of r by I means a map $f \colon r \to I$. Of course, we refer to a value $f(i)$ of such a labeling as the *label* of i in f. To represent such a labeling f, we label each of the boxes of r with its corresponding value (i.e., we label the box in position i by $f(i)$). Thus, a labeling of an MLQ $\mathbf{q} = (q_1, q_2, \ldots, q_\ell)$ is a sequence of labelings $f_i \colon q_i \to [i]$ of each of its queues q_i; and its visual representation is obtained by stacking the visual representations of the f_i one atop another.

Now, fix a $k \in \mathbb{N}$ and two queues r and q with $|r| \leq |q|$. Also fix a labeling $f \colon r \to [k]$ of r by $[k]$. Then, we can define a labeling $q(f)$ of q by $[k+1]$ as follows:

(1) Let $\left(i_1, i_2, \ldots, i_{|r|}\right)$ be a list of all elements of r in the order of increasing label in f (that is, $f(i_1) \leq f(i_2) \leq \cdots \leq f(i_{|r|})$. (The relative order between elements with equal label is immaterial.)
(2) Initialize a labeling g of q by $[k+1]$. For now, none of its values is set.
(3) For $i = i_1, i_2, \ldots, i_{|r|}$, do the following. Find the first site j weakly to the right (cyclically) of i such that $j \in q$ and $g(j)$ is not set. Then set $g(j) = f(i)$.
(4) For all sites $j \in q$ for which $g(j)$ is not set yet, set $g(j) = k + 1$.
(5) Define $q(f)$ to be the resulting labeling g.

The connection between this labeling and the action of queues on words is simple. Indeed, let $k \in \mathbb{N}$, let r be a queue, and let $f \colon r \to [k]$ be a labeling. Then we can can assign to f a word $\omega_k f \in \mathcal{W}_n$ as follows: For each $j \in r$, we set $(\omega_k f)_j = f(j)$. All other letters of $\omega_k f$ shall be $k+1$. Now, it is easy to see that if q is a queue satisfying $|r| \leq |q|$, then

$$q(\omega_k f) = \omega_{k+1}(q(f)). \tag{A.2}$$

Thus, the action of a queue on a labeling can be computed in terms of its action on a word. (The converse is not true, since the condition $|r| \leq |q|$ is restrictive. Thus, our action of queues on words, with its two phases, is more general than their action on labelings.)

Now, let $\mathbf{q} = (q_1, q_2, \ldots, q_\ell)$ be an ordinary MLQ (so that $|q_1| \leq |q_2| \leq \cdots \leq |q_\ell|$), and consider the labeling $\mathbf{f_q} = (f_1, \ldots, f_\ell)$ constructed by the above recursive procedure (not by (A.1)). Then, the recursive procedure clearly yields

$$f_i = q_i (f_{i-1}) \qquad \text{for all } i \in [\ell] ,$$

where f_0 is understood to be the (trivial) labeling of the empty queue $\varnothing$. In view of (A.2), this entails

$$\omega_i (f_i) = q_i (\omega_{i-1} (f_{i-1})) \qquad \text{for all } i \in [\ell] .$$

Thus, by induction,

$$\omega_i (f_i) = q_i(\cdots q_1(1^n) \cdots) \qquad \text{for all } i \in \{0, 1, \ldots, \ell\}$$

(since $\omega_0 (f_0) = 1^n$). Hence,

$$f_i(j) = \left(q_i(\cdots q_1(1^n) \cdots)\right)_j \qquad \text{for all } i \in \{0, 1, \ldots, \ell\} \text{ and } j \in q_i.$$

But this is precisely the equality (A.1). Hence, we have shown that the recursive construction yields the same labeling $\mathbf{f_q}$ as the equality (A.1).

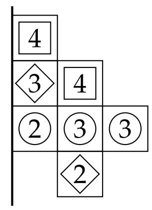
Note that the canonical labeling $\mathbf{f_q}$ are the elements in the circles in the graveyard diagram of $\mathbf{q}$ (as in Example 5.5).

Now consider the labeling procedure in [AL18, §2.2] given by *k-bully paths*. Note that a k-bully path from one queue to the next precisely corresponds to the path of a letter k under Phase II of our definition of a queue as a function on words. For example, the bully paths would correspond to the blue paths in Example 2.4. In addition, this is exactly the recursive labeling procedure given above. Thus, the labeling $\mathbf{f_q}$ is equivalently constructed following the labeling procedure of [AL18] using bully paths.

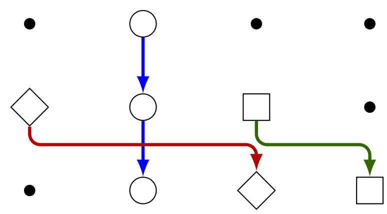
A.2. **Connection with Kohnert diagrams and Assaf–Searles theory.** Next we relate the action of queues on words with the *Kohnert labelings* in [AS18, Def. 2.5] and the *thread decomposition* in [AS18, Def. 3.5]. We remark that the thread decomposition is the same as the Kohnert labeling when the shape is an antipartition (*i.e.* a weakly increasing sequence of positive integers). Roughly speaking, Kohnert diagrams are MLQs built of queues that live on a half-line (instead of a circle), and the construction of the Kohnert labeling (and the thread decomposition) is a standardization of the bully path construction.

In more detail (and using the notations of [AS18]): If α is a weak composition, and if $D \in \mathrm{KM}(\alpha)$ is a Kohnert diagram, then we view the columns of D as queues. This time, our queues are subsets of $\mathbb{N}$ (or $\mathbb{Z}$) instead of $\mathbb{Z}/n\mathbb{Z}$; thus, there is no "wrapping around". We consider the reflection of D across the line $x = y$ as an MLQ $\mathbf{q}_D = (q_1, \ldots, q_\ell)$: namely, a cell in row i and column j in the reflected diagram corresponds to a $j \in q_i$, and ℓ is the number of columns in D. We then apply the bully path construction to the boxes of this reflected Kohnert diagram. To obtain the thread decomposition we need to distinguish paths with a fixed label such that these paths are also constructed by the bully path algorithm, where we consider the labels to be decreasing from left to right. Hence, this can be considered as a standardization of our construction or, equivalently, as fixing specific permutations for how the queues act on words.

Example A.1. Consider the thread decomposition of the Kohnert diagram



given in [AS18, Fig. 11]. Thus, the corresponding MLQ is $(\{2\}, \{1,2,3\}, \{2,3,4\})$; we can draw it using bully paths as follows:



where each bully path matches with a thread in the decomposition. Note that for each fixed k, the k-bully paths must be constructed from left to right.[5] If not, the diamond and square in the bottom row would be interchanged.

Note that the distinction between $\mathbb{N}$ and $\mathbb{Z}/n\mathbb{Z}$ never arises since, for $\mathbf{q}_D$ and all i, we have

$$|\{j \in q_i \mid j \geq k\}| \leq |\{j \in q_{i+1} \mid j \geq k\}|,$$

(by [AS18, Lemma 2.2]), which means that there is no "wrapping" around the cylinder.

To obtain a Kohnert labeling from a Kohnert diagram of height K (for this, we require $n \gg 1$), we can construct an MLQ

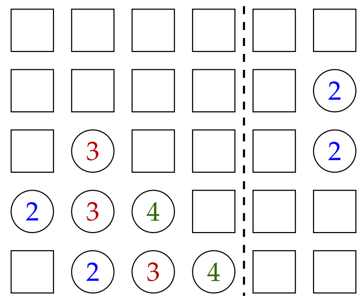
$$\widetilde{\mathbf{q}}_D = (\widetilde{q}_1, \widetilde{q}_2, \dots, \widetilde{q}_{\widetilde{\ell}}, q_1, q_2, \dots, q_\ell)$$

from the MLQ $\mathbf{q}_D = (q_1, q_2, \dots, q_\ell)$ to obtain the correct labelings, where $\ell + \ell'$ is the largest label appearing in the Kohnert labeling. Indeed, a label added in column i comes from a $k \in q$ with $k > K$ for sufficiently many queues q before q_i. In particular, when a smaller label appears, it must be in the bottom row of the Kohnert diagram, which would correspond to the bully path wrapping around the cylinder. We then only consider the labels in the regime $k \in q_i$ for all $1 \leq k \leq K$ and all $1 \leq i \leq \ell$. We leave the precise details for the interested reader.

Example A.2. We consider the Kohnert labeling from Example A.1. The following MLQ, given as a graveyard diagram, is an MLQ that gives the corresponding

[5]In terms of our permutation $(i_1, i_2, \dots, i_n)$ for Phase II, for $u_{i_j} = u_{i_{j+1}} = \cdots = u_{i_{j'}} = k$, we must have $i_j < i_{j+1} < \cdots < i_{j'}$.

Kohnert labeling:



where we have suppressed the $(i+1)$'s that appear in row i. Note that all circles that appear to the left of the dashed line correspond to the Kohnert labeling and that the labels match.

Appendix B. Appendix: Old proof of Theorem 3.1

This section contains a previous proof of Theorem 3.1. This proof uses similar ideas to the proof given in Section 6 above, but differs in its details (in particular, it defines the dual configuration in a different way, although the two definitions can be shown to be equivalent).

We need some definitions first.

An *(r_1, r_2)-configuration* shall mean a pair $C = (q_1, q_2)$, where q_1 is an r_1-queue and q_2 is an r_2-queue. As usual, we consider C as a function on words by $C(u) := q_2(q_1(u))$, and we define the weight of C by $\operatorname{wt}(C) := \operatorname{wt}(q_1)\operatorname{wt}(q_2)$. We construct the *dual*[6] (r_2, r_1)-configuration to C, which we denote by C', as follows.

We first consider the case $r_1 = r_2$, in which case we define $C' = C$. Thus, assume $r_1 \neq r_2$.

For any two sites i and j, let $\operatorname{int}[i, j]$ denote a closed (cyclic) interval from i to j. This is the set $\{i, i+1, \ldots, j\}$ when $i \leq j$ and the set $\{i, i+1, \ldots, n, 1, 2, \ldots, j\}$ when $i > j$.[7] Let $c^{\uparrow}[i, j]$ (resp. $c^{\downarrow}[i, j]$) denote the number of $\ell \in \operatorname{int}[i, j]$ such that $\ell \in q_1$ (resp. $\ell \in q_2$).

We say that a closed cyclic interval $\operatorname{int}[i, j]$ is *balanced* if $c^{\uparrow}[i, j] = c^{\downarrow}[i, j]$ and for each $k \in \operatorname{int}[i, j]$, we have $c^{\uparrow}[i, k] \geq c^{\downarrow}[i, k]$. Equivalently, $\operatorname{int}[i, j]$ is balanced if and only if $c^{\downarrow}[i, j] = c^{\uparrow}[i, j]$ and for each $k \in \operatorname{int}[i, j]$, we have $c^{\downarrow}[k, j] \geq c^{\uparrow}[k, j]$. A set S of sites is *balanced* if it is a disjoint union of balanced intervals. For $i \in [n]$, we say that i is *balanced* if i belongs to some balanced interval, and *unbalanced* otherwise.

The following facts are straightforward:

(1) For any balanced interval $\mathcal{I}$, we have $|q_1 \cap \mathcal{I}| = |q_2 \cap \mathcal{I}|$.
(2) The empty interval is balanced.
(3) The set $[n]$ of all sites is not balanced, since $r_1 \neq r_2$.
(4) If A and B are two balanced sets, then the set $A \cap B$ is balanced.[8]

[6] This is a different duality than the contragredient duality of Lemma 2.5.

[7] We trivially consider an empty interval to be balanced, and note that if $j = i - 1$, then $\operatorname{int}[i, j] = [n]$.

[8] To prove this, first check it when A and B are balanced intervals.

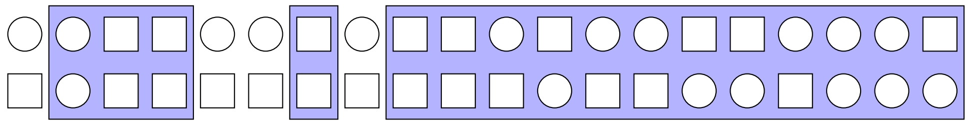


FIGURE 7. We draw a $\bigcirc$ in position i in row j corresponding to $i \in q_j$ and a $\square$ if $i \notin q_j$. The maximal balanced intervals are boxed.

(5) If A and B are balanced intervals, and if $A \cup B$ is an interval, then $A \cup B$ is balanced.[9]
(6) Thus, the union of all balanced intervals (*i.e.*, the set of all balanced $i \in [n]$) is also the disjoint union of all maximal balanced intervals (where "maximal" means "maximal under inclusion").
(7) For $r_1 < r_2$ and $j \in [n]$ unbalanced, we have $j \notin q_1$ and $j \in q_2$.
(8) For $r_1 > r_2$ and $j \in [n]$ unbalanced, we have $j \in q_1$ and $j \notin q_2$.
(9) There are exactly $|r_1 - r_2|$ unbalanced sites i.

We construct $C' = (q'_1, q'_2)$ by letting $q'_i \cap \mathcal{I} = q_i \cap \mathcal{I}$ for $i = 1, 2$ and each balanced interval $\mathcal{I}$ of C. For unbalanced j, we have $j \in q'_i$ if and only if $j \in q_{3-i}$ for $i = 1, 2$. Note that C and C' have the same balanced intervals. It is clear that $C'' = C$ and $\mathrm{wt}(C) = \mathrm{wt}(C')$.

(We could adapt this construction to the case $r_1 = r_2$, but we would need to be more precise about defining intervals, since the set of all sites can be written as $\mathrm{int}[i, i-1]$ for any value of i. If done correctly, this results in every $i \in [n]$ being balanced when $r_1 = r_2$, and thus the construction yields $C' = C$ as we defined.)

Example B.1. Consider the configuration C given in Figure 4. The dual configuration C' is given by sliding all of the circles not boxed from the upper level to the lower level. In particular, we have $q'_1 = q_1 \setminus \{1, 5, 6, 8\}$ and $q'_2 = q_2 \cup \{1, 5, 6, 8\}$.

Recall the notations introduced just before Lemma 2.5.

Remark B.2. Let $C = (q_1, q_2)$ be an (r_1, r_2)-configuration. Then, the interval $\mathcal{I}$ of C is balanced if and only if the interval $\mathcal{I}^{\mathrm{refl}} = \{n + 1 - i \mid i \in \mathcal{I}\}$ of the $(n - r_1, n - r_2)$-configuration $C^* = (q_1^*, q_2^*)$ is balanced. In other words, the balanced intervals of C are exactly the balanced intervals of C^* but reflected through the middle of $[n]$. Thus, the dual configuration of $C^* = \left(q_1^*, q_2^*\right)$ is obtained from the dual configuration $\left(q'_1, q'_2\right)$ of C by

$$(C^*)' = \left((q'_1)^*, (q'_2)^*\right). \tag{B.1}$$

In addition, applying Lemma 2.5 twice, we obtain

$$C(u) = q_2\big(q_1(u)\big) = q_2\big((q_1^*(u^*))^*\big) = \big(q_2^*\big(q_1^*(u^*)\big)\big)^* = \big(C^*(u^*)\big)^*.$$

In particular, for $u \in \{1, 2\}^n$, we have $C(u)_i = 5 - C^*(u^*)_{n+1-i}$, where we treat u and $C^*(u^*)$ as a word with 2 and 4 classes, respectively.

[9]This is obvious when one of A and B contains the other. Otherwise, argue that $A \cap B$, $A \setminus B$ and $B \setminus A$ are balanced.

Fix $k \geq 1$. In the following, we simplify our terminology and say that an MLQ is a k-tuple of queues (without any restriction on their sizes). We want to define an action of $\mathfrak{S}_k$ on MLQs. For each $i \in [k-1]$, we define a map $\mathfrak{s}_i \colon \{\text{MLQs}\} \to \{\text{MLQs}\}$ by

$$\mathfrak{s}_i(q_1, q_2, \ldots, q_k) = (q_1, \ldots, q_{i-1}, q'_i, q'_{i+1}, q_{i+2}, \ldots, q_k),$$

where (q'_i, q'_{i+1}) is the dual configuration of (q_i, q_{i+1}). From the definition of a dual configuration, it is clear that $\mathfrak{s}_i \mathfrak{s}_i \mathbf{q} = \mathbf{q}$. It is also clear from the definition that $\mathfrak{s}_i \mathfrak{s}_j \mathbf{q} = \mathfrak{s}_j \mathfrak{s}_i \mathbf{q}$ if $|i - j| > 1$. Thus, the following proposition shows that $\mathfrak{s}_i$ defines an action of $\mathfrak{S}_k$ on the set of all MLQs.

Proposition B.3. *We have*

$$\mathfrak{s}_i \mathfrak{s}_{i+1} \mathfrak{s}_i \mathbf{q} = \mathfrak{s}_{i+1} \mathfrak{s}_i \mathfrak{s}_{i+1} \mathbf{q}$$

for any MLQ $\mathbf{q} = (q_1, \ldots, q_k)$ *and any* $i \in \{1, 2, \ldots, k-2\}$.

Proof. We shall deduce the claim from [Lot02, Ch. 5, (5.6.3)].

Let A be the $(k+1)$-element set $\{\circ, 1, 2, \ldots, k\}$. Let A^* denote the set of all words on the alphabet A (of any finite length).

We construct a $k \times n$-matrix $M_{\mathbf{q}} \in A^{k \times n}$ from $\mathbf{q}$ by setting the (i,j)-th entry to i if $j \in q_i$ and $\circ$ otherwise. We then construct a word $\operatorname{word}(\mathbf{q}) \in A^*$ by reading $M_{\mathbf{q}}$ from top-to-bottom, left-to-right (*i.e.*, column by column). For example, if $n = 5$, $k = 3$, then

$$\mathbf{q} = (\{1,3\}, \{2\}, \{2,5\}) \longleftrightarrow M_{\mathbf{q}} = \begin{matrix} 1 & \circ & 1 & \circ & \circ \\ \circ & 2 & \circ & \circ & \circ \\ \circ & 3 & \circ & \circ & 3 \end{matrix}$$
$$\longrightarrow \operatorname{word}(\mathbf{q}) = 1 \circ \circ \circ 231 \circ \circ \circ \circ \circ \circ \circ 3.$$

Clearly, an MLQ $\mathbf{q}$ is uniquely determined by $\operatorname{word}(\mathbf{q})$ since n is fixed. In other words, the map $\operatorname{word} \colon \{\text{MLQs}\} \to A^*$ is injective.

Now, for each $i \in \{1, 2, \ldots, k-1\}$, we recall the operator $\sigma_i \colon A^* \to A^*$ from [Lot02, §5.5]. This operator σ_i acts on a word $p \in A^*$ as follows:

(1) Treat all letters i in p as opening parentheses '(', all letters $i+1$ as closing parentheses ')', and consider all other letters to be frozen. Now, match as many parentheses as possible according to the standard parenthesis-matching algorithm (*i.e.*, every time you find an opening parenthesis to the left of a closing one, with only frozen letters between them, you match these two parentheses and declare them frozen). Notice that this algorithm is non-deterministic, but the outcome is independent of the steps chosen; the result is always a word whose non-frozen part (*i.e.*, the word obtained by removing all frozen letters) is $\underbrace{))\cdots)}_{a \text{ parentheses}} \underbrace{((\cdots(}_{b \text{ parentheses}}$ for some integers $a, b \geq 0$. We call this the *reduced signature* of p.

(2) Now, replace this non-frozen part by $\underbrace{))\cdots)}_{b \text{ parentheses}} \underbrace{((\cdots(}_{a \text{ parentheses}}$ while keeping all frozen letters in their places. The resulting word is $\sigma_i p$.

From [Lot02, Eq. (5.6.3)], these operators σ_i satisfy

$$\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1} \tag{B.2}$$

for all $i \in \{1, 2, \ldots, k-2\}$.

Let $\zeta \colon \mathcal{W}_n \to \mathcal{W}_n$ be the cyclic shift map that sends each word $w_1 w_2 \cdots w_n$ to $w_2 w_3 \cdots w_n w_1$. We also abuse the notation ζ for the map that sends each queue q to the queue $\zeta q = \{i-1 \mid i \in q\}$ (recall that $0 = n$ as sites). This map ζ shall act on MLQs entrywise (since an MLQ is a tuple of queues). Clearly,

$$\operatorname{word}(\zeta \mathbf{q}) = \zeta^k \operatorname{word}(\mathbf{q}) \tag{B.3}$$

for any MLQ $\mathbf{q} = (q_1, \ldots, q_k)$.

Now, we claim that

$$\operatorname{word}(\mathfrak{s}_i \mathbf{q}) = \sigma_i\big(\operatorname{word}(\mathbf{q})\big) \qquad \text{for each MLQ } \mathbf{q} \text{ and each } i. \tag{B.4}$$

Note that it is sufficient to show that $\operatorname{word}(\mathfrak{s}_1 \mathbf{q}) = \sigma_1\big(\operatorname{word}(\mathbf{q})\big)$ for $\mathbf{q} = (q_1, q_2)$ (because the definition of σ_i only relies on the letters i and $i+1$, while all other letters stay in their places and have no effect).

Thus, let $\mathbf{q} = (q_1, q_2)$. We want to show $\operatorname{word}(\mathfrak{s}_1 \mathbf{q}) = \sigma_1\big(\operatorname{word}(\mathbf{q})\big)$. If $|q_1| = |q_2|$, then $\mathfrak{s}_1 \mathbf{q} = \mathbf{q}$ by the definition of $\mathfrak{s}_1$. Moreover, we have $\sigma_1\big(\operatorname{word}(\mathbf{q})\big) = \operatorname{word}(\mathbf{q})$ in this case, since the word $\operatorname{word}(\mathbf{q})$ has as many letters 1 as it has letters 2, but the map σ_1 leaves such words unchanged. Hence, the claim holds when $|q_1| = |q_2|$. Thus, we assume that $|q_1| \neq |q_2|$. Therefore, there exists at least one unbalanced site for the configuration $\mathbf{q} = (q_1, q_2)$.

The operator $\mathfrak{s}_1$ commutes with the cyclic shift map ζ on MLQs because ζ merely shifts the balanced intervals. The operator σ_1 commutes with the cyclic shift map ζ on words in A^* by [Lot02, Prop. 5.6.1]. Hence, and because of (B.3), we can apply ζ to $\mathbf{q}$ any number of times without loss of generality. Thus, we assume that the site 1 is unbalanced for the configuration $\mathbf{q} = (q_1, q_2)$, since at least one unbalanced site j exists and we can cyclically shift until it is 1. Therefore, no balanced interval has the form $\operatorname{int}[i, j]$ with $i > j$.

We construct a sequence of parentheses as follows: For each $j = 1, 2, \ldots, n$ (in that order), we write

- an opening parenthesis '(' if $j \in q_1$ and $j \notin q_2$,
- a closing parenthesis ')' if $j \notin q_1$ and $j \in q_2$,
- a matched pair of parentheses '()' if $j \in q_1$ and $j \in q_2$,
- nothing otherwise.

Note that this is exactly the sequence of parentheses constructed when applying σ_1 to $\operatorname{word}(\mathbf{q})$ (removing all $\circ$ letters). Furthermore, every '(' and ')' corresponds to a contribution to $c^{\uparrow}[1, n]$ and $c^{\downarrow}[1, n]$, respectively. Additionally, every matched pair of parentheses from sites $j \leq j'$ under the standard matching algorithm corresponds to the endpoints of a balanced interval $\operatorname{int}[j, j']$. (This is easily proven by induction on the time at which the parentheses got matched: At this time, all the parentheses inbetween have already been matched, thus forming balanced intervals, and the newly matched pair merely wraps them in a bigger balanced interval.) Thus, if the algorithm would leave both a ')' and a '(' unmatched, then the (cyclic) interval between the rightmost unmatched ')' and the leftmost unmatched '(' would also be a balanced interval, which would contradict the fact that no balanced interval has the form $\operatorname{int}[i, j]$ with $i > j$. Consequently, the algorithm either leaves only ')' parentheses unmatched, or leaves only '(' parentheses unmatched. The precise outcome depends on which of $|q_1|$ and $|q_2|$

is larger. Consequently, the sites of the unmatched parentheses are precisely the unbalanced sites.

Now, recall that $\mathfrak{s}_1$ merely toggles the unbalanced sites between q_1 and q_2, whereas σ_1 switches the number of unmatched ')'s with the number of unmatched '('s (which, in the case of $\operatorname{word}(\mathbf{q})$, boils down to just turning each unmatched ')' into a '(' or vice versa, because one of the numbers is 0). Since the sites of the unmatched parentheses are precisely the unbalanced sites, this shows that the two maps agree – that is, we have $\operatorname{word}(\mathfrak{s}_1\mathbf{q}) = \sigma_1(\operatorname{word}(\mathbf{q}))$. This proves (B.4).

The equality (B.4) can be rewritten as the commutative diagram

$$\begin{tikzcd}
\{\text{MLQs}\} \arrow[r, "\mathfrak{s}_i"] \arrow[d, "\text{word}"'] & \{\text{MLQs}\} \arrow[d, "\text{word}"] \\
A^* \arrow[r, "\sigma_i"'] & A^*
\end{tikzcd}$$

for all $i \in \{1, 2, \ldots, k-1\}$. In view of the injectivity of the map $\operatorname{word}\colon \{\text{MLQs}\} \to A^*$, this diagram allows us to translate (B.2) into $\mathfrak{s}_i\mathfrak{s}_{i+1}\mathfrak{s}_i = \mathfrak{s}_{i+1}\mathfrak{s}_i\mathfrak{s}_{i+1}$. □

Remark B.4. Our letters $1, \ldots, k$ correspond to the letters $a_k, \ldots, a_1$ in [Lot02], since the definition of σ_i in [Lot02] involves a_i rather than $i+1$ as closing parenthesis and a_{i+1} rather than i as opening one. Also, [Lot02] does not include the letter $\circ$ in the alphabet, but this makes no difference to the proof, since all letters $\circ$ are always frozen.

Remark B.5. The operator σ_i is essentially a combination of co-plactic operators. Moreover, it corresponds to the Weyl group action on a tensor product of crystals [BS17]. Note that the bracketing rule given above is precisely the usual signature rule (see, *e.g.*, [BS17, Sec. 2.4] for a description) for computing tensor products. This arises from considering the MLQ as a binary $m \times n$ matrix and the natural $(\mathfrak{sl}_m \oplus \mathfrak{sl}_n)$-action.

Next, we define two queues corresponding to a word $w \in \mathcal{W}_n$ of type $\mathbf{m}$. Namely, for $k \in \{p_i(\mathbf{m}) \mid i \geq 0\}$, let $[w]_k$ denote the set of the indices $i \in [n]$ corresponding to the k smallest letters w_i of w.

The crucial tools in our proof of Theorem 3.1 will be the following two facts.

Lemma B.6. *Let $w, w' \in \mathcal{W}_n$ be two words of the same type $\mathbf{m}$. Assume that $[w]_k = [w']_k$ for each $k \in \{p_i(\mathbf{m}) \mid i \geq 0\}$. Then, $w = w'$.*

Proof. Fix some $i \geq 1$. The sites containing the letter i in w are the elements of $[w]_{p_i(\mathbf{m})} \setminus [w]_{p_{i-1}(\mathbf{m})}$ (since w has type $\mathbf{m}$). Likewise, the sites containing the letter i in w' are the elements of $[w']_{p_i(\mathbf{m})} \setminus [w']_{p_{i-1}(\mathbf{m})}$. Since our assumption ($[w]_k = [w']_k$) yields $[w]_{p_i(\mathbf{m})} \setminus [w]_{p_{i-1}(\mathbf{m})} = [w']_{p_i(\mathbf{m})} \setminus [w']_{p_{i-1}(\mathbf{m})}$, we conclude that these are the same sites. Since this holds for all letters i, we have $w = w'$. □

Proposition B.7. *Let $u \in \mathcal{W}_n$ be a word of type $\mathbf{m}$. Let $k = p_\alpha(\mathbf{m})$ for some α. Let q be a queue. The dual configuration of $([u]_k, q)$ has the form $\left(q^\dagger, [q(u)]_k\right)$, where $q^\dagger$ is some queue.*

Proof. Let $C' = (q_1', q_2')$ denote the dual configuration of $C = ([u]_k, q)$. The notations $c^\uparrow$ and $c^\downarrow$ as well as the concept of balanced intervals shall refer to C.

Choose a permutation $(i_1, i_2, \ldots, i_n)$ of $(1, 2, \ldots, n)$ such that $u_{i_1} \leq u_{i_2} \leq \cdots \leq u_{i_n}$. Use this permutation to construct $q(u)$ (as in the definition of $q(u)$). For each $p \in [n]$, let j_p be the site j that is found in this construction when $i = i_p$. Thus, $j_p \in q$ if $p \leq |q|$, whereas $j_p \notin q$ if $p > |q|$. Also, $q(u)_{j_1} \leq q(u)_{j_2} \leq \cdots \leq q(u)_{j_n}$, so that $[q(u)]_k = \{j_1, j_2, \ldots, j_k\}$ for each k for which $[q(u)]_k$ is well-defined.

We want to prove that (q'_1, q'_2) has the form $\left(q^\dagger, [q(u)]_k\right)$. In other words, we want to prove that $q'_2 = [q(u)]_k$. If $k = |q|$, then this is obvious (because in this case, the two queues in the configuration C have the same size, so that its dual configuration C' equals C, and thus $q'_2 = q$; but the assumption $k = |q|$ also yields $[q(u)]_k = q$ because of the construction of $q(u)$, and therefore we obtain $q'_2 = q = [q(u)]_k$).

Suppose next that $k < |q|$. Thus, each site in $[u]_k$ is balanced. But $[u]_k = \{i_1, i_2, \ldots, i_k\}$ by the definition of the permutation.

If S is a set of sites, then the *connected components* of S are the maximal intervals contained in S.

We say that a closed cyclic interval $\operatorname{int}[i, j]$ is *top-heavy* if for each $\ell \in \operatorname{int}[i, j]$, we have $c^\uparrow[i, \ell] \geq c^\downarrow[i, \ell]$. Thus, a balanced cyclic interval is just a top-heavy cyclic interval $\operatorname{int}[i, j]$ satisfying $c^\uparrow[i, j] = c^\downarrow[i, j]$.

For each $p \in [k]$, we define an interval I_p by $I_p = \operatorname{int}[i_p, j_p]$.

We first observe that $I_p \cap q \subseteq \{j_1, j_2, \ldots, j_p\}$ for each $p \in [k]$. [*Proof.* Let $p \in [k]$. Recall that the site j_p is found (in Phase II of the algorithm computing $q(u)$) as the first site $j \in q$ weakly to the right of i_p such that $q(u)_j$ is not set yet. In other words, j_p is the first site $j \in q$ weakly to the right of i_p that is not one of $j_1, j_2, \ldots, j_{p-1}$ (since the sites $j \in q$ such that $q(u)_j$ has already been set are $j_1, j_2, \ldots, j_{p-1}$). In other words, all sites in $I_p = \operatorname{int}[i_p, j_p]$ that belong to q must be among $j_1, j_2, \ldots, j_{p-1}, j_p$. In other words, $I_p \cap q \subseteq \{j_1, j_2, \ldots, j_p\}$, qed.]

Let $U = I_1 \cup I_2 \cup \cdots \cup I_k$. Then, $U \cap q \subseteq \{j_1, j_2, \ldots, j_k\}$ (since $I_p \cap q \subseteq \{j_1, j_2, \ldots, j_p\}$ for each $p \in [k]$). Combining this with $\{j_1, j_2, \ldots, j_k\} \subseteq U \cap q$ (since each j_p satisfies $j_p \in I_p \subseteq U$ and $j_p \in \left\{j_1, j_2, \ldots, j_{|q|}\right\} = q$), we obtain $U \cap q = \{j_1, j_2, \ldots, j_k\}$.

If we had $U = [n]$, then this would rewrite as $q = \{j_1, j_2, \ldots, j_k\}$, which would entail $|q| = k$; this would contradict $k < |q|$. Hence, $U \neq [n]$.

Each connected component of the set U is top-heavy. [*Proof.* Let $\operatorname{int}[a, b]$ be a connected component of U. Then, we must prove that $\operatorname{int}[a, b]$ is top-heavy. Indeed, since U is the union of the connected intervals $I_1, I_2, \ldots, I_k$, its connected component $\operatorname{int}[a, b]$ must have the form $\operatorname{int}[a, b] = \bigcup_{p \in R} I_p$ for some nonempty subset R of $[k]$, and be disjoint from all the I_p with $p \notin R$. Consider this R. Thus, $\operatorname{int}[a, b] \cap \{i_1, i_2, \ldots, i_k\} = \{i_p \mid p \in R\}$ (indeed, $\operatorname{int}[a, b] = \bigcup_{p \in R} I_p$ shows that all of the i_p with $p \in R$ must lie in $\operatorname{int}[a, b] \cap \{i_1, i_2, \ldots, i_k\}$; on the other hand, none of the remaining elements of $\{i_1, i_2, \ldots, i_k\}$ can belong to $\operatorname{int}[a, b]$, since $\operatorname{int}[a, b]$ is disjoint from all the I_p with $p \notin R$). Likewise, $\operatorname{int}[a, b] \cap \{j_1, j_2, \ldots, j_k\} = \{j_p \mid p \in R\}$. From $[u]_k = \{i_1, i_2, \ldots, i_k\}$, we obtain

$$\operatorname{int}[a, b] \cap [u]_k = \operatorname{int}[a, b] \cap \{i_1, i_2, \ldots, i_k\} = \{i_p \mid p \in R\}.$$

From $\operatorname{int}[a,b] \subseteq U$, we obtain

$$\operatorname{int}[a,b] \cap q = \operatorname{int}[a,b] \cap \underbrace{U \cap q}_{=\{j_1,j_2,\ldots,j_k\}} = \operatorname{int}[a,b] \cap \{j_1,j_2,\ldots,j_k\} = \{j_p \mid p \in R\}.$$

Thus, for each $\ell \in \operatorname{int}[a,b]$, the number $c^{\uparrow}[a,\ell] = |\operatorname{int}[a,\ell] \cap [u]_k|$ counts all of the i_p with $p \in R$ that fall into the interval $\operatorname{int}[a,\ell]$, while the number $c^{\downarrow}[a,\ell] = |\operatorname{int}[a,\ell] \cap q|$ counts all of the j_p with $p \in R$ that fall into this interval. Hence, the former number is at least as large as the latter number (because if $p \in R$ is such that j_p falls into $\operatorname{int}[a,\ell]$, then i_p must also fall into $\operatorname{int}[a,\ell]$[10]). We have thus proven that for each $\ell \in \operatorname{int}[i,j]$, we have $c^{\uparrow}[a,\ell] \geq c^{\downarrow}[a,\ell]$. In other words, the interval $\operatorname{int}[a,b]$ is top-heavy, qed.]

Consider again the connected components of U. Each of them is top-heavy (as we have just shown), and thus contains at least as many elements of $[u]_k$ as it contains elements of q. Hence, the set U altogether contains at least as many elements of $[u]_k$ as it contains elements of q, and this inequality becomes an equality only if each connected component of U is balanced. But this inequality does become an equality, because the set U contains all k elements of $[u]_k$ (since $[u]_k = \{i_1,i_2,\ldots,i_k\} \subseteq U$) and contains exactly k elements of q (since $U \cap q = \{j_1,j_2,\ldots,j_k\}$). Thus, each connected component of U is balanced. In other words, U is balanced. Hence, each element of $[q(u)]_k$ is balanced (since $[q(u)]_k = \{j_1,j_2,\ldots,j_k\} = U \cap q \subseteq U$). Due to how the dual configuration (q_1', q_2') was defined, we thus conclude that each element of $[q(u)]_k$ is in q_2'. In other words, $[q(u)]_k \subseteq q_2'$. Combining this with $|q_2'| = |[u]_k| = k = |[q(u)]_k|$, we obtain $q_2' = [q(u)]_k$. Thus, Proposition B.7 is proven in the case when $k < |q|$.

Now suppose $k > |q|$. Let ℓ be the number of classes in u, and consider the contragredient duals q^* and u^* as in Lemma 2.5. Thus, $n-k < n-|q| = |q^*|$. Hence, applying the $k < |q|$ case (proven above) to $n-k$, u^* and q^* instead of k, u and q, we see that $([u^*]_{n-k}, q^*)' = \left(q^{\dagger}, [q^*(u^*)]_{n-k}\right)$ for some queue $q^{\dagger}$. Now,

$$\begin{aligned}\left((q_1')^*, (q_2')^*\right) &= \left(([u]_k)^*, q^*\right)' = \left([u^*]_{n-k}, q^*\right)' \\ &= \left(q^{\dagger}, [q^*(u^*)]_{n-k}\right) = \left(q^{\dagger}, [(q(u))^*]_{n-k}\right),\end{aligned}$$

where

- the first equality follows from (B.1);
- the second equality is because $([u]_k)^* = [u^*]_{n-k}$; and
- the fourth equality follows from Lemma 2.5.

Therefore, $(q_2')^* = [(q(u))^*]_{n-k} = ([q(u)]_k)^*$, so that $q_2' = [q(u)]_k$. Hence, Proposition B.7 is proven in the case when $k > |q|$. □

Proof of Theorem 3.1. Recall that any permutation in $\mathfrak{S}_{\ell-1}$ is a product of simple transpositions $s_1, s_2, \ldots, s_{\ell-2}$. Hence, in order to prove Theorem 3.1, it suffices to show that $\langle u \rangle_{\sigma} = \langle u \rangle_{\sigma s_i}$ for each $\sigma \in \mathfrak{S}_{\ell-1}$ and $i \in [\ell-2]$. Then, Theorem 3.1

[10] In fact, let $p \in R$ be such that $j_p \in \operatorname{int}[a,\ell]$. But $\operatorname{int}[i_p, j_p] = I_p$ is a subinterval of $\operatorname{int}[a,b]$, due to $\operatorname{int}[a,b] = \bigcup_{p \in R} I_p$. The interval $\operatorname{int}[a,\ell]$ is a "prefix" of $\operatorname{int}[a,b]$; thus, every subinterval of $\operatorname{int}[a,\ell]$ that ends inside $\operatorname{int}[a,\ell]$ must also begin inside $\operatorname{int}[a,\ell]$. Applying this to the subinterval $\operatorname{int}[i_p, j_p]$, we conclude that $i_p \in \operatorname{int}[a,\ell]$, as we wanted to show. Here, we have tacitly used the fact that $\operatorname{int}[a,b] \neq [n]$, which follows from $\operatorname{int}[a,b] \subseteq U \neq [n]$.

follows by induction on length, *i.e.* the minimal number of simple transpositions needed to write σ.

In order to prove $\langle u \rangle_\sigma = \langle u \rangle_{\sigma s_i}$, we need to show, for a σ-twisted MLQ $\mathbf{q}$ of type $\mathbf{m}$ satisfying $u = \mathbf{q}(1^n)$, that $\mathfrak{s}_i \mathbf{q}$ is a σs_i-twisted MLQ of type $\mathbf{m}$ satisfying $u = (\mathfrak{s}_i \mathbf{q})(1^n)$ (since this will show that $\mathfrak{s}_i$ bijects the former MLQs to the latter). The only nontrivial part is showing $u = (\mathfrak{s}_i \mathbf{q})(1^n)$. More generally, we will show that $(\mathfrak{s}_i \mathbf{q})(w) = \mathbf{q}(w)$ for any word $w \in \mathcal{W}_n$. The proof of this claim reduces to showing that for any configuration $C = (q_1, q_2)$ and any word $w \in \mathcal{W}_n$ the dual configuration $\mathfrak{s}_1 C = C' = (q_1', q_2')$ of C satisfies $C'(w) = C(w)$.

Each word w can be obtained from a standard word by a sequence of merges (each of which sends a word u to $\vee^{(k)} u$ for some $k \in \{p_j(\mathbf{m}) \mid j \geq 1\}$, where $\mathbf{m}$ is the type of u). Lemma 3.2 shows that these merges commute with the action of a queue (and thus of an MLQ). Hence, it is sufficient to consider standard words w. Thus, assume that w is standard of type $\mathbf{m}$. It is straightforward to see (using Equation (2.2) and $|q_2'| = |q_1|$ and $|q_1'| = |q_2|$) that the words $C(w) = q_2(q_1(w))$ and $C'(w) = q_2'(q_1'(w))$ have the same type. Let $\mathbf{n}$ be this type. We shall now show that $[C'(w)]_k = [C(w)]_k$ for all $k \in \{p_i(\mathbf{n}) \mid i \geq 0\}$. According to Lemma B.6, this will yield $C'(w) = C(w)$, and thus our proof will be complete.

Let $k \in \{p_i(\mathbf{n}) \mid i \geq 0\}$. Thus, $k \in \{0, 1, \ldots, n\} = \{p_i(\mathbf{m}) \mid i \geq 0\}$ (since w is standard). Hence, $[w]_k$ is well-defined. Note that $q_i(w)$ and $q_i'(w)$ are also standard words. Using Proposition B.7 to compute dual configurations, we can see how the MLQ $\mathbf{q} = ([w]_k, q_1, q_2)$ transforms under the action of $\mathfrak{s}_1\mathfrak{s}_2\mathfrak{s}_1$: Namely, we have

$$\begin{aligned}
([w]_k, q_1, q_2) &\xmapsto{\mathfrak{s}_1} (*, [q_1(w)]_k, q_2) \\
&\xmapsto{\mathfrak{s}_2} \left(*, *, \left[q_2(q_1(w))\right]_k\right) \\
&\xmapsto{\mathfrak{s}_1} \left(*, *, \left[q_2(q_1(w))\right]_k\right),
\end{aligned}$$

where $*$ denotes some queue. Likewise, the action of $\mathfrak{s}_2\mathfrak{s}_1\mathfrak{s}_2$ is given by

$$\begin{aligned}
([w]_k, q_1, q_2) &\xmapsto{\mathfrak{s}_2} ([w]_k, q_1', q_2') \\
&\xmapsto{\mathfrak{s}_1} (*, [q_1'(w)]_k, q_2') \\
&\xmapsto{\mathfrak{s}_2} \left(*, *, \left[q_2'(q_1'(w))\right]_k\right).
\end{aligned}$$

(See Figure 8 for the actions depicted using crossing diagrams.) Yet, the two maps are equal by Proposition B.3. Thus, the resulting MLQs must be identical:

$$\left(*, *, \left[q_2'(q_1'(w))\right]_k\right) = \left(*, *, \left[q_2(q_1(w))\right]_k\right).$$

Hence, we have

$$\left[q_2'(q_1'(w))\right]_k = \left[q_2(q_1(w))\right]_k.$$

In other words, $[C'(w)]_k = [C(w)]_k$. ☐

References

[AAMP11] Chikashi Arita, Arvind Ayyer, Kirone Mallick, and Sylvain Prolhac. Recursive structures in the multispecies TASEP. *J. Phys. A*, 44(33):335004, 2011.

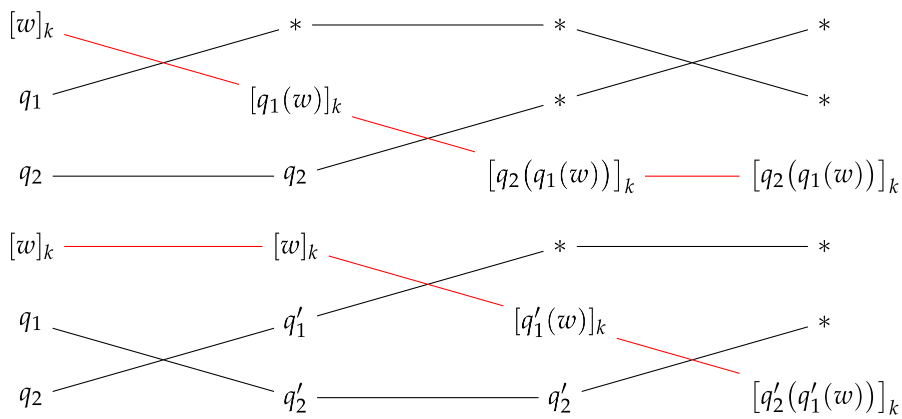


FIGURE 8. Crossing diagrams representing the action of $\mathfrak{s}_1\mathfrak{s}_2\mathfrak{s}_1$ (top) and $\mathfrak{s}_2\mathfrak{s}_1\mathfrak{s}_2$ (bottom).

[AGS18] Erik Aas, Darij Grinberg, and Travis Scrimshaw. Multiline queues with spectral parameters. *Séminaire Lotharingien de Combinatoire*, 80B:46, 2018. http://www.mat.univie.ac.at/~slc/wpapers/FPSAC2018/46-Aas-Grinberg-Scrimshaw.html.

[AHR98] Peter F. Arndt, Thomas Heinzel, and Vladimir Rittenberg. First-order phase transitions in one-dimensional steady states. *J. Stat. Phys.*, 90:783–815, 1998.

[AHR99] Peter F. Arndt, Thomas Heinzel, and Vladimir Rittenberg. Spontaneous breaking of translational invariance and spatial condensation in stationary states on a ring. I. The neutral system. *J. Statist. Phys.*, 97(1-2):1–65, 1999.

[AL14] Arvind Ayyer and Svante Linusson. An inhomogeneous multispecies TASEP on a ring. *Adv. in Appl. Math.*, 57:21–43, 2014.

[AL18] Erik Aas and Svante Linusson. Continuous multi-line queues and TASEP. *Ann. Inst. H. Poincaré Comb. Phys. Interact.*, 2018. To appear, arXiv:1501.04417v2.

[AM13] Chikashi Arita and Kirone Mallick. Matrix product solution of an inhomogeneous multi-species TASEP. *J. Phys. A*, 46(8):085002, 11, 2013.

[Ang06] Omer Angel. The stationary measure of a 2-type totally asymmetric exclusion process. *J. Combin. Theory Ser. A*, 113(4):625–635, 2006.

[AS18] Sami Assaf and Dominic Searles. Kohnert tableaux and a lifting of quasi-schur functions. *Journal of Combinatorial Theory, Series A*, 156:85–118, May 2018.

[Bax89] Rodney J. Baxter. *Exactly solved models in statistical mechanics*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London, 1989. Reprint of the 1982 original.

[BE07] Richard A. Blythe and Martin R. Evans. Nonequilibrium steady states of matrix-product form: a solver's guide. *J. Phys. A*, 40(46):R333–R441, 2007.

[BP14] Alexei Borodin and Leonid Petrov. Integrable probability: from representation theory to Macdonald processes. *Probab. Surv.*, 11:1–58, 2014.

[BS17] Daniel Bump and Anne Schilling. *Crystal Bases: Representations and Combinatorics*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2017.

[CdGW15] Luigi Cantini, Jan de Gier, and Michael Wheeler. Matrix product formula for Macdonald polynomials. *J. Phys. A*, 48(38):384001, 25, 2015.

[CEM01] M. Clincy, M. R. Evans, and D. Mukamel. Symmetry breaking through a sequence of transitions in a driven diffusive system. *J. Phys. A*, 34(47):9923–9937, 2001.

[CL99] Tom Chou and Detlef Lohse. Entropy-driven pumping in zeolites and biological channels. *Phys. Rev. Lett.*, 82(17):3552–3555, 1999.

[CW11] Sylvie Corteel and Lauren K. Williams. Tableaux combinatorics for the asymmetric exclusion process and Askey-Wilson polynomials. *Duke Math. J.*, 159(3):385–415, 2011.

[DEHP93] B. Derrida, M. R. Evans, V. Hakim, and V. Pasquier. Exact solution of a 1D asymmetric exclusion model using a matrix formulation. *J. Phys. A*, 26(7):1493–1517, 1993.

[DJLS93] B. Derrida, S. A. Janowsky, J. L. Lebowitz, and E. R. Speer. Exact solution of the totally asymmetric simple exclusion process: shock profiles. *J. Statist. Phys.*, 73(5-6):813–842, 1993.

[DK05] Vladimir I. Danilov and Gleb A. Koshevoy. Arrays and the combinatorics of young tableaux. *Russian Mathematical Surveys*, 60(2):269–334, 2005.

[EFGM95] M. R. Evans, D. P. Foster, C. Godréche, and D. Mukamel. Spontaneous symmetry breaking in a one dimensional driven diffusive system. *Phys. Rev. Lett.*, 74(2):208–211, 1995.

[EFM09] Martin R. Evans, Pablo A. Ferrari, and Kirone Mallick. Matrix representation of the stationary measure for the multispecies TASEP. *J. Stat. Phys.*, 135(2):217–239, 2009.

[EKKM98] M. R. Evans, Y. Kafri, H. M. Koduvely, and D. Mukamel. Phase separation in one-dimensional driven diffusive systems. *Phys. Rev. Lett.*, 80:425, 1998.

[EPSZ05] David W. Erickson, Gunnar Pruessner, Beate Schmittmann, and Royce K. P. Zia. Spurious phase in a model for traffic on a bridge. *J. Phys. A*, 38(41):L659–L665, 2005.

[Fer92] Pablo A. Ferrari. Shock fluctuations in asymmetric simple exclusion. *Prob. Theor. Rel. Fields*, 91(1):81–101, 1992.

[FF94a] Pablo A. Ferrari and Luiz R. G. Fontes. Current fluctuations for the asymmetric simple exclusion process. *Ann. Probab.*, 22(2):820–832, 1994.

[FF94b] Pablo A. Ferrari and Luiz R. G. Fontes. Shock fluctuations in the asymmetric simple exclusion process. *Probab. Theory Related Fields*, 99(2):305–319, 1994.

[FFK94] Pablo A. Ferrari, Luiz R. G. Fontes, and Yoshiharu Kohayakawa. Invariant measures for a two-species asymmetric process. *J. Statist. Phys.*, 76(5-6):1153–1177, 1994.

[FM06] Pablo A. Ferrari and J. B. Martin. Multi-class processes, dual points and $M/M/1$ queues. *Markov Process. Related Fields*, 12(2):175–201, 2006.

[FM07] Pablo A. Ferrari and James B. Martin. Stationary distributions of multi-type totally asymmetric exclusion processes. *Ann. Probab.*, 35(3):807–832, 2007.

[GLE$^+$95] C. Godréche, J. M. Luck, M. R. Evans, D. Mukamel, S. Sandow, and E. R. Speer. Spontaneous symmetry breaking: exact results for a biased random walk model of an exclusion process. *J. Phys. A*, 28(21):6039–6071, 1995.

[Gor17] Alexey L. Gorodentsev. *Algebra. II. Textbook for students of mathematics*. Springer, Cham, 2017. Originally published in Russian, 2015.

[GV85] Ira Gessel and Gérard Viennot. Binomial determinants, paths, and hook length formulae. *Adv. in Math.*, 58(3):300–321, 1985.

[JNH$^+$09] Rui Jiang, Katsuhiro Nishinari, Mao-Bin Hu, Yong-Hong Wu, and Qing-Song Wu. Phase separation in a bidirectional two-lane asymmetric exclusion process. *J. Stat. Phys.*, 136(1):73–88, 2009.

[Joh00] Kurt Johansson. Shape fluctuations and random matrices. *Comm. Math. Phys.*, 209(2):437–476, 2000.

[KKM$^+$92] Seok-Jin Kang, Masaki Kashiwara, Kailash C. Misra, Tetsuji Miwa, Toshiki Nakashima, and Atsushi Nakayashiki. Perfect crystals of quantum affine Lie algebras. *Duke Math. J.*, 68(3):499–607, 1992.

[KLM$^+$02] Y. Kafri, E. Levine, D. Mukamel, G. M. Schütz, and J. Török. Criterion for phase separation in one-dimensional driven systems. *Phys. Rev. Lett.*, 89:035702, 2002.

[KMO15] Atsuo Kuniba, Shouya Maruyama, and Masato Okado. Multispecies TASEP and combinatorial R. *J. Phys. A*, 48(34):34FT02, 19, 2015.

[KMO16a] Atsuo Kuniba, Shouya Maruyama, and Masato Okado. Inhomogeneous generalization of a multispecies totally asymmetric zero range process. *J. Stat. Phys.*, 164(4):952–968, 2016.

[KMO16b] Atsuo Kuniba, Shouya Maruyama, and Masato Okado. Multispecies TASEP and the tetrahedron equation. *J. Phys. A*, 49(11):114001, 22, 2016.

[KMO16c] Atsuo Kuniba, Shouya Maruyama, and Masato Okado. Multispecies totally asymmetric zero range process: I. Multiline process and combinatorial R. *J. Integrable Syst.*, 1(1), 2016.

[KMO16d] Atsuo Kuniba, Shouya Maruyama, and Masato Okado. Multispecies totally asymmetric zero range process: II. Hat relation and tetrahedron equation. *J. Integrable Syst.*, 1(1), 2016.

[KNL05] Stefan Klumpp, Theo M. Nieuwenhuizen, and Reinhard Lipowsky. Movements of molecular motors: Ratches, random walks, and traffic phenomena. *Physica E*, 29(1):380–389, 2005.

[Kru91] Joachim Krug. Boundary-induced phase transitions in driven diffusive systems. *Phys. Rev. Lett.*, 67(14):1882–1885, 1991.

[KT07] Masahiro Kasatani and Yoshihiro Takeyama. The quantum Knizhnik-Zamolodchikov equation and non-symmetric Macdonald polynomials. *Funkcial. Ekvac.*, 50(3):491–509, 2007.

[Lam15] Thomas Lam. The shape of a random affine Weyl group element and random core partitions. *Ann. Probab.*, 43(4):1643–1662, 2015.

[Lev73] David G. Levitt. Dynamics of a single pore: non-Fickian behavior. *Phys. Rev. A*, 8(6):3050–3054, 1973.

[Lig76] Thomas M. Liggett. Coupling the simple exclusion process. *Ann. Probability*, 4(3):339–356, 1976.

[Lig99] Thomas M. Liggett. *Stochastic Models of Interacting Systems: Contact, Voter and Exclusion Processes*. Springer-Verlag Berlin Heidelberg, 1999.

[Lin73] Bernt Lindström. On the vector representations of induced matroids. *Bull. London Math. Soc.*, 5:85–90, 1973.

[LLPT18] D. Laksov, A. Lascoux, P. Pragacz, and A. Thorup. The LLPT Notes. Preprint, http://web.math.ku.dk/noter/filer/sympol.pdf, March 28, 2018.

[Lot02] M. Lothaire. *Algebraic Combinatorics on Words*, volume 90 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2002.

[MGP68] Carolyn T. MacDonald, Julian H. Gibbs, and Allen C. Pipkin. Kinetics of biopolymerization on nuclear acid templates. *Biopolymers*, 6:1–25, 1968.

[NY97] Atsushi Nakayashiki and Yasuhiko Yamada. Kostka polynomials and energy functions in solvable lattice models. *Selecta Math. (N.S.)*, 3(4):547–599, 1997.

[PEM09] S. Prolhac, M. R. Evans, and K. Mallick. The matrix product solution of the multispecies partially asymmetric exclusion process. *J. Phys. A*, 42(16):165004, 25, 2009.

[PK07] Ekaterina Pronina and Anatoly B. Kolomeisky. Spontaneous symmetry breaking in two-channel asymmetric exclusion processes with narrow entrances. *J. Phys. A*, 40(10):2275–2286, 2007.

[RSS00] N. Rajewsky, T. Sasamoto, and E. R. Speer. Spatial particle condensation for an exclusion process on a ring. *Physica A*, 289(1-4):123–142, 2000.

[Sag18] The Sage Developers. *Sage Mathematics Software (Version 8.3)*, 2018. http://www.sagemath.org.

[SCc08] The Sage-Combinat community. Sage-Combinat: enhancing Sage as a toolbox for computer exploration in algebraic combinatorics, 2008. http://combinat.sagemath.org.

[Sch00] Andreas Schadschneider. Statistical physics of traffic flow. *Physica A*, 285:101–120, 2000.

[Sch01] G. M. Schütz. Exactly solvable models for many-body systems far from equilibrium. In C. Domb and J. L. Lebowitz, editors, *Phase transitions and critical phenomena, Vol. 19*, pages 1–251. Academic Press, San Diego, CA, 2001.

[Shi02] Mark Shimozono. Affine type A crystal structure on tensor products of rectangles, Demazure characters, and nilpotent varieties. *J. Algebraic Combin.*, 15(2):151–187, 2002.

[Spi70] Frank Spitzer. Interaction of Markov processes. *Adv. in Math.*, 5:246–290 (1970), 1970.

[Sta99] Richard P. Stanley. *Enumerative combinatorics. Vol. 2*, volume 62 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1999. With a foreword by Gian-Carlo Rota and appendix 1 by Sergey Fomin.

[Sta12] Richard P. Stanley. *Enumerative combinatorics. Vol. 1*, volume 49 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, second edition, 2012.

[Ste01] John R. Stembridge. Multiplicity-free products of Schur functions. *Ann. Comb.*, 5(2):113–121, 2001.

[SZ95] Beate Schmittmann and Royce K. P. Zia. Statistical mechanics of driven diffusive systems. In C. Domb and J. L. Lebowitz, editors, *Phase Transitions and Critical Phenomena, Vol. 17*, pages 3–214. Academic Press, San Diego, CA, 1995.

[TJHJ17] Bo Tian, Rui Jiang, Mao-Bin Hu, and Bin Jia. Spurious symmetry-broken phase in a bidirectional two-lane ASEP with narrow entrances: A perspective from mean field analysis and current minimization principle. *Chin. Phys. B*, 26(2):020503, 2017.

[TW09] Craig A. Tracy and Harold Widom. Asymptotics in ASEP with step initial condition. *Comm. Math. Phys.*, 290(1):129–154, 2009.

[USW04] Masaru Uchiyama, Tomohiro Sasamoto, and Miki Wadati. Asymmetric simple exclusion process with open boundaries and Askey-Wilson polynomials. *J. Phys. A*, 37(18):4985–5002, 2004.

[vL06] Marc A. A. van Leeuwen. Double crystals of binary and integral matrices. *Electron. J. Combin.*, 13(1):Research Paper 86, 93, 2006.

[Zam80] Alexander B. Zamolodchikov. Tetrahedra equations and integrable systems in three-dimensional space. *Zh. Èksper. Teoret. Fiz.*, 79(2):641–664, 1980.

(E. Aas) Department of Mathematics, Pennsylvania State University, McAllister Building, State College, PA 116802, USA

Email address: eaas@kth.se

(D. Grinberg) School of Mathematics, University of Minnesota, 206 Church St. SE, Minneapolis, MN 55455

Email address: darijgrinberg@gmail.com

URL: http://www.cip.ifi.lmu.de/~grinberg/

(T. Scrimshaw) School of Mathematics and Physics, The University of Queensland, St. Lucia, QLD 4072, Australia

Email address: tcscrims@gmail.com

URL: https://people.smp.uq.edu.au/TravisScrimshaw/