Control over the KKR bijection with respect to the nesting structure on rigged configurations and a CSP instance involving Motzkin numbers

by

William Chan

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

Waterloo, Ontario, Canada, 2024

Author's Declaration

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

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

Abstract

There are two disjoint main projects that this thesis covers. The Motzkin numbers are a sort of "relaxed version" of the Catalan numbers. For example, Catalan numbers count perfect non-crossing matchings, while Motzkin numbers count not necessarily perfect non-crossing matchings. The first project deals with instances of the cyclic sieving phenomenon involving Motzkin numbers and their standard $q-$analogue. We also show that the standard $q-$analogue for Motzkin numbers satisfy a similar generating series interpretation to that of the $q-$Catalan numbers. The second project deals with understanding the Kerov-Kirillov-Reshetikhin (KKR) bijection between semistandard tableaux and rigged configurations with a particular emphasis on the standard case. In particular, we understand inducing perturbations on the corresponding rigged configuration via direct operations on the tableau. We develop a technique to take a standard tableau T, and output a new standard tableau T' which has the same corresponding rigged configuration up to a rigging on any desired row of the first rigged partition. We also develop an alternate technique to Kuniba, Okado, Sakamoto, Takagi, and Yamada that "unwraps" the natural nesting structure on rigged configurations. The primary operation from which all the above follows from is "raise" which we introduce and give various combinatorial models for. The operation raise induces a very simple, controlled perturbation on the corresponding rigged configuration. Our results are formulated in terms of paths or (classically) highest weight elements of tensor products of Kirillov-Reshetikhin crystals.

Acknowledgements

I would like to thank Olya Mandelshtahm for supervising me throughout my Master's career, and in particular for suggesting the first project that this thesis covers. I would also like to thank Ben Webster and Oliver Pechenik for agreeing to read my thesis and providing their comments. Finally, I would also like to thank Travis Scrimshaw for helpful discussions.

Contents

List of Figures

0.1 Introduction

This thesis details the work that the author has done during his Master's degree. It consists of two unrelated projects, one investigating the mysterious cyclic sieving phenomenon (CSP) involving a relaxed analogue of the Catalan numbers known as Motzkin numbers, and the other involving understanding the KKR bijection between semistandard tableaux and rigged configurations. The work done in investigating the CSP involving Motzkin numbers will be covered in Chapter 1. The work done on rigged configurations and the KKR bijection willbe covered in Chapter 2.

Each chapter has its own introduction, and we give an additional overview of both chapters here. Because the introduction to Chapter 1 occurs soon after this section, the overview of Chapter 1 given here will be minimal.

The overview of Chapter 2 given here contrasts the introduction in Chapter 2, Section 2.1, by being less technical, wider in scope, and providing more context from the literature.

0.1.1 Motzkin Numbers and the Cyclic Sieving Phenomenon

In some sense, the objects which Motzkin numbers count are objects that Catalan numbers count except they are "injected with some amount of empty space". For example, while the Catalan numbers count perfect non-crossing matchings on $2n$ vertices, the Motzkin numbers count (not necessarily perfect) non-crossing matchings on n vertices. Here, the "empty space" that is injected are the isolated vertices of the non-crossing matching. This description implies a natural formula for Motzkin numbers in terms of Catalan numbers.

The Catalan numbers have a canonical $q-$analogue, and it is a famous result that substituting q for a root of unity counts the number of perfect non-crossing matchings which are fixed under a rotation of the diagram. This is a specific instance of a more general phenomenon known as the cyclic sieving phenomenon (CSP).

Using the natural formula for Motzkin numbers in terms of Catalan numbers, one can derive the natural $q-$analogue for Motzkin numbers. A natural question to ask is whether root of unity evaluations of this $q-$analogue similarly count the number of (not necessarily perfect) non-crossing matchings fixed under rotation of the diagrams. We confirm this in Theorem 1.3.2. The method we employ was used by Rhoades in [14, (8.10)]. We discuss and spell out the nuances of this argument in great detail. Moreover, one can directly recover Theorem 1.3.2 through careful usage of [14, (8.10)].

We also show (Theorem 1.4.18) that the canonical $q-$analogue of the Motzkin numbers admit a generating series interpretation in terms of the major index statistic on words, mirroring the canonical $q-$analogue of the Catalan numbers. This has implications for how the the cyclic action is related to the major index. In particular, it is a general result (see [12]) that any polynomial which counts the fixed points of a cyclic action via root of unity evaluations yield a combinatorial description of its coefficients in terms of the cyclic action.

The relationship between Motzkin numbers and Catalan numbers is not an uncommon one. In the specific case of CSP, the idea of injecting "empty space" into objects to obtain new instances of CSP has appeared in other places in the literature (for example see [11]). Hence, we generalize our techniques to show that for any two polynomials whose root of unity evaluations count fixed points of a cyclic action on finite sets X, Y respectively naturally extend to a polynomial whose root of unity evaluations count fixed points of the induced cyclic action on the set of objects of X which are "injected" with the objects in Y. To reconcile this with Motzkin and Catalan numbers,

X are perfect non-crossing matchings, and Y consists of empty graphs (on which the cyclic action is trivial).

0.1.2 Rigged Configurations and the KKR bijection

Rigged configurations were introduced by Kerov, Kirillov, and Reshetikhin in [7], [6]. They are a finite sequence of partitions, with numbers (called riggings) attached to the rows subject to certain conditions. They arose from the study of the spin 1/2 XXX model of Heisenberg quantum spin chains. In particular, rigged configurations paramaterize solutions to the "Bethe Ansatz"— a heuristic method in mathematical physics. Despite their origin in mathematical physics, they have found use and application in algebraic combinatorics.

The Kerov-Kirillov-Reshetikhin (KKR) bijection is an algorithm which bijectively maps (partition content) semistandard tableaux to rigged configurations.

The KKR bijection has many remarkable properties by which we are able to understand certain aspects of tableau combinatorics better on their corresponding rigged configurations. For example, the charge of a tableau behaves more nicely on the corresponding rigged configuration. Indeed, charge can be thought of as a "continuous" function on rigged configurations in the sense that the charge is linear in the riggings. That is, the way the charge of two rigged configurations with the same sequence of shapes differ, is by the difference of their total riggings.

Another example where tableau combinatorics is more natural on rigged configurations is evacuation. The description of evacuation on tableau is complicated. However, the description on rigged configurations is very simple and amounts to "inverting" all the riggings.

The last nice property we will mention here is that the definition of the KKR bijection easily extends to all (not necessarily partition content) semistandard tableaux with no added complexity. Over this larger domain, the KKR bijection is no longer a bijection. However, two semistandard tableaux yield the same rigged configuration if and only if they are equivalent modulo the action of the symmetric group generated by crystal reflection operators/Lascoux-Schützenberger involutions.

Despite these properties, the KKR bijection is still not well understood due to its opaque description. The algorithm is chaotic in the sense that small changes to the rigged configuration, or semistandard tableau lead to unpredictable changes in the corresponding objects.

Nevertheless, the existence of these properties do provide evidence that the interpretation of some of the basic data of a rigged configuration has significant meaning on the corresponding tableau.

Currently, even simple questions like the meaning of the shapes in the rigged configuration are largely unknown. We understand the zeroeth and first shapes, but only conjectural formulas are known for the rest. The zeroeth shape corresponds to the content of the semistandard tableau. The first shape is known to describe the asymptotic state of the multicolour box-ball system (a soliton decomposition— see [9] or Theorem 2.7.17). The multicolour box-ball system is highly related to integrable systems, and hence the study of the first rigged partition is of particular interest to many mathematicians.

In [10], Lam, Pylyavskyy, and Sakamoto were able to describe the first shape directly on the tableau and conjectured similar formulas for the other shapes. Following their work, Scrimshaw found an interpretation for the riggings of the first rigged partition, and conjectured similar formulas for the other rigged partitions [17]. Both interpretations given in [10] and [17] rely on tropicalizations of cylindric loop and loop Schur functions. Using the persepctive of the multicolour box-ball system, Kuniba, Okado, Sakamoto, Takagi, and Yamada showed that the natural time evolution

1	3	5	6	$\mapsto$	1	2	3	3
2	4				2	3		
7	8				4	4		

Figure 1: An example of colouring the cells of a standard tableau by ascent sequences

rule changes the corresponding rigged configuration in a very controlled manner (see [9][Proposition 2.6] or Theorem 2.7.15). The shapes remain the same, and the riggings on the first rigged partition increase by the length of the row they are attached to. In fact, they show that similar notions of time evolution do the same to the other rigged partitions.

In this present work, we also study the first rigged partition. However, it should be noted that our motivations are purely to investigate how the naturally presented data of rigged configurations manifest themselves on the corresponding tableaux through the KKR bijection. We largely restrict ourselves to standard tableaux and their corresponding rigged configurations. However our study of this case is highly successful; we manage to obtain and develop many results and techniques to understand and manipulate the first rigged partition directly on the tableau.

Our study of the shape is less complete than Lam, Pylyavskyy, and Sakamoto. However, our perspective is different. Our results for the shape are largely due to imposing control on the KKR bijection. That is, finding sufficient conditions where certain behaviour of the KKR bijection becomes predictable. Our conditions are simple and occur frequently. Under these conditions, we understand exactly how the first shape of the rigged configuration behaves with respect to the individual steps of the KKR bijection (Theorem 2.4.21), which is not done in [10]. Using a statistic related to these conditions, we also derive a simple characterization of the smallest row of the first shape (Theorem 2.7.35).

Our study of the riggings differ from [17] in the sense that our results are relative, while Scrimshaw's are absolute. In particular, we study what perturbations of the riggings do to the corresponding tableau, while Scrimshaw describes what the riggings should be from the tableau. The way our results differ from Kuniba, Okado, Sakamoto, Takagi, and Yamada, is that our change in the riggings is "smaller" (and hence finer) than theirs.

To be more specific about our results, we define a simple operation, "raise" on words which leaves the corresponding rigged configuration unchanged aside from one small controlled perturbation of its riggings.

The operation raise is described as a refactorization (see Section 2.6), but because this refactorization is unique, there are a number of different combinatorial models for it. The most practical model is a certain cyclic shift of the letters in the corresponding Yamanouchi word. However, there are other realizations of raise that are far more striking.

For example, if we colour the cells of a standard tableau based on which maximal north east sequence it belongs to (with respect to the typical standardization reading) then we obtain a semistandard tableau whose fillings are the colours (see Figure 1). This procedure is also known as *destandardization* in [1]. The operation raise can then be interpreted as performing a crystal raising operator on this semistandard tableau before restandardizing it.

The effect that raise has on the corresponding rigged configuration is small. In fact, it increases one specific rigging on one specific row in the first rigged partition by one. Despite this, we show that careful usage of raise allows one complete control over the riggings of the first rigged partition.

The operation raise is an example of what we later define as a *spell*. Roughly, a spell is a function on a tableau which preserves the “nesting structure” of the corresponding rigged configuration. That is, it preserves the rigged configuration obtained by removing the zeroeth partition. Many times we will be interested in performing spells on subtableaux. That is, we remove the last few cells of a tableau, perform a spell on the resulting tableau, and then and the removed cells back. When the result of this procedure is also a spell, then we say it is an *induced spell*. Understanding certain sufficient conditions for a spell to be induced (Theorem 2.8.6) turns out to yield powerful techniques to move between two tableaux whose corresponding rigged configurations have the same nesting structure.

In particular, using induced spells which come from applications of raise to subtableaux, we are able to manipulate any rigging of the first rigged partition by only considering the corresponding tableau. This yields a procedure to move between any two standard tableaux whose corresponding rigged configurations only differ at the first rigged partition.

We can also use raise (and the definition of the KKR bijection) to derive an alternate procedure to [9] that “unwraps” the natural nesting structure of a rigged configuration directly on the corresponding tableau.

Understanding this nesting structure allows one to compute the underlying configuration (the sequence of partitions) directly on the tableau. Indeed, each partition in the configuration is the content of the semistandard tableau obtained after unravelling each layer of the nesting structure.

The technique we present only works on standard tableaux. However, we provide an algorithm to standardize a semistandard tableau in a way which preserves the nesting structure. As such, the techniques in this thesis can be used to completely unwrap the nesting structure directly on the corresponding tableau.

Chapter 1

Cyclic Sieving Phenomenon of Motzkin Numbers

1.1 Introduction

The notion of a "$q-$analogue" appears often in enumerative combinatorics. They typically appear as finite generating function generalizations of counting formulae. One of the first things a student learns in a class on generating functions is that the indeterminant should not take on any real or complex value. That is, despite its name, one should not treat a generating function as a function. However, when a generating function is in fact a polynomial, this interpretation becomes reasonable. One might ask whether this perspective has any combinatorial significance. Surprisingly, it turns out that many times substituting the "q" in a $q-$analogue for a root of unity leads to an intelligble result.

This phenomenon is known as the cyclic sieving phenomenon (CSP) which was first defined explicitly by Reiner, Stanton and White in [12]. To be precise, let X be a finite set, $\langle g \rangle$ a cyclic group with generator g of order n which acts on X and f a polynomial. The triple $(X, \langle g \rangle, f)$ is said to exhibit the *cyclic sieving phenomenon* (CSP) if $f(\zeta_n^j) = \#\{x \in X : g^j \cdot x = x\}$ where $\zeta_n := e^{\frac{2\pi i}{n}}$ is a primitive nth root of unity.

The most interesting instances of CSP are those that occur "in nature". Instead of cooking the polynomial f given a finite set X with a cyclic action, we are more interested the reverse problem. That is, if we start with a $q-$analogue f, can we find a finite set X with a cyclic group $\langle g \rangle$ acting on it so that $(X, \langle g \rangle, f)$ exhibits CSP?

To investigate this question, we should be more explicit with what a $q-$analogue is and what makes one "good" or "canonical". The term "$q-$analogue of an expression" does not have a precise definition, with the only requirement being to be a function of q which approaches the original expression as $q \to 1$. Despite there being no formal definition of a $q-$analogue, there are standard $q-$analogues of the typical building blocks of combinatorial formulae, which enable us to construct many "canonical" $q-$analogues of combinatorial expressions. These standard $q-$analogues are: the $q-$integer $[n]_q := 1+q+...+q^{n-1}$, the $q-$factorial $[n]_q! := [n]_q[n-1]_q...[2]_q[1]_q$, and the $q-$binomial coefficient

$$\begin{bmatrix} n \\ k \end{bmatrix} := \frac{[n]_q!}{[k]_q![n-k]_q!}.$$

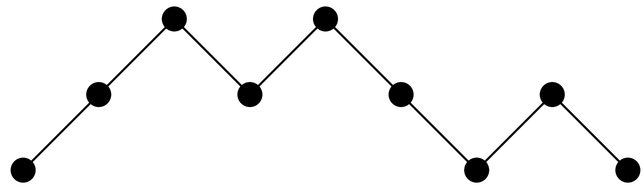


Figure 1.1: A Dyck path from $(0,0)$ to $(8,0)$

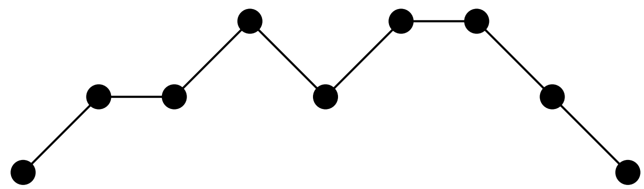


Figure 1.2: A Motzkin path from $(0,0)$ to $(8,0)$

As an example, consider the Catalan numbers C_n. The nth Catalan number is defined as the number of *Dyck paths* from $(0,0)$ to $(2n,0)$. A Dyck path from $(0,0)$ to $(2n,0)$ is a sequence of integer steps (either one unit north and one unit east or one unit south and one unit east) from $(0,0)$ to $(2n,n)$ which does not fall below the x-axis (see Figure 1.1). There is a well known formula for the nth Catalan numbers, $C_n = \frac{1}{n+1}\binom{2n}{n}$.

The formula for C_n then has a "canonical" $q-$analogue obtained by replacing the building blocks in the expression with the standard $q-$analogues discussed earlier. That is,,

$$C_n(q) = \frac{1}{[n+1]_q}\begin{bmatrix}2n\\n\end{bmatrix}_q. \tag{1.1}$$

We would like to know whether or not this $q-$analogue is involved with the cyclic sieving phenomenon in any way. This was first shown by Reiner, Stanton, and White in [12, Theorem 7.2] by finding an instance of CSP involving Narayana numbers, which are in some sense a refinement of the Catalan numbers. However, this instance of CSP is in terms of the non-crossing partitions of $[n]$, which is not the interpretation of the Catalan numbers we will be using. The version of the Catalan numbers we will be primarily interested in are the perfect non-crossing matchings on n vertices, $\mathcal{D}_n$ (see Figure 1.3).

That is, imagine placing the numbers $1, 2, ..., 2n$ on the unit circle equally spaced apart in a clockwise fashion. Each element of $\mathcal{D}_n$ is a drawing of straight lines between the numbers such that every number has exactly one line touching it, and none of the lines cross. Note that there is a natural cyclic action on these objects where the action of the generator is given by rotation $\frac{\pi}{n}$ clockwise. A famous result due to Rhoades and White is that $(\mathcal{D}_n, \langle g\rangle, C_n(q))$ exhibits CSP [13, Theorem 8.2].

There is a related sequence to the Catalan numbers called the Motzkin numbers. Generally, the Motzkin numbers enumerate "relaxed" versions of the objects that Catalan numbers enumerate. For example, instead of Dyck paths, the nth Motzkin number counts the number of *Motzkin paths* from $(0,0)$ to $(2n,0)$. These are defined in the same way, except we allow for horizontal steps (that is, steps which move one unit east but do not move north nor south, see 1.2). As another example, the Motzkin numbers also count the number of (not necessarily perfect) non-crossing matchings on n vertices, $\mathcal{M}_n$ (see 1.3). This description of the nth Motzkin number implies the formula $M_n := \sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{2k} C_k$. Like the nth Catalan number, this expressions is comprised of the three typical building blocks of combinatorial formulae, so we again have the canonical $q-$analogue,

$$M_n(q) := \sum_{k=0}^{\lfloor n/2 \rfloor} \begin{bmatrix} n \\ 2k \end{bmatrix}_q C_k(q). \tag{1.2}$$

We also note that like perfect non-crossing matchings, $\mathcal{M}_n$ admits a natural cyclic action given by rotation of $\frac{2\pi}{n}$ clockwise. Then, a natural question is whether we have an analogous instance of CSP for Motzkin numbers using $\mathcal{M}_n$, the natural cyclic action, and the canonical $q-$analogue $M_n(q)$.

There are two typical approaches to proving an instance CSP: one using representation theory, and one using a pure combinatorial approach. Typically, the representation theory approach is more powerful (see [13]). However, in our case, a pure combinatorial proof will be sufficient. In fact, the approach we take is direct, simple, and extremely transparent (see Proposition 1.3.1).

We will bootstrap off of the existence of a CSP involving Catalan numbers to find many instances of CSP for Motzkin numbers. The technique we use is the same as the one Rhoades used to show [14, (8.10)], except our discussion with the present nuances is highly detailed. In fact, Theorem 1.3.2 can be deduced from [14, (8.10)] by taking $\frac{n-s}{2} + s$ total blocks, and s singleton blocks, and then summing over all s such that $n-s$ is even. Indeed, a non-crossing partition with $\frac{n-s}{2} + s$ total blocks, and s singleton blocks must have every non-singleton block be a block of size 2. Therefore, any non-crossing partition with these parameters is simultaneously a non-crossing matching.

We also show the natural $q-$analogue of Motzkin numbers satisfy a major index formula which mirrors the natural $q-$analogue of Catalan numbers. We do this both for the classical definition of major index, as well as an alternate definition which sums over ascents rather than descents. We also show these formulas are in some sense independent of the way we choose to define a Motzkin word.

The paradigm illustrated in this chapter for obtaining new CSPs from old ones via injected "empty space" has appeared previously in the literature. In [14], Rhoades used this idea to produce another proof of an instance of CSP involving the Narayana numbers, originally due to Reiner, Stanton, and White in [12]. A similar idea is also presented by Mandel and Pechenik in [11].

We make this paradigm explicit by including a generalization of some of the methods we use. A notable use case of this generalization finds an instance of CSP where the finite set X is the set of linear extensions of the poset $P_1 + P_2$, and the cyclic action is promotion.

1.2 Catalan and Motzkin paths as words

In this short section, we will explain how Dyck paths and Motzkin paths will be used in practice.

In Section1.1, we introduced Catalan and Motzkin numbers, as well as certain objects that they count. In particular, we defined Dyck paths and Motzkin paths as paths in the integer lattice $\mathbb{Z}^2$.

However, in practice it is more useful to think of Dyck paths and Motzkin paths as words. That is, we realize a Dyck path as a word with letters 1 and 2 by following the path left to right and

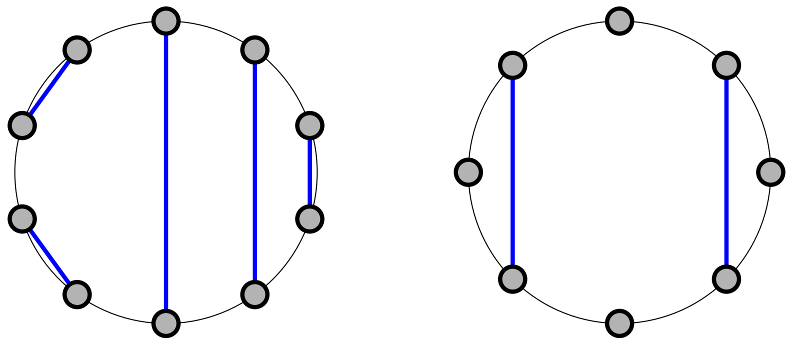


Figure 1.3: On the left: An element in $\mathcal{D}_5$. On the right: An element in $\mathcal{M}_8$.

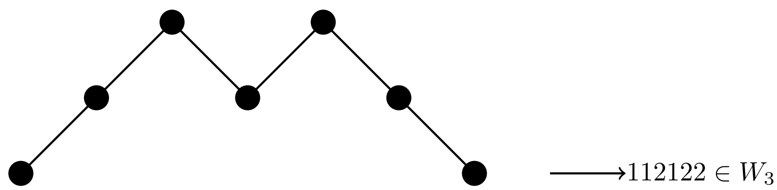


Figure 1.4: Turning a Dyck path into a word

replacing each up step with a 1 and each down step with a 2. We can realize a Motzkin path as a word with letters 1, 2, and 3 similarly to that of a Dyck path, but we replace a horizontal step with a 3 (see Figures 1.4, 1.5). We will use W_n to denote the set of Dyck paths (as words) of length $2n$, and $\mathcal{P}_n$ to denote the set of Motzkin paths (as words) of length n.

Therefore, we can say W_n consists of words $w = w_1...w_{2n}$ using exactly n 1s and n 2s such that for each i, $\#\{j : w_j = 1, j \leq i\} \geq \#\{j : w_j = 2, j \leq i\}$.

We can also say that $\mathcal{P}_n$ are the set of words $w = w_1...w_n$ on letters $1, 2, 3$ such that the number of 1s and 2s are equal with moreover, the restriction to the subword containing no 3s is contained in W_k for some k.

We denote the set of perfect matchings on $2n$ as $\mathcal{D}_n$ so that $|\mathcal{D}_n| = C_n$. The action of $\langle g \rangle$ on $\mathcal{D}_n$ mentioned in section 1 is defined by the action of the generator, g, which is to rotate the matching clockwise by $\frac{2\pi}{2n} = \frac{\pi}{n}$. So, the CSP in section 1 shows that the number of perfect matchings on $2n$ fixed by rotating the diagram by $\frac{2\pi \cdot d}{2n}$ clockwise is $C_n(\zeta_{2n}^d)$.

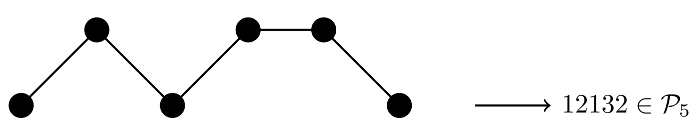


Figure 1.5: Turning a Motzkin path into a word

1.3 The first Motzkin CSP triple

First we begin by finding a CSP for Motzkin numbers using the canonical $q-$analogue. The other CSPs involving Motzkin numbers we find will build off of this one. Recall the canonical $q-$analogue 1.2 from Section 1.1 as

$$M_n(q) = \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{1}{[k+1]_q} \begin{bmatrix} n \\ 2k \end{bmatrix}_q \begin{bmatrix} 2k \\ k \end{bmatrix}_q . \tag{1.3}$$

At first it seems simple to show this expression exhibits CSP noting that the canonical $q-$analogue of the Catalan numbers $C_k(q)$ and the $q-$binomial coefficients $\begin{bmatrix} n \\ 2k \end{bmatrix}_q$ are both polynomials in a CSP triple ([12], [13]). However, the obstacle is that the two CSPs have actions of different orders ($2k$ and n respectively), and so when substituting q for an nth root of unity ζ, it is not immediately clear that ζ is also a $2k$th root of unity, let alone the "correct" one.

Let $T = \{a_1, ..., a_k\} \subseteq [n]$ where $a_1 < ... < a_k$, and let the permutation $g := (1\ 2\ ...\ n) \in S_n$ written in cycle notation act naturally on the subsets of $[n]$. Suppose $g^d \cdot T = T$ and $d|n$. Then $\langle g^d \rangle$, the cyclic group with generator g^d also has an induced action $\varphi : \langle g^d \rangle \to S_T$ where S_T is the group of permutations of T. Intuitively, the nature of this action should be cyclic. That is, we should expect $\varphi(g^d) = (a_1\ a_2\ ...\ a_k)^r$ for some r. The following proposition confirms this.

Proposition 1.3.1. The map φ satisfies the following properties

(i) $\varphi(\langle g^d \rangle) \subseteq \langle (a_1\ a_2\ ...\ a_k) \rangle$

(ii) The map φ is an injective homomorphism

(iii) $\varphi(g^d) = (a_1\ a_2\ ...\ a_k)^r$ where r is the number of distinct residues of T mod d

(iv) $\frac{k}{r} = \frac{n}{d}$

Proof. Note we can identify $[n]$ with the set of residue classes mod n. Also, we see that each $g^{dt} : x \mapsto x + dt \pmod{n}$.

Let $0 \leq \mu_1 < ... < \mu_r < d$ be representatives of the distinct resdiues of T modulo d. Because $(g^d)^t \cdot a_i \equiv a_i + td \pmod{n}$ and $d|n$ we have $(g^d)^t \cdot a_i \equiv a_i \pmod{d}$. Moreover, if $a_j \equiv a_i \pmod{d}$ write $a_j \equiv a_i + td \pmod{n}$. Then $(g^d)^{n-t} \cdot a_j \equiv a_j + (n-t)d \pmod{n} \equiv a_j - td \pmod{n} \equiv a_i \pmod{n}$ so $(g^d)^{n-t} \cdot a_j = a_i$. This implies that all the elements of T equivalent mod d lie in the same cycle in $\varphi(g^d)$. Therefore, we can write

$$\varphi(g^d) = (b_1^{(1)}\ b_2^{(1)}\ ...\ b_{s_1}^{(1)})(b_1^{(2)}\ b_2^{(2)}\ ...\ b_{s_2}^{(2)})...(b_1^{(r)}\ b_2^{(r)}\ ...\ b_{s_r}^{(r)})$$

Where each $\{b_1^{(i)}, ..., b_s^{(i)}\} = \{a_j \in T : a_j \equiv \mu_i \pmod{d}\}$ and each $b_{j+1}^{(i)} \equiv b_j^{(i)} + d \pmod{n}$. We also assume without loss of generality that $b_1^{(i)} < b_2^{(i)} < ... < b_s^{(i)}$. In particular, we have $b_{j+1}^{(i)} = b_j^{(i)} + d$. Then, it follows that $b_j^{(i)} = \mu_i + (j-1)d$. We also note that $\mu_i + (j-1)d \in T$ as long as it is at most n. So, $\mu_i + (j-1)d \in T \iff 1 \leq \mu_i + (j-1)d \leq n$. Then $s_i - 1$ is the maximum integer such that $\mu_i + s_i d \leq n$. We conclude that $s_1 = s_2 = ... = s_r = \frac{n}{d}$. Then write $s = \frac{n}{d}$ and note that

$$(b_1^{(1)}\ b_1^{(2)}\ ...\ b_1^{(r)}\ b_2^{(1)}\ ...\ b_2^{(r)}\ ...\ b_s^{(1)}\ b_s^{(2)}\ ...\ b_s^{(r)})^r = \varphi(g^d)$$

We want to show that the order in which the $b_j^{(i)}$ are presented above is increasing. Note, $b_j^{(i)} = \mu_i + (j-1)d < \mu_{i+1} + (j-1)d = b_j^{(i+1)}$ also $0 \le \mu_1 < \mu_r < d \implies \mu_r < \mu_1 + d \implies \mu_r + (j-1)d < \mu_1 + jd$ so $b_j^{(r)} = \mu_r + (j-1)d < \mu_1 + jd = b_{j+1}^{(1)}$. Because $\{b_i^{(j)} : 1 \le i \le s, 1 \le j \le r\} = T$, it follows that

$$(a_1\ a_2\ ...\ a_k) = (b_1^{(1)}\ b_1^{(2)}\ ...\ b_1^{(r)}\ b_2^{(1)}\ ...\ b_2^{(r)}\ ...\ b_s^{(1)}\ b_s^{(2)}\ ...\ b_s^{(r)})$$

and hence $\varphi(g^d) = (a_1\ a_2\ ...\ a_k)^r$. We note that $\varphi(g^d) = (a_1\ a_2\ ...\ a_k) \in \langle a_1\ a_2\ ...\ a_k\rangle$ and hence φ is a homomorphism into $\langle a_1\ a_2\ ...\ a_k\rangle$. Also, because $s = \frac{n}{d}$ and $\varphi(g^d)$ is the product of disjoint cycles of length s the order of $\varphi(g^d)$ is $\frac{n}{d}$. Note that this implies that $\varphi(g^{td}) = 1$ if and only if t is a multiple of $\frac{n}{d}$. In particular, if and only if $g^{td} = 1$. Hence, φ is injective. Finally, $\varphi(g^d) = (a_1\ a_2\ ...\ a_k)^r$ where $\frac{rn}{d} = k \implies \frac{n}{d} = \frac{k}{r}$. □

The description of a non-crossing matching given in Section 1.1 is rather informal, and we need a more precise definition to write mathematical arguments. We will do this using the language of graph theory.

For $X \subseteq \mathbb{N}$, let $\mathcal{M}_X$ be the graphs on X of degree ≤ 1 satisfying the condition that whenever $x < y < z < w$ and x is adjacent to z we have y not adjacent to w. Each vertex having degree at most 1 forces X to be a matching. The other condition gives a "embedding free" description of what it means for the matching to be non-crossing.

We further define $\mathcal{D}_X \subseteq \mathcal{M}_X$ to be the graphs in $\mathcal{M}_X$ where each vertex has degree 1. In other words, $\mathcal{D}_X$ requires the matchings to be perfect. Note that $\mathcal{M}_n := \mathcal{M}_{[n]}$ are exactly the non-crossing matchings on $[n]$, of which there are M_n of, and $\mathcal{D}_n := \mathcal{D}_{[2n]}$ are exactly the perfect non-crossing matchings on $[2n]$, of which there are exactly C_n of.

We can define a $\langle g\rangle$ action on $([n], E) \in \mathcal{M}_n$ by $g \cdot ([n], E) = ([n], \{(x+1)(y+1) : xy \in E\})$ where $x+1, y+1$ are computed mod n. Note that this action corresponds to clockwise rotation by $\frac{2\pi}{n}$ if we embed the vertices $[n]$ on a circle increasing in a clockwise manner. In particular, it agrees with the action defined in Section 1.1. Using Proposition 1.3.1(iii), one can understand how the structure of the degree 1 vertices "rotates" when rotating a non-crossing matching in such a way that all the isolated vertices remain fixed.

Let $([n], E) \in \mathcal{M}_n$ and let $T = \{u \in [n] : \deg(u) = 1\}$. Write $T = \{a_1, ..., a_{2k}\}$ with $a_1 < ... < a_{2k}$ and suppose $g^d \cdot T = T$. Proposition 1.3.1 implies that the edge set E' of $g \cdot ([n], E) = ([n], E')$ is $\{a_{i+r}a_{j+r} : a_ia_j \in E\}$ where $i+r, j+r$ are calculated mod $2k$ and r is the number of distinct residues of T mod d (see Figure 1.6). Note that $([2k], \{ij : a_ia_j \in E\}) \in \mathcal{D}_{[2k]}$ and that $g^d \cdot ([n], E) = ([n], E)$ if and only if $(1\ 2\ ...\ 2k)^r \cdot ([2k], \{ij : a_ia_j \in E\}) = ([2k], \{ij : a_ia_j \in E\})$. Hence, if $g^d \cdot T = T$ then the number of non-crossing matchings whose degree 1 vertices are exactly T is the number of perfect non-crossing matchings on $|T| = 2k$ vertices fixed by $(1\ 2\ ...\ 2k)^r$. Proposition 1.3.1(iv) implies that $\zeta_{2k}^r = \zeta_n^d$ so the number of perfect non-crossing matchings fixed by $(1\ 2\ ...\ 2k)^r$ is

$$C_k(\zeta_{2k}^r) = C_k(\zeta_n^d) = \frac{1}{[k+1]_{\zeta_n^d}} \begin{bmatrix} 2k \\ k \end{bmatrix}_{\zeta_n^d}.$$

By [12, Theorem 1.1b], there are $\begin{bmatrix} n \\ 2k \end{bmatrix}_{\zeta_n^d}$ subsets $T \subseteq [n]$ of size $2k$ which satisfy $g^d \cdot T = T$. Therefore, the number of graphs $([n], E) \in \mathcal{M}_n$ fixed by g^d and have $2k$ vertices of degree 1 is

$$\frac{1}{[k+1]_{\zeta_n^d}} \begin{bmatrix} n \\ 2k \end{bmatrix}_{\zeta_n^d} \begin{bmatrix} 2k \\ k \end{bmatrix}_{\zeta_n^d}.$$

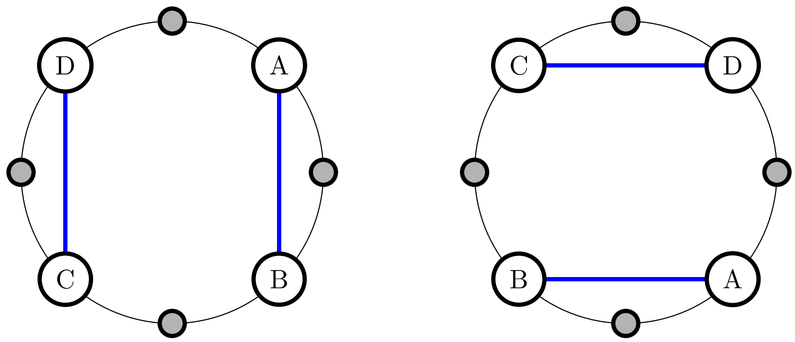


Figure 1.6: In our definitions, the "labels" $1, 2, ..., 8$ of the vertices only indicate where on the circle they are placed. Imagine we place additional colours A, B, C, D on the vertices which have degree one. If we rotate the diagram on the left by $\frac{\pi}{4}$ clockwise, we obtain the diagram on the right. The set of locations the coloured vertices occupy remains unchanged. However, by distinguishing the colours A, B, C, D we are able to see how the coloured vertices were "actually" rotated by one place clockwise. The r in Proposition 1.3.1 tells us exactly the amount of this "actual" rotation. This in turn tells us how the set of degree one vertices rotated if they formed their own perfect matching on 4 vertices.

The number of degree 1 vertices in a graph $([n], E) \in \mathcal{M}_n$ can be any even number from 0 to n. Hence, summing the above expression over all even numbers between 0 and n we obtain that the number of graphs $([n], E) \in \mathcal{M}_n$ fixed by g^d is

$$\sum_{k=0}^{\lfloor n/2 \rfloor} \frac{1}{[k+1]_{\zeta_n^d}} \begin{bmatrix} n \\ 2k \end{bmatrix}_{\zeta_n^d} \begin{bmatrix} 2k \\ k \end{bmatrix}_{\zeta_n^d}.$$

We arrive at the conclusion that

Theorem 1.3.2. The following triple exhibits CSP

$$\left(\mathcal{M}_n, \langle g \rangle, \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{1}{[k+1]_q} \begin{bmatrix} n \\ 2k \end{bmatrix}_q \begin{bmatrix} 2k \\ k \end{bmatrix}_q \right) = \left(\mathcal{M}_n, \langle g \rangle, M_n(q) \right).$$

1.4 Statistic for the $q-$analogue generating function

The major index statistic on words whose letters lie in any totally ordered set is $\text{maj}(w) = \sum_{w_i > w_{i+1}} i$ where $w = w_1 ... w_n$. Recall the interpretation of Dyck paths as words, W_n in Section 1.2. It is well known that the canonical $q-$analogue of the Catalan numbers has the generating series interpretation

$$\frac{1}{[n+1]_q} \begin{bmatrix} 2n \\ n \end{bmatrix}_q = \sum_{w \in W_n} q^{\text{maj}(w)}.$$

In particular, we can rephrase the CSP triple $(\mathcal{D}_n, \langle g\rangle, C_n(q))$ as

$$\left(W_n, \langle g\rangle, \sum_{w\in W_n} q^{\mathrm{maj}(w)}\right)$$

Being able to write the polynomial as a generating function for the finite set is desirable because it lets us interpret evaluation of the polynomial at roots of unity as each object having its own "contribution" to the output with each contribution being determined by the statistic. All objects are "given" a point on the unit circle, whose angle to the real line is a multiple of $\frac{2\pi}{n}$. The statistic then tells each object how much to rotate this point around the origin. The sum of all resulting points on the circle yields the number of objects fixed by the cyclic action.

It is also known [12] that the coefficients of a polynomial in a CSP triple admit a combinatorial description. In particular, if $f(q) = \sum_i a_i q^i$ then for each $i < n$, $\sum_{j\geq 0} a_{i+nj}$ is the number of orbits whose order divides i.

By construction, generating series coefficients are of combinatorial significance— they count the number of elements with the same weight. As such, we can compare these two combinatorial descriptions of the coefficients.

Example 1.4.1. Major index can be defined on perfect non-crossing matchings through the canonical bijection between Dyck paths and perfect non-crossing matchings. It yields the description of being the sum of all vertices less than $2n-1$ which is a clockwise "end point", and whose immediate clockwise neighbour is a clockwise "start point" (see Figure 1.7). Fix $i < 2n$. The number of perfect non-crossing matchings whose major index is of the form $i + 2nj$ for some $j \geq 0$ is the number of orbits of the cyclic action whose order divides i. In the case of perfect non-crossing matchings, it can also be shown that for a perfect non-crossing matching $X \in \mathcal{D}_n$, we have $\mathrm{maj}(g\cdot X) \equiv \mathrm{maj}(X) + k \pmod{n}$ where the k are the number of such vertices (which only depends on the orbit). This highlights another typical relationship between the statistic and action, although this does not happen in general. When a CSP triple has a polynomial defined in terms of a statistic satisfying the above relationship we say the CSP instance is *local*.

The canonical $q-$analogue $M_n(q)$ has a nice form (just by virtue of it being built from the standard $q-$analogue building blocks), but can it be realized as the generating function of some objects enumerated by Motzkin numbers similar to how there is a generating series interpretation for the $q-$analogue of Catalan numbers?

Although it is not strictly necessary, we forgo the traditional definition of a word and its content. This doesn't change the math in an essential way and also allows us to minimally change the relative order among letters. For example, we use a "3" to represent a horizontal step in a Motzkin path. This was only done because "1" and "2" already represent an up and down step in Dyck paths and we want the words corresponding to Motzkin paths to be a natural extension of words corresponding to Dyck paths (which we really want to be Yamanouchi words). However, one may intuitively see that it may make more sense to put horizontal steps in between an up and down step in terms of numbering, but this cannot be done if we only allow natural numbers. In fact, we will later consider what happens if we replace our representation of horizontal steps with some other number.

Our new definition of a word allows the letters to be any real number, and as such, we define the content of a word $w = w_1...w_n$ to be a function $\alpha : \mathbb{R} \to \mathbb{N}$ where $\alpha(r) = \#\{i : w_i = r\}$. The support of α is $\mathrm{supp}(\alpha) := \{r : \alpha(r) \neq 0\}$. We label the elements in the support of α by $\alpha_1, \alpha_2, ...,$ where $\alpha_1 < \alpha_2 < ....$

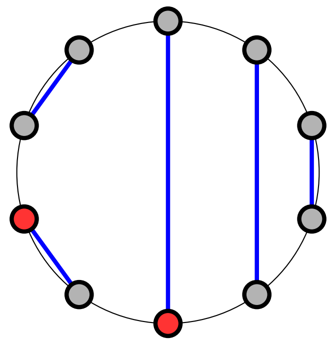


Figure 1.7: The vertices less than 10 which are clockwise endpoints, and whose immediate clockwise neighbours are clockwise start points are coloured in red. There are two of them whose vertices have labels $6, 8$. Therefore, the major index of this perfect non-crossing matching is $6 + 8 = 14 \equiv 4 \pmod{10}$. This also shows that modulo 10 the major index increases by 2 for each clockwise rotation by $\frac{2\pi}{10}$.

Definition 1.4.2. Define a *word poset* of content $\alpha : \mathbb{R} \to \mathbb{N}$ to be the disjoint union of chains $\sum_{\alpha_i} P_{\alpha_i}$ where α_i ranges over the (finite) support of α.

Let P be a poset and $\omega : P \to [n]$ a bijection. Following [19] we make the following definitions

Definition 1.4.3. A *linear extension* f is a bijection $f : P \to [n]$ satisfying the condition that for all $a, b \in P$ with $a < b$ we have $f(a) < f(b)$. The set of linear extensions of P is written as $\mathcal{L}(P)$.

Definition 1.4.4. The *Jordan–Hölder set* $\mathcal{L}(P, \omega)$ is the set of linear extensions of P with respect to ω. That is, $\mathcal{L}(P, \omega)$ is the set of permutations $\sigma_1 ... \sigma_{|P|} \in S_{|P|}$ such that for each $i < j$, we have $\omega^{-1}(\sigma_i) < \omega^{-1}(\sigma_j)$ in P

We will occasionally identify permutations $\omega \in \mathcal{L}(P, \omega)$ with their corresponding linear extension. For a word poset $P = \sum_{i=1}^{\ell} P_{\alpha_i}$, with each P_{α_i} being the chain $p_1^{(\alpha_i)} < p_2^{(\alpha_i)} < ... < p_{|P_{\alpha_i}|}^{(\alpha_i)}$ we have a natural labelling $\omega : P \to \mathbb{N}$ of P in which the inverse ω^{-1} is given by listing the elements of P_{α_1} in order, then the elements of P_{α_2}, etc. Note that for each $\sigma \in \mathcal{L}(P, \omega)$ we can uniquely define a word $\gamma_P(\sigma) := \gamma_1 \gamma_2 ... \gamma_n \in \mathbb{R}^n$ of content α by letting $\gamma_i = t$ if and only if $\sigma^{-1}(i) \in P_t$ viewing σ as a linear extension. When P is clear, we will omit the subscript.

Example 1.4.5. Consider the content $\alpha : \mathbb{R} \to \mathbb{N}$ where $\alpha(1) = 3$, $\alpha(2.4) = 3$ and $\alpha(3) = 1$. Then the word poset P of content α consists of three chains $P_1, P_{2.4}, P_3$ of lengths 3, 3, and 1 respectively. The word $w = (1)(1)(2.4)(1)(3)(2.4)(2.4)$ has content α. Writing P_1 as $p_1^{(1)} < p_2^{(1)} < p_3^{(1)}$, $P_{2.4}$ as $p_1^{(2.4)} < p_2^{(2.4)} < p_3^{(2.4)}$ and P_3 as $p_1^{(3)}$ the natural labelling ω is

$$p_1^{(1)} p_2^{(1)} p_3^{(1)} p_1^{(2.4)} p_2^{(2.4)} p_3^{(2.4)} p_1^{(3)}$$

Where the place at which element p occurs reading left to right is the value $\omega(p)$. For example, $\omega(p_2^{(2.4)}) = 5$ because $p_2^{(2.4)}$ is the fifth element of the string reading left to right. The function ω

itself is a linear extension. The function $\overline{\sigma}$ given by

$$p_1^{(1)}p_1^{(2.4)}p_2^{(2.4)}p_1^{(3)}p_2^{(1)}p_3^{(2.4)}p_3^{(1)}$$

is also a linear extension where again the place at which an element p occurs is the value of $\overline{\sigma}$ is also a linear extension. The corresponding permutations of ω and $\overline{\sigma}$ are $1234567, 1457263 \in \mathcal{L}(P,\omega)$ respectively. We have $\gamma(1234567) = (1)(1)(1)(2.4)(2.4)(2.4)(3)$ and also $\gamma(1457263) = (1)(2.4)(2.4)(3)(1)(2.4)(1)$. Note that when giving linear extensions in the forms written for $\omega, \overline{\sigma}$ that applying γ to the corresponding permuation is the same as reading off the superscripts left to right.

Lemma 1.4.6. The map γ from $\mathcal{L}(P,\omega)$ to words of content α is a bijection.

Proof. It suffices to provide an inverse map to γ. For a word $w = w_1w_2...w_n$ of content α define $\rho(w) : P \to [n]$ as follows. If $n = 1$ then P consists of one chain with exactly one element. Define $\rho(p_1^{(t)}) = 1$ where $P = P_t$. Suppose $\rho(w)^{-1}(i)$ has been defined for $i < k$. Define $\rho(w)^{-1}(k)$ to be the smallest element of P_{w_k} which does not belong to $\{\rho(w)^{-1}(1), ..., \rho(w)^{-1}(k-1)\}$. Such an element will always exist because w is of content α. Note that by construction $\rho(w)$ is a linear extension of P with respect to ω. Then, γ and ρ are mutually inverse bijections which completes the proof. □

Example 1.4.7. Consider w from the previous example. We have that $\gamma^{-1}(w)$ is the permutation $1243756 \in \mathcal{L}(P,\omega)$ and linear extension given by

$$p_1^{(1)}p_2^{(1)}p_1^{(2.4)}p_3^{(1)}p_1^{(3)}p_2^{(2.4)}p_3^{(2.4)}$$

Remark 1.4.8. The map γ can be thought of as the "Yamanouchi" word of the word poset. We postpone the discussion of tableaux at the moment to avoid the conflicting definitions of major index. See 2.2 for background on tableaux and Yamanouchi words

Lemma 1.4.9. For $\sigma \in \mathcal{L}(P,\omega)$, $\mathrm{maj}(\sigma) = \mathrm{maj}(\gamma(\sigma))$.

Proof. Note that $\gamma(\sigma) = \gamma_1...\gamma_n$ and $\sigma = \sigma_1...\sigma_n$ have the same length. So we show that $\sigma_i\sigma_{i+1}$ is a descent if and only if w_iw_{i+1} is a descent. We note that σ corresponds to some linear extension $\overline{\sigma} : P \to \mathbb{N}$ given by $\overline{\sigma}^{-1}(i) = \omega^{-1}(\sigma_i)$ so that $\sigma_1...\sigma_n = \omega(\overline{\sigma}^{-1}(1))...\omega(\overline{\sigma}^{-1}(n))$. Suppose $\sigma_i > \sigma_{i+1}$. That is, $\omega(\overline{\sigma}^{-1}(i)) > \omega(\overline{\sigma}^{-1}(i+1))$. Because $\overline{\sigma}$ is a linear extension and $i < i+1$, it follows that either $\overline{\sigma}^{-1}(i)$ is incomparable to $\overline{\sigma}^{-1}(i+1)$ or $\overline{\sigma}^{-1}(i) < \overline{\sigma}^{-1}(i+1)$. Because ω itself is a linear extension, the latter case cannot occur. Thus, $\overline{\sigma}^{-1}(i)$ is incomparable to $\overline{\sigma}^{-1}(i+1)$. Let $\overline{\sigma}^{-1}(i) \in P_a$ and $\overline{\sigma}^{-1}(i+1) \in P_b$. It follows that $a \neq b$. Moreover because ω lists the chains P_t in increasing order and $\omega(\overline{\sigma}^{-1}(i)) > \omega(\overline{\sigma}^{-1}(i+1))$ we must in fact have $a > b$. By construction of $\gamma(\sigma)$, we have $\gamma_i > \gamma_{i+1}$. Conversely, if $\gamma_i > \gamma_{i+1}$, then $\overline{\sigma}^{-1}(i) \in P_a$ and $\overline{\sigma}^{-1}(i+1) \in P_b$ with $a > b$. Then, as ω is a linear extension we have $\sigma_i = \omega(\overline{\sigma}^{-1}(i)) > \omega(\overline{\sigma}^{-1}(i+1)) = \sigma_{i+1}$. □

Example 1.4.10. Again consider $w = (1)(1)(2.4)(1)(3)(2.4)(2.4)$ from example 4.1 which has two descents that occur at indices 3 and 5, yielding a major index of 8. From example 4.2, the corresponding permutation is 1243756 which also has two descents occuring at indices 3 and 5 resulting in the same major index of 8.

We will also need the concept of "partial" linear extensions

Definition 1.4.11. A *partial linear extension* f is an injective map $f : P_{|Q} \to [n]$ for some $Q \subseteq P$ satisfying $a, b \in P$ with $a < b$ implies $f(a) < f(b)$

Definition 1.4.12. The *generalized Jordan–Hölder set*, $\mathcal{L}^+(P, \omega)$ is the set of partial linear extensions with respect to ω. That is, $\mathcal{L}(P, \omega)$ is the set of words without repeating letters $\sigma_1 ... \sigma_k$ such that for each $i < j$ we have $\omega^{-1}(\sigma_i) < \omega^{-1}(\sigma_j)$ in P.

Note the map γ_P extends to elements in $\mathcal{L}^+(P, \omega)$, which we will denote γ_P^+. As with γ_P^+, we omit the subscript when the poset in question P is clear from context.

Definition 1.4.13. Let P be a poset and Q be an induced poset. Let $\omega : P \to [|P|], \theta : Q \to [|Q|]$ be labellings of P, Q respectively. We say that θ is an *induced labelling* of ω if for any $q_1, q_2 \in Q$, $\omega(q_1) < \omega(q_2) \iff \theta(q_1) < \theta(q_2)$.

An important example of induced labellings is are certain subword posets of a word poset.

Example 1.4.14. Let P be a word poset of content $\alpha + \beta$ where α, β have disjoint support. Let Q be the word poset of content α so that Q is an induced poset of P. The natural labelling of the word poset on Q is an induced labelling of the natural labelling of P.

As a Corollary of Definition 1.4.13,

Corollary 1.4.15. Let w be a word of content α. Let β have disjoint support from α and consider the word poset P of content $\alpha + \beta$ and the word poset Q of content α. Then,

$$\mathrm{maj}(\gamma_Q(w)) = \mathrm{maj}(\gamma_P^+(w))$$

These concepts allow us to generalize [19, Exercise 162(b)] to words with arbitrary disjoint content.

Theorem 1.4.16. Let $w = w_1 ... w_n$ and $u = u_1 ... u_k$ be two words such with disjoint support. Let V be the set of words $v_1 ... v_{n+k}$ that contain both w and u as a subword. Then,

$$\sum_{v \in V} q^{\mathrm{maj}(v)} = q^{\mathrm{maj}(w) + \mathrm{maj}(u)} \begin{bmatrix} n+k \\ k \end{bmatrix}_q$$

Proof. Let λ, μ be the contents of w, u respectively, and $\alpha = \lambda + \mu$. Let R, S, P be the word posets of content λ, μ, α respectively. Write $\sigma = {\gamma_P^+}^{-1}(w), \pi = {\gamma_P^+}^{-1}(u)$.

Let W be the set of permutations in $S_{|\lambda| + |\mu|}$ which contain both σ, π as subwords. We observe that $W \subseteq \mathcal{L}(P, \omega)$ because R, S have disjoint support. Moreover, we note that $\gamma_{P|W} : W \to V$ is a bijection. Indeed, note it is injective by Lemma 1.4.6 and we can also see that it is surjective by restricting the map in Lemma 1.4.6. This bijection is moreover weight preserving by Lemma 1.4.9. The result then follows by [19, Exercise 162(b)] noting that $\mathrm{maj}(\sigma) = \mathrm{maj}(w), \mathrm{maj}(\pi) = \mathrm{maj}(v)$ by Corollary 1.4.15 and Lemma 1.4.9. □

The above theorem in its stated form follows easily from [19, Exercise 162(b)]. However, the given proof is more flexible in the sense that it implies the result for alternate definitions of major index which we will see later. This can be accomplished by changing the labelling ω on a word poset. In fact, Theorem 1.4.16 is true for any labelling ω on words posets which satisfy Corollary 1.4.15 and Lemma 1.4.9.

As a Corollary of Theorem 1.4.16 we obtain the well known result,

Corollary 1.4.17. Let V be the set of binary words $v_1 ... v_n$ of which exactly k letters are 1 and $n-k$ letters are 0. Then,

$$\sum_{v \in V} q^{\mathrm{maj}(v)} = \begin{bmatrix} n \\ k \end{bmatrix}_q$$

Theorem 1.4.16 along with Corollary 1.4.17 also yields a generating series interpretation for the polynomial in Theorem 1.3.2 or 1.2.

Theorem 1.4.18.

$$\sum_{w \in \mathcal{P}_n} q^{\mathrm{maj}(w)} = \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{1}{[k+1]_q} \begin{bmatrix} n \\ 2k \end{bmatrix}_q \begin{bmatrix} 2k \\ k \end{bmatrix}_q$$

Proof. Following Section 1.2, let W_n be the set of Dyck paths, realized as words in $\{1,2\}^n$ with an up step being represented by a 1 and a down step being represented by a 2. Also recall from section 1 that the set of Motzkin paths $\mathcal{P}_n$ are realized as words in $[1,2,3]^n$. A string of 3s has major index 0. So, for each Dyck path u of length $2\ell \leq n$, we note that Theorem 1.4.16 implies that

$$\sum_{v \in V} q^{\mathrm{maj}(v)} = q^{\mathrm{maj}(u)+0} \begin{bmatrix} n \\ 2\ell \end{bmatrix}_q = q^{\mathrm{maj}(u)} \begin{bmatrix} n \\ 2\ell \end{bmatrix}_q$$

Where V is the set of Motzkin paths whose substring obtained by restricting to the letters 1 and 2 yield u. The result then follows by summing over all possible u. □

In fact, the relative order of the horizontal steps compared to the up and down steps does not matter. That is, if we were to redefine $\mathcal{P}_n$ so that horizontal steps are instead represented using another number which is neither 1 nor 2 such as 0 or 1.5 then notice the major index of each individual Motzkin path will change, but Theorem 1.4.18 will still hold with $M_n(q)$ on the right hand side.

1.4.1 Variations of the major index of Motzkin paths

This subsection uses standard tableaux. For the reader unfamiliar with standard tableaux see Section 2.2.

There are a few more variations of major index on Motzkin paths we will discuss. The first, is similar to before but we use "flipped" major index denoted $\mathrm{fmaj}(w) := \sum_{w_i < w_{i+1}} i$ the difference being that the indices of the ascents are summed rather than the indices of the descents. If T is a standard tableau, its major index is the sum of indices i such that the cell containing i appears above the cell containing $i+1$. If we let w be the Yamanouchi word corresponding to T we then notice $\mathrm{maj}(T) = \mathrm{fmaj}(w)$, motivating the definition of $\mathrm{fmaj}(w)$. Dyck paths are in bijection with standard tableaux of shape nn, $\mathrm{SYT}(nn)$. The bijective correspondence between $\mathrm{SYT}(nn)$ and Dyck paths is given by taking Yamanouchi word.

It turns out that this bijection is also weight preserving if weighting Dyck paths by fmaj and standard tableaux by maj. Hence using the hook length formula [20, Corollary 7.21.5]

$$\sum_{w \in W_n} q^{\mathrm{fmaj}(w)} = \sum_{Y \in \mathrm{SYT}(nn)} q^{\mathrm{maj}(T)} = \frac{q^n [2n]_q!}{[n]_q! [n+1]_q!} = \frac{q^n [2n]_q!}{[n]_q!^2 [n+1]_q} \tag{1.4}$$

$$= \frac{q^n}{[n+1]_q} \begin{bmatrix} 2n \\ n \end{bmatrix}_q \tag{1.5}$$

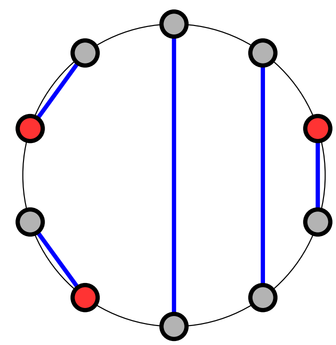


Figure 1.8: The vertices less than 10 which are the clockwise start of a shortest edge are coloured in red. There are three of them whose vertices have labels $3, 7, 9$. Therefore, the flipped major index of this perfect non-crossing matching is $3 + 7 + 9 = 19 \equiv 9 \pmod{10}$. This also shows that modulo 10 the flipped major index increases by 3 for each clockwise rotation by $\frac{2\pi}{10}$.

Example 1.4.19. Using fmaj instead of maj is also more natural on perfect non-crossing matchings (see Example 1.4.1). Indeed, now the major index is just the sum of the clockwise start of all the shortest edges (excluding potentially vertex $2n$), and the major index increase modulo $2n$ is given by the number of shortest edges (see Figure 1.8).

By modifying the natural labelling ω of a word poset to start at the chains corresponding to larger letters first, see that Theorem 1.4.16 is still true if "maj" is replaced by "fmaj". This implies that

$$\sum_{w \in \mathcal{P}_n} q^{\mathrm{fmaj}(w)} = \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{q^k}{[k+1]_q} \begin{bmatrix} n \\ 2k \end{bmatrix}_q \begin{bmatrix} 2k \\ k \end{bmatrix}_q \tag{1.6}$$

Unfortunately, the right hand side of 1.6 does not agree with $M_n(q)$ when evaluated at nth roots of unity. The problem is that the additional power of q in each summand is only raised to the kth power rather than $2k$th power. One way to fix this problem is to consider a "translated flipped major index" $\mathrm{tfmaj}(w) := \mathrm{fmaj}(w) + \#\{i : w_i = 1\}$ to get

$$\sum_{w \in \mathcal{P}_n} q^{\mathrm{tfmaj}(w)} = \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{q^{2k}}{[k+1]_q} \begin{bmatrix} n \\ 2k \end{bmatrix}_q \begin{bmatrix} 2k \\ k \end{bmatrix}_q . \tag{1.7}$$

We note that the expression of each summand (in 1.6, 1.7) upon substituting $q \mapsto \zeta$ where ζ is a dth root of unity (with $d|n$) will only be non-zero when $d|2k$ (this can be seen algebraically, or by [12, Theorem 1.1b]). In particular, when substituting ζ if the summand is non-zero it is equal to the equivalent summand in $M_n(\zeta)$ because $\zeta^{2k} = 1$. When $d \not| 2k$ the summand in 1.6 and $M_n(\zeta)$ are both zero. This shows that that 1.7 can replace $M_n(q)$ in Theorem 1.3.2.

However, this is a very crude fix as the additional factor that tfmaj has is unnatural. We can do better while keeping the spirit of flipped major index as long as we are willing to sacrifice some refinement in our cyclic action.

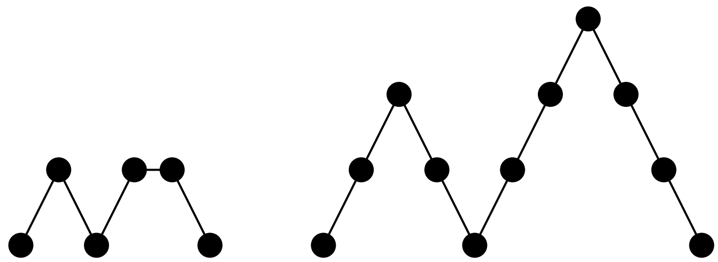


Figure 1.9: On the left: a Motzkin path in $\mathcal{P}_5$. On the right: the corresponding doubled Motzkin path in $\mathcal{D}_5$.

Definition 1.4.20. Given a Motzkin path w define the doubled Motzkin path $d(w)$ as follows. Replace each instance of a 1 with 11, each instance of 2 with 22, and each instance of 3 by 12.

Example 1.4.21. $d(12132) = 1122111222$. See Figure 1.9 for a visual description.

Let $\mathcal{P}_{2n+k}(k)$ be the Motzkin paths in $\mathcal{P}_{2n+k}$ with exactly k horizontal steps.

Lemma 1.4.22. We have the generating series equivalence

$$q^k \sum_{T \in \mathrm{SYT}((n+1)^2 1^k/11)} q^{2\mathrm{maj}(T)} = \sum_{w \in \mathcal{P}_{2n+k}(k)} q^{\mathrm{fmaj}(d(w))}$$

Proof. Define a weight preserving bijection from LHS to RHS, φ as follows. For a tableaux $T \in \mathrm{SYT}((n+1)^2 1^k/1^2)$ consider the Yamanouchi word $u = u_1 u_2 ... u_{2n+k}$ of T. Then, construct $\varphi(u) = d(w_1 ... w_{2n+k})$ where $w_i = u_i$ if $u_i \in \{1, 2\}$ and $w_i = 3$ if $u_i > 2$. The weight of T is $2\mathrm{maj}(T) + k = 2\mathrm{fmaj}(u_1 u_2 ... u_{2n+k}) + k$. Write the ascents $\mathrm{asc}(u_1 u_2 ... u_{2n+k}) = A \cup B$ where A for each $i \in A$, $u_{i+1} > 2$. We note that the ascent set $\mathrm{asc}(d(w_1 ... w_{2n+k})) = \{2i+1 : i \in A\} \cup \{2i : i \in B\}$. Also, we note that A is in bijection with the letters greater than 2, which is in turn in bijection with the entries in a row greater than 2. That is, $|A| = k$. Hence, the major index

$$\begin{aligned} \mathrm{fmaj}(d(w_1 ... w_{2n+k})) &= \mathrm{fmaj}(\varphi(T)) \\ &= \sum_{i \in A}(2i+1) + \sum_{i \in B} 2i \\ &= |A| + 2\mathrm{fmaj}(u_1 u_2 ... u_{2n+k}) \\ &= k + 2\mathrm{fmaj}(u_1 u_2 ... u_{2n+k}) \end{aligned}$$

So the map φ is indeed weight preserving. It also has the inverse ψ where given a doubled motzkin word w, find the original Motzkin word w' which is a Yamanouchi word for a standard Young Tableau of shape $(n+1)^2 1^k/1^2$. □

Proposition 1.4.23. We have the generating series equivalence

$$\sum_{T \in \mathrm{SYT}((n+1)^2 1^k/11)} q^{\mathrm{maj}(T)} = q^{\frac{k(k-1)}{2}} \begin{bmatrix} 2n+k \\ k \end{bmatrix}_q \cdot \frac{q^n}{[n+1]} \begin{bmatrix} 2n \\ n \end{bmatrix}_q$$

Proof. Let $\lambda = (n+1)^2 1^k$. Consider the set of words V where each $v_1...v_{2n+k} \in V$ contains both $345...k$ and some $w \in W_n$ as a subword. Then,

$$\sum_{T \in \mathrm{SYT}(\lambda/11)} q^{\mathrm{maj}(T)} = \sum_{v \in V} q^{\mathrm{fmaj}(v)} = q^{\frac{k(k-1)}{2}} \begin{bmatrix} 2n+k \\ k \end{bmatrix}_q \cdot \frac{q^n}{[n+1]} \begin{bmatrix} 2n \\ n \end{bmatrix}_q$$

Where the last equality follows from Theorem 1.4.16 and equation 1.5 □

Corollary 1.4.24. The generating series for doubled Motzkin paths weighted by flipped major index is

$$\sum_{k=0}^{\lfloor n/2 \rfloor} q^{(n-2k)^2} \begin{bmatrix} n \\ 2k \end{bmatrix}_{q^2} \cdot \frac{q^{2k}}{[k+1]_{q^2}} \begin{bmatrix} 2k \\ k \end{bmatrix}_{q^2} \tag{1.8}$$

A similar analysis that was done for 1.7 can be used to show that substituting q for ζ_n^j in (4.3) agrees with $M_n(\zeta_n^{2j})$. Hence, we obtain another CSP involving Motzkin numbers. Except here, the polynomial is 1.8 and the action of the generator is rotation by $\frac{2\pi}{n}$ (rotating twice as much as the action given in the original CSP for Motzkin numbers found in Section 1.3).

1.5 Generalization

The setting in which the techniques of Section 1.2 were used is far more specific than it needs to be. As such, we describe how they can be generalized. Let $B_{n,k} := \{T \subseteq [n] : |T| = k\}$. Write $g := (1\ 2\ ...\ n)$. Recall from Section 1.3 that if $g^d \cdot T = T$ then we have a map $\varphi : \langle g^d \rangle \to S_T$.

Let X, Y be finite sets. Suppose that there is a cyclic group $\langle g_1 \rangle$ of order n acting on X and a cyclic group $\langle g_2 \rangle$ of order m acting on Y. Then we can define cyclic group $\langle g \rangle = \langle (1\ 2\ ...\ n+m-1) \rangle$ of order $n+m$ acting on $B_{n+m,n} \times X \times Y$. The action of the generator is given by $g \cdot (A, x, y) = (g \cdot A, g_1 \cdot x, y)$ if $n+m \in A$ and $g \cdot (A, x, y) = (g \cdot A, x, g_2 \cdot y)$ if $n+m \notin A$ where the action of g is the natural action of $(1\ 2\ ...\ n+m)$ on $B_{n,n+m}$.

Lemma 1.5.1. Let $w = (A, x, y) \in B_{n+m,n} \times X \times Y$. Suppose $g^d \cdot A = A$. Then, $g^d \cdot w = (A, g_1^r x, g_2^s y)$ where r is the number of distinct residues of A modulo d, and s is the number of distinct residues of $[n+m] \setminus A$ modulo d.

Proof. By the definition, we have $g^d \cdot w = (A, g_1^a x, g_2^b y)$ where a is the number of sets in $\{A, g \cdot A, ..., g^{d-1} \cdot A\}$ which contain 1 and b is the number of sets in $\{A, g \cdot A, ..., g^{d-1} \cdot A\}$ which don't contain 1. r is equal to the number of distinct residues of A mod d, which is equal to the size of the set $\#(\{1, 2, ..., d\} \cap A)$. We have the bijective correspondence $\{1, 2, ..., d\} \cap A \to \{X \in \{A, g \cdot A, ..., g^{d-1} \cdot A\} : 1 \in X\}$ via $j \mapsto g^{j-1} \cdot A$. So $a = r$. One similarly shows that $b = s$. □

Theorem 1.5.2. Suppose $(X, \langle g_1 \rangle, f), (Y, \langle g_2 \rangle, h)$ are CSP triples with g_1 of order n and g_2 of order m. Define the action of $\langle g \rangle$ on $B_{n+m,n} \times X \times Y$ of order $n+m$ as above. Then

$$\left(B_{n+m,n} \times X \times Y, \langle g \rangle, \begin{bmatrix} n+m \\ m \end{bmatrix}_q fh \right)$$

Is a CSP triple.

Proof. Write $F(q) := \begin{bmatrix} n+m \\ m \end{bmatrix}_q f(q)h(q)$. If $\begin{bmatrix} n+m \\ n \end{bmatrix}_{\zeta_{n+m}^d} = 0$, then $g^d \cdot A \neq A$ for any $A \subseteq [n+m]$ with $|A| = n$. In particular, no triple $w \in B_{n+m,n} \times X \times Y$ is fixed by the action of g^d. So, $F(\zeta_{n+m}^d) = 0$ is equal to the number of triples fixed by g^d. Now suppose that $\begin{bmatrix} n+m \\ n \end{bmatrix}_{\zeta_{n+m}^d} \neq 0$. To enumerate the triples (A, x, y) that satisfy $g^d \cdot (A, x, y) = (A, x, y)$, we first pick any $A \in B_{n+m,n}$ satisfying $g^d \cdot A = A$. Next, we pick any x satisfying $g_1^r \cdot x = x$ where r is the number of distinct residues of A modulo d, and any y satisfying $g_2^s \cdot y = y$ where s is the number of distinct residues of $[n+m] \setminus A$ modulo d. Using Proposition 1.3.1, $\frac{d}{n+m} = \frac{r}{n} = \frac{s}{m}$. Hence, the number of ways to pick the pair (x, y) is $f(\zeta_n^r)h(\zeta_m^s) = f(\zeta_{n+m}^d)h(\zeta_{n+m}^d)$. Therefore, the number of triples fixed by g^d is $F(\zeta_{n+m}^d)$ as required. □

Remark 1.5.3. The reader familiar with promotion on linear extensions of a poset will notice that the action of g on $B_{n+m,n} \times X \times Y$ behaves the same way (inverse) promotion does on the linear extensions of the disjoint union of posets $P_1 + P_2$.

In light of the above remark, Theorem 1.5.2 implies

Theorem 1.5.4. Suppose $(\mathcal{L}(P_1), \partial, f), (\mathcal{L}(P_2), \partial, h)$ exhibit CSP. Then, the following triple exhibits CSP.

$$\left(\mathcal{L}(P_1 + P_2), \partial, \begin{bmatrix} |P_1| + |P_2| \\ |P_1| \end{bmatrix}_q f(q)h(q) \right)$$

Chapter 2

Rigged Configurations

2.1 Introduction

Rigged configurations are combinatorial objects which parameterize solutions to a heuristic method in mathematical physics known as the "Bethe Ansatz". They were introduced by Kerov, Kirillov, and Reshetikhin in [6]. In [6] it is shown that rigged configurations are in bijective correspondence with standard young tableaux under an algorithm called the Kerov-Kirillov-Reshetikhin (KKR) bijection. Rigged configurations and the KKR bijection were further extended by Kirillov and Reshetikhin in [7] to be in correspondence with partition content semistandard young tableaux. If we drop the condition that the semistandard tableaux must have partition content, then we no longer have a bijection, but we have the remarkable result from [8] that two semistandard tableaux produce the same rigged configuration if and only if they lie in the same S_n orbit (where the S_n action is generated by crystal reflection operators).

If semistandard tableaux can be viewed as standard tableaux such that the cells belonging to certain intervals form horizontal strips, then the generalization of semistandard tableaux to Littlewood-Richardson (LR) tableaux is to replace horizontal strips with fixed tableaux of other shapes. In [8], Kirillov, Schilling, and Shimozono further extended the KKR bijection and rigged configurations from semistandard tableaux to LR tableaux.

The modern view of the KKR bijection is not between LR tableaux and rigged configurations, but rather (classically) highest weight elements of tensor products Kirillov-Reshetikhin (KR) crystals and rigged configurations. Crystals (or Kashiwara crystals, or crystal bases) were introduced by Kashiwara in [5]. They are combinatorial analogues of representations of quantum groups. It has been shown that there is a connection between classical algebraic combinatorial objects and crystals (see [2]; we will also discuss this in Section 2.5). In particular, semistandard tableaux can be interpreted as highest weight elements of tensor products of KR crystals. KR crystals are also examples of what are known as *affine crystals*— tensor products of which are equipped with an isomorphism called the *combinatorial* R (also known as the *combinatorial* R*-matrix*) which certifies that the tensor product of affine crystals is "commutative". It turns out that when viewing semistandard tableaux as highest weight elements in tensor products of KR crystals, the combinatorial R is the same operation as crystal reflection operators on tableaux. This leads to the modern interpretation of the results of [8] which is that the KKR "bijection" is invariant under combinatorial R.

Rigged configurations are admittedly complicated objects, and are less intuitive than their tableaux counterparts. However, there are a number of advantages which makes the use of rigged configurations more desirable in certain situations.

For example, the data structure of rigged configurations is sometimes more efficient than that of semistandard tableaux. In particular, it is more efficient to use rigged configurations to compute the Kostka-Foulkes polynomials (see [3]) which are generating series of semistandard tableaux weighted by charge. The advantage in using rigged configurations over semistandard tableaux in this case is because small perturbations of the riggings of a rigged configuration induce small, controlled perturbations of the charge of the corresponding semistandard tableaux. Thus, one effectively only needs to compute all possible configurations, rather than all rigged configurations. In contrast, no similar optimizations are available if computing the Kotska-Foulkes polynomials with semistandard tableaux.

Rigged configurations also model the behaviour of the coloured box-ball system which is a generalization of the single colour box-ball system introduced in [22] by Takahashi and Satsuma to model the Korteweg-de Vries (KdV) equation. To illustrate the connection, under time evolution a reverse Yamanouchi word will eventually be broken up into "solitons" whose sizes are the rows of the first partition of the corresponding rigged configuration. Hence, the asymptotic behavior of time evolution of the coloured box-ball system can be studied by considering the corresponding rigged configurations.

These examples suggest that the interpretation of natural rigged configuration structures on tableaux through the KKR bijection should yield interesting structures on tableaux. However, because of how difficult the KKR bijection is, most of these structures are not understood. In this chapter, we attempt to understand some of them.

First we will review some definitions in Section 2.2. This includes partitions, tableaux, and rigged configurations. Then we will recall the KKR bijection in Section 2.3. We will results on rigged configurations and the KKR bijection in Section 2.4 before moving to Section 2.5, where we review tableau combinatorics from a crystal perspective. In the short Section 2.6 we define a few key concepts follows before the two main sections: Section 2.7 and Section 2.8. In Section 2.7 we prove our most crucial results, and in Section 2.8 we give extensions and applications.

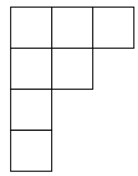
2.2 Definitions

We take $\mathbb{N}$ to include 0.

A partition λ of an integer $n \in \mathbb{N}$ is a non-increasing sequence of positive integers $(\lambda_1, ..., \lambda_\ell)$ such that $\sum_{i=1}^{\ell} \lambda_i = n$. In particular, there is exactly one partition of 0, which is the empty sequence.

We say that n is the size of λ. We can also write $|\lambda| = n$. The positive integers $\lambda_1, ..., \lambda_\ell$ are called the parts of the partition. ℓ is referred to as the length of the partition. Sometimes, it is convenient to interpret a partition as having an infinite number of trailing zeroes. In which case, we can define the length to be the number of positive parts. We can also write $\ell(\lambda)$ to refer to the length of a partition λ. If λ is a partition of n we write $\lambda \vdash n$. Partitions can be represented diagrammatically with *Young diagrams*. For a partition $\lambda = (\lambda_1, ..., \lambda_\ell) \vdash n$, we can draw ℓ rows with each of the $1 \leq i \leq \ell$ rows having λ_i boxes.

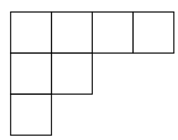
Example 2.2.1. We draw the Young diagram of the partition $(3, 2, 1, 1) \vdash 7$ below.



There are alternative ways of representing a partition that is more suitable for a text medium. The most common way to represent $\lambda = (\lambda_1, ..., \lambda_\ell)$ is to just concatenate the parts $\lambda_1\lambda_2...\lambda_\ell$. A more compact way of representing λ is to replace $\lambda_1...\lambda_x$ with ${\lambda_1}^x$ if $\lambda_1 = ... = \lambda_x$. In the above example, the partition $(3, 2, 1, 1)$ can also be written as 3211 or 321^2. The last useful interpretation of a partition is as a multiset consisting of its (positive). So, the partition $(3, 2, 1, 1)$ can be interpreted as a multiset $\{3, 2, 1, 1\}$.

We can rotate the Young diagram of a partition $\lambda = (\lambda_1, ..., \lambda_\ell)$ counter-clockwise by $\frac{\pi}{2}$, then reflect it across the $x-$axis. The partition corresponding to the resulting Young diagram is called the conjugate partition of λ and is written as $\overline{\lambda} = (\overline{\lambda}_1, ..., \overline{\lambda}_{\lambda_1})$. Note that the length of the conjugate partition is equal to the first row of the original partition. For this reason, we say that λ_1 is the width of the partition λ.

Example 2.2.2. The conjugate of the partition $(3, 2, 1, 1)$ is $(4, 2, 1)$ and is depicted below.

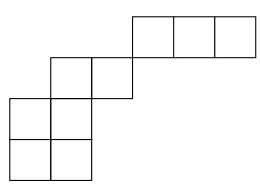


Example 2.2.3. A *composition* of n with length k is an ordered tuple of positive integers $\alpha = (\alpha_1, ..., \alpha_k)$ such that $\alpha_1 + ... + \alpha_k = n$. Let σ be a permutation such that $\alpha_{\sigma(1)} \geq ... \geq \alpha_{\sigma(k)}$. We then say that $(\alpha_{\sigma(1)}, ..., \alpha_{\sigma(k)})$ is the underlying partition of α and write $\alpha^+ := (\alpha_{\sigma(1)}, ..., \alpha_{\sigma(k)})$.

Example 2.2.4. For any infinite tuple $\alpha = (\alpha_1, \alpha_2, ...)$ with finite support and each $\alpha_i \in \mathbb{N}$ consider the set $S = \{\alpha_i : \alpha_i > 0, i \in \mathbb{N}\}$. The set S is finite. Write $S = \{\beta_1, \beta_2, ..., \beta_k\}$ where $\beta_1 \geq ... \geq \beta_k$. We call $(\beta_1, ..., \beta_k)$ the underlying partition of α.

Definition 2.2.5. A *skew partition* (or skew shape) of shape λ/μ is the Young diagram of λ, with the boxes from the Young diagram of μ removed.

Example 2.2.6. We draw a skew partition of shape 6322/32



The *size* of a skew shape $|\lambda/\mu| := |\lambda| - |\mu|$. So the size of the skew partition in the above example is 9.

A standard tableau of shape $\lambda \vdash n$ is a Young diagram of the partition λ where the boxes are filled with the numbers $\{1, ..., n\}$ such that the numbers are increasing left to right, top to bottom, and every number is used once.

Example 2.2.7. We draw the two standard tableaux of shape $(2, 1)$.

$$\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array} \qquad \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array}$$

We write $\mathrm{SYT}(\lambda)$ to denote the set of standard tableau of shape λ. We refer to partition properties of a tableau as properties of its shape. For example, the size of a tableau $T \in \mathrm{SYT}(\lambda)$ is $|\lambda|$. We can associate a word $w = w_1...w_n$ for each standard tableau $T \in \mathrm{SYT}(\lambda)$ with $|\lambda| = n$.

Definition 2.2.8. Define the set of Yamanouchi words to be the set of words $w = w_1...w_n$ $(n \in \mathbb{N})$ satisfies the property that $\#\{w_i : i \leq j, w_i = k\} \leq \#\{w_i : i \leq j, w_i = k-1\}$ for all $1 \leq j \leq n$ and $k > 1$.

The set of Yamanouchi words of length n is in bijection with standard tableaux of size n. The correspondence is to map a tableau T of size n ot the word $w_1...w_n$ where w_i is the row in which the letter i occurs (reading top to bottom).

Example 2.2.9. Consider the tableau

$$\begin{array}{|c|c|c|}\hline 1 & 3 & 4 \\ \hline 2 & 6 & \multicolumn{1}{c}{} \\ \cline{1-2} 5 & 7 & \multicolumn{1}{c}{} \\ \cline{1-2} \end{array}$$

The corresponding Yamanouchi word is 1211323.

A semistandard young tableau of shape λ is a Young diagram of the partition λ with the boxes filled with numbers such that the fillings weakly increase left to right and strictly increase top to bottom. Suppose that a semistandard young tableau uses the number i α_i times. Then we say that the content of that tableau is $\alpha(T) = (\alpha_1, \alpha_2, ...)$ and write $\alpha(T)_i = \alpha_i$.

Example 2.2.10. We draw a semistandard tableau of shape $(4, 3, 1)$ and content $(3, 2, 2, 1)$.

$$\begin{array}{|c|c|c|c|}\hline 1 & 1 & 1 & 3 \\ \hline 2 & 2 & 3 & \multicolumn{1}{c}{} \\ \cline{1-3} 4 & \multicolumn{3}{c}{} \\ \cline{1-1} \end{array}$$

The content does not need to be a partition. For example, here is a semistandard tableau with content $(0, 1, 3, 0, 2, 2)$.

$$\begin{array}{|c|c|c|c|}\hline 2 & 3 & 3 & 5 \\ \hline 3 & 4 & 5 & \multicolumn{1}{c}{} \\ \cline{1-3} 4 & \multicolumn{3}{c}{} \\ \cline{1-1} \end{array}$$

When the content is a partition, we say the tableau has *partition content*.

Example 2.2.11 (Schur functions)**.** We can take the multivariate generating series of all semistandard tableau of shape λ whose largest entry is n,

$$s_\lambda(x_1, ..., x_n) = \sum_T {x_1}^{\alpha(T)_1} {x_2}^{\alpha(T)_2} ... {x_n}^{\alpha(T)_n}$$

The s_λ are called *Schur functions*. This definition extends if we remove the restriction that the largest entry of the tableau is n, and we end up with $s_\lambda \in \mathbb{C}[x_1, x_2, ...]$. The construction for infinitely many variables satisfies $s_\lambda(x_1, ..., x_n, 0, 0, ...) = s_\lambda(x_1, ..., x_n)$. It is a non-trivial fact that the Schur functions are examples of *symmetric functions*. The symmetric functions,

$$\Lambda := \{f \in \mathbb{Q}[x_1, x_2, ...] : f(x_1, x_2, ...) = f(x_{\sigma(1)}, x_{\sigma(2)}, ...) \text{ for all bijections } \sigma : \mathbb{N} \to \mathbb{N}\}$$

form a commutative $\mathbb{Q}-$algebra, and the s_λ form a basis for Λ. The s_λ also have positive structure constants so that $s_\lambda s_\mu = \sum_\nu c^\nu_{\lambda\mu} s_\nu$ with each $c^\nu_{\lambda\mu} \in \mathbb{N}$. The coefficients $c^\nu_{\lambda\mu}$ are called *Littlewood-Richardson coefficients*.

The Schur functions, along with their structure constants $c^\nu_{\lambda\mu}$ will be needed in Sections 2.5, 2.6.

Definition 2.2.12. Let T be a semistandard tableau. The *row-reading word*, RR(T), is the word obtained by reading the entries left to right, bottom to top.

Example 2.2.13. Consider the semistandard tableau

$T =$

1	1	1	3
2	2	3	
4			

.

The corresponding row-reading word is

$$\mathrm{RR}(T) = 42231113.$$

We write SSYT(λ, μ) to denote the set of semistandard tableaux of shape λ and content μ. Also, we will use Row$_i(T)$ to refer to the ith row of T from the top. One can *standardize* a semistandard tableau by reading the entries in weakly increasing order, with equal entries being read left to right. We fill in the boxes with numbers $1, ..., |\lambda|$ in the same order that we read the entries in. More precisely, start with $x = 1$. Consider the entries of the semistandard tableau listed in increasing order. Start with the smallest one and read the cells in the tableau with that entry left to right. Replace the current cell with x and set $x = x + 1$.

Example 2.2.14. The semistandard tableau in Example 2.2.10 can be standardized as

1	2	3	7
4	5	6	
8			

Standard and semistandard tableaux can be defined for skew partitions as well.

Definition 2.2.15. A semistandard tableau of shape λ/μ and content α is a Young diagram of shape λ/μ where the boxes are filled with numbers weakly increasing left to right and strictly increasing top to bottom. The number of boxes which each number is recorded by α. A standard tableau of shape λ/μ is a semistandard tableau of shape λ/μ with content $1^{|\lambda/\mu|}$.

Just as for non-skew shapes we use SYT(λ/μ) and SSYT$(\lambda/\mu, \alpha)$ to denote standard tableaux of shape λ/μ and semistandard tableaux of shape λ/μ and content α respectively.

Definition 2.2.16. A *configuration* ν is a sequence of partitions $(\nu^{(0)}, \nu^{(1)}, ..., \nu^{(k)})$. A configuration ν is admissible if for all $a > 0$ and all $1 \leq t \leq \ell(\nu^{(a)})$. The *vacancy number* $P^{(a)}_{\nu^{(a)}_t}(\nu)$ is defined as

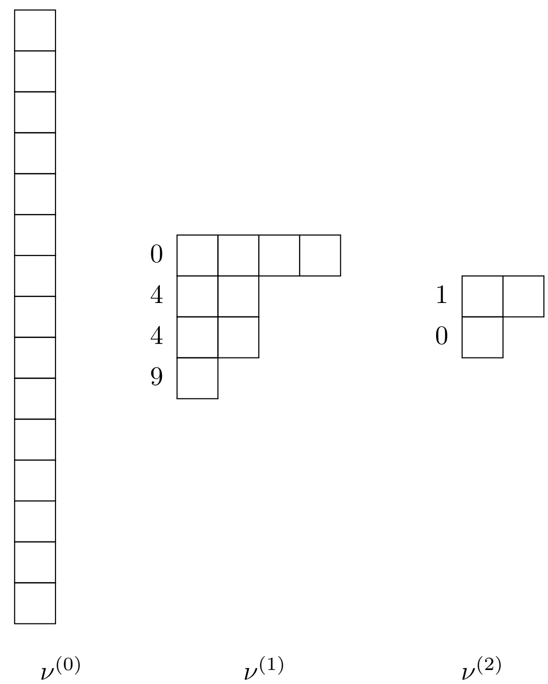
$$P^{(a)}_{\nu^{(a)}_t}(\nu) = \sum_{i=1}^{\nu^{(a)}_t} \overline{\nu^{(a-1)}_i} - 2\sum_{i=1}^{\nu^{(a)}_t} \overline{\nu^{(a)}_i} + \sum_{i=1}^{\nu^{(a)}_t} \overline{\nu^{(a+1)}_i} \tag{2.1}$$

where we define $\nu^{(a+1)} = (0, 0, ...)$ if $a + 1 > k$.

Definition 2.2.17. A configuration is *admissible* if all of its vacancy numbers are non-negative.

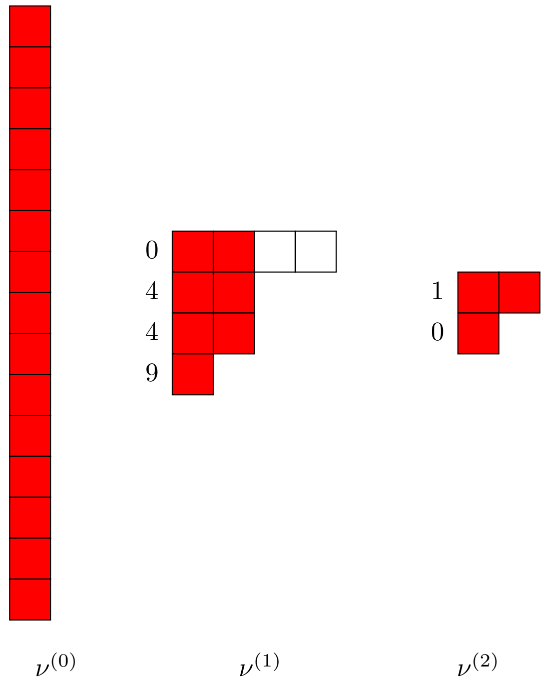
Configurations are usually drawn as a sequence of the partitions $\nu^{(0)}, \nu^{(1)}, ..., \nu^{(k)}$ next to each other in the plane where the vacancy number $P^{(a)}_{\nu^{(a)}_t}(\nu)$ is written in the Young diagram corresponding to $\nu^{(a)}$, to the left of its tth row (of course we only draw these vacancy numbers for $a > 1$).

Example 2.2.18. We draw an admissble configuration below, with the vacancy numbers drawn on the left following the description given above



Explanation. To explain how each of the vacancy numbers are computed above, we imagine each of the sums $\sum_{i=1}^{r} \overline{\nu^{(a-1)}}, \sum_{i=1}^{r} \overline{\nu^{(a)}}, \sum_{i=1}^{r} \overline{\nu^{(a+1)}}$ as counting the boxes in the first r columns of the corresponding Young diagrams. So, if we want to compute $P^{(1)}_2(\nu)$, we count the boxes in the first

2 columns of the diagrams $\nu^{(0)}, \nu^{(1)}, \nu^{(2)}$: (highlighting them with the colour red):

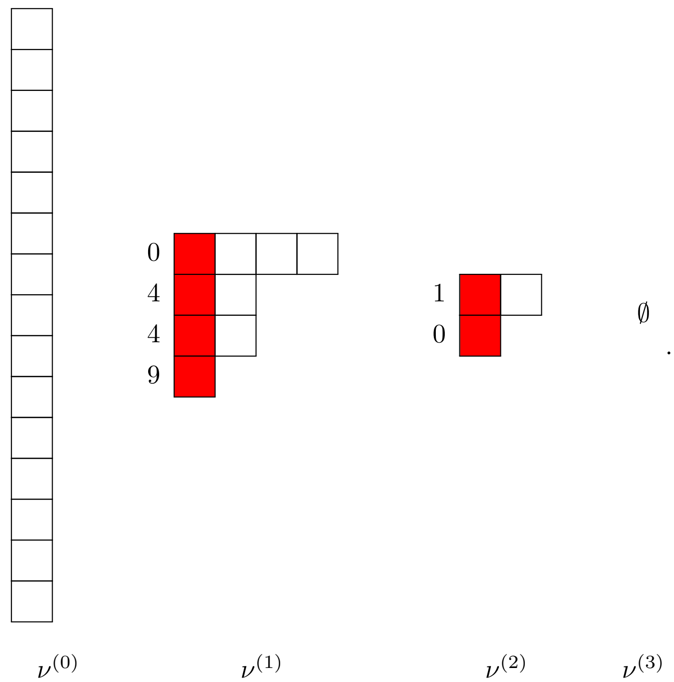


There are 15 red boxes in $\nu^{(0)}$, so the contribution from $\nu^{(0)}$ is 15. There are 7 red boxes in $\nu^{(1)}$, so the contribution from $\nu^{(1)}$ is $-2 \times 7 = 14$. There are 3 red boxes in $\nu^{(2)}$. So the contribution from $\nu^{(2)}$ is 3. Hence,

$$P_2^{(1)}(\nu) = 15 - 2 \times 7 + 3 = 4.$$

As another example, let us compute $P_1^{(2)}(\nu)$. Note that there is no third partition. So we interpret

$\nu^{(3)} = (0, 0, ...)$. Again, we count the number of boxes in the first column of $\nu^{(1)}, \nu^{(2)}, \nu^{(3)}$:



There are 4 red boxes in $\nu^{(1)}$. So the contribution from $\nu^{(1)}$ is 4. There are 2 red boxes in $\nu^{(2)}$, so the contribution from $\nu^{(2)}$ is -2×2. There are no red boxes in $\nu^{(3)}$ so the contribution from $\nu^{(3)}$ is 0. Hence,

$$P_1^{(2)}(\nu) = 4 - 2 \times 2 + 0 = 0.$$

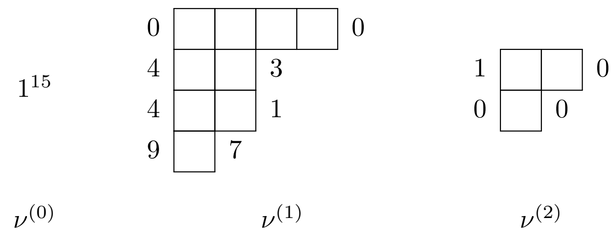
□

Informally, a rigged configuration is a drawing of an admissible configuration (as in Example 2.2.18), but we draw extra numbers to the right of each row which are at most the corresponding vacancy number (which we call riggings). The order of riggings among rows of the same length do not matter. More precisely,

Definition 2.2.19. A rigged configuration $X = (\nu, J)$ is a configuration $\nu = (\nu^{(0)}, \nu^{(1)}, ..., \nu^{(k)})$ with a set of riggings $J = (J^{(1)}, ..., J^{(k)})$ where each $J_x^{(a)}$ is a partition of width $P_x^{(a)}(\nu)$.

If $X = (\nu, J)$ is a rigged configuration, we will identify the vacancy numbers $P_x^{(a)}(X) = P_x^{(a)}(\nu)$. In contrast to the above definition, it is more convenient think of the rigged configuration (ν, J) in terms of its rigged partitions $(\nu, J)^{(1)}, ..., (\nu, J)^{(k)}$ where each $(\nu, J)^{(a)}$ is a multiset containing pairs of the form (x, α) where x is a part of $\nu^{(a)}$ and $\alpha \leq P_x^{(a)}(\nu)$ is its corresponding rigging. We can also define the corigged partitions $\widetilde{(\nu, J)}^{(a)} := \{(x, P_x^{(a)} - \alpha) : (x, \alpha) \in (\nu, J)^{(a)}\}$. Note that the partition $\nu^{(0)}$ does not have any vacancy numbers of riggings. This motivates the definition of the proper configuration $(\nu^{(1)}, ..., \nu^{(k)})$. When drawing a rigged configuration, the riggings are drawn to the right of each row, contrasting how the vacancy numbers are drawn.

Example 2.2.20. We draw a rigged configuration below, where we have omitted the Young diagram of $\nu^{(0)} = 1^{15}$.



We write $\mathrm{RC}(\lambda, \mu)$ to be the set of rigged configurations (ν, J) satisfying $\lambda_i = |\nu^{(i-1)}| - |\nu^{(i)}|$ for each $i \geq 1$ and $\nu^{(0)} = \mu$. The above example is a rigged configuration in $\mathrm{RC}((4,4),(1^{12}))$. Rows whose riggings are equal to their corresponding vacancy number are called *singular*. In the above example, the first rigged partition has one singular row of length 4. The second rigged partition has one singular row of length 1.

Note that the computation of vacancy numbers is local. Thus, if we remove the content partition $\nu^{(0)}$, and the riggings on the first rigged partition $(\nu, J)^{(1)}$ we get another rigged configuration. More precisely,

Definition 2.2.21. Let $X = (\nu, J)$ be a rigged configuration. Define the *hide* of a rigged configuration X to be the rigged configuration $\mathrm{hide}(X) = Y = (\nu', J')$ such that ${\nu'}^{(i-1)} = \nu^{(i)}$ for $1 \leq i \leq \ell(\nu)$ and $Y^{(i)} = X^{(i)}$ for $i > 1$.

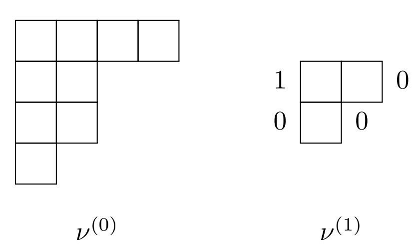
This gives us a "nesting structure" on rigged configurations. In particular, we can think of the hide of a rigged configuration as being "contained" in the original. In this light, we have the chain of rigged configurations where each successive configuration has one less partition.

$$X \supset \mathrm{hide}(X) \supset \mathrm{hide}(\mathrm{hide}(X)) \supset ... \supset \emptyset$$

The nesting structure also has a crystal theoretic interpretation [9]. As we will see later, the top layer of this nesting structure (that is, the first rigged partition) encodes a lot of information about the "descent information" of a tableau. So computing hide lets us determine which tableaux are equivalent modulo some properties about their descents. Respectively, it allows us to determine when two rigged configurations are the same modulo rigging on the first rigged partition (assuming they have the same $\nu^{(0)}$). We will use this language to later define *spells* in Section 2.8.

Additionally, it is far easier to compute $\nu^{(0)}$ directly from the tableau (we will see in the next section that this is just the content of the tableau) than it is to compute any other component of the configuration. As such, if hide can be understood as a function directly on tableaux, then the corresponding configurations can be computed directly on the tableau.

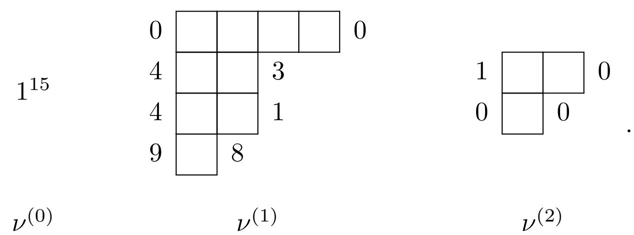
Example 2.2.22. The hide of the rigged configuration in Example 2.2.20 is



For later use, we will define an operation on rigged configurations which induces a small perturbation on the riggings of the first rigged partition.

Definition 2.2.23. Let $X = (\nu, J)$ be a rigged configuration. Let $\rho = \min_{1\le i\le \ell(\nu^{(1)})} \nu_i^{(1)}$, and let $\alpha = \max_{1\le i\le \ell(\nu^{(1)})}(x : (\nu_i^{(1)}, x) \in X^{(1)})$. Define $X^\sharp := Y$ where X, Y have the same configuration, $Y^{(i)} = X^{(i)}$ for $i > 1$ and $Y^{(1)} = (X^{(1)} \setminus \{(\rho,\alpha)\}) \cup \{(\rho, \alpha+1)\}$.

Example 2.2.24. Say the rigged configuration in Example 2.2.20 is X. Then $X^\sharp$ is depicted below,



Sometimes we will need to talk about subsets of rigged partitions. For this, we introduce the following notation.

Definition 2.2.25. Let X be a rigged configuration and $I \subseteq \mathbb{N}$. Define

$$X_I^{(i)} := \{(x,\alpha) \in X^{(i)} : x \in I\}$$

and similarly,

$$\widetilde{X_I}^{(i)} := \{(x,\alpha) \in \tilde{X}^{(i)} : x \in I\}.$$

When $I = (-\infty, k]$ then we write $X_{\le k}^{(i)} := X_I^{(i)}$ (respectively $\widetilde{X_{\le k}}^{(i)} := \widetilde{X_I}^{(i)}$). Similarly when $I = [k, \infty)$ we write $X_{\ge k}^{(i)} := X_I^{(i)}$ (respectively, $\widetilde{X_{\ge k}}^{(i)} = \widetilde{X_I}^{(i)}$).

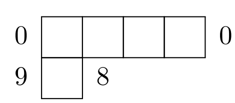
Example 2.2.26. Consider Example 2.2.20. The first rigged partition is

$$X^{(1)} = \{(4,0), (2,3), (2,1), (1,8)\}.$$

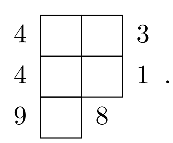
If we let $I = \{1, 4\}$, then

$$X_I{}^{(1)} = \{(4,0), (1,8)\}.$$

Pictorially,

$X_I{}^{(1)} =$  .

As another example,

$X_{\le 3}^{(1)} =$  .

2.3 The KKR Bijection

In [7], the authors defined a bijecton Φ between $\mathrm{SSYT}(\lambda, \mu)$ and $\mathrm{RC}(\lambda, \mu)$ now known as the Kerov-Kirillov-Reshetikhin (KKR) bijection. Some authors also call it the Kirillov-Schilling-Shimozono (KSS) bijection, or the rigged configuration bijection. It should be stated that the KSS bijection is actually a generalization of the KKR bijection, but this level of generality will not be necessary to us. As such, the version of the KKR bijection we will be concerned with is that of [7]. That is, we will describe the KKR bijection as a map between semistandard tableaux and rigged configurations.

For the purposes of this algorithm, we view a configuration ν as a finite sequence of partitions followed by an infinite number of empty partitions (i.e $\nu^{(M)} = \emptyset$ for large M).

Algorithm 2.3.1. Let T be a semistandard tableau, and read the entries in weakly increasing order with equal entries being read left to right. We interatively build a rigged configuration, starting with the empty configuration. For each entry that is read, we perform the following procedure. Let $X = (\nu, J)$ be the rigged configuration that has already been constructed by this step in the procedure. Let α be the current entry that is read in T, and let r be the row in which the read cell occurs.

(1) Define $m_1 \leq ... \leq m_{r-1}$ so that m_{r-1} is the length of the largest singular row in $X^{(r-1)}$ (or $m_{r-1} = 0$ if one doesn't exist), and for $i < r-1$, m_i is the length of the largest singular row in $X^{(i)}$ (or $m_i = 0$ if one doesn't exist) subject to the condition that $m_i \leq m_{i+1}$.

(2) For each $1 \leq i < r$ define rigged partitions

$$X'^{(i)} = (X^{(i)} \setminus \{(m_i, P^{(i)}_{m_i}(X))\}) \cup \{(m_i + 1, g_i)\}$$

where g_i is to be determined at a later step.

(3) Let m_0 be the number of copies of α read before the current cell. Define the partition $\mu^{(0)}$ to be $\nu^{(0)}$ with a row of length m_0 increased by one. In particular, if $m_0 = 0$, then $\mu^{(0)}$ is $\nu^{(0)}$ with one additional part of size 1.

(4) Let $\mu^{(i)}$ for $1 \leq i < r$ be the shape of $X'^{(i)}$ and $\mu^{(i)} = \nu^{(i)}$ for $i \geq r$. Define the configuration $\mu = (\mu^{(0)}, \mu^{(1)}, ..., \mu^{(r-1)}, \mu^{(r)}, ...)$.

(5) Set $g_i = P^{(i)}_{m_i+1}(\mu)$.

(6) Replace X with the rigged configuration whose configuration is ν and whose rigged partitions are $X'^{(i)}$ for $1 \leq i < r$, and $X^{(i)}$ for $i \geq r$.

The resulting rigged configuration $X = (\nu, J)$ after all entries of T have been read is $X = \Phi(T)$.

Remark 2.3.2. At each step, the zeroeth partition $\nu^{(0)}$ records the content of the partially read tableau up to that step.

Remark 2.3.3. Often, we will consider the chain of singular rows $m_0 \leq m_1 \leq ... \leq m_{r-1}$ to contain a zeroeth element. Here m_0 is the length of the row in $\nu^{(0)}$ which is lengthened.

Informally, we can describe the algorithm as looking at the row in which the cells of our tableau occur according to our reading order, find $m_1 \leq ... \leq m_{r-1}$ according to step (1). Change $\nu^{(0)}$ to be equal to content of the partial tableau read. Then, increase each of the singular rows in $X^{(i)}$ of length m_i by one cell, giving them placeholder riggings g_i. We then pick g_i so that the lengthened rows remain singular under the new vacancy numbers.

We will give two worked examples below. The first example is small and will be done in its entirety, using the formal description given in Algorithm 2.3.1. The second example will be done on a large tableau, but only the steps corresponding to the last couple of cells read. We will also do a full example of a large tableau, but without words of explanation for the steps (see Example 2.3.5).

First, let us find the rigged configuration corresponding to the following tableau.

$$T = \begin{array}{|c|c|c|}\hline 1 & 1 & 3 \\ \hline 2 & 2 \\ \cline{1-2} \end{array}$$

We decorate our tableau with subscripts which indicates where they occur in our reading order:

$$T = \begin{array}{|c|c|c|}\hline 1_1 & 1_2 & 3_5 \\ \hline 2_3 & 2_4 \\ \cline{1-2} \end{array}.$$

We will identify the cells of the tableau with their entry in this decorated tableau (for example 2_4 refers to the right most cell in the second row, and has filling 2).

The cell 1_1 occurs in the first row so $r = 1$. Therefore, steps (1) and (2) do not do anything (note there is no zeroeth *rigged* partition. At step (3), we note that zero copies of 1 have been read previously (because this is our first step). Hence, $m_0 = 0$ so $\mu^{(0)} = (0 + 1) = (1)$. At step (4) we see that

$$\mu = \underset{\mu^{(0)}}{\begin{array}{|c|}\hline \\ \hline \end{array}}.$$

At step (6) we conclude the rigged configuration corresponding to the partial tableau

$$T' = \begin{array}{|c|}\hline 1 \\ \hline \end{array}$$

is just the configuration we found earlier,

$$\underset{\nu^{(0)}}{\begin{array}{|c|}\hline \\ \hline \end{array}}.$$

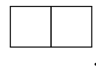
The next cell to be read is 1_2, which also occurs in the first row so again $r = 1$. Because we again have $r = 1$, the only steps which apply here are (3), (4), and (6). For (3), we note that exactly one previously read cell had filling 1 (namely the first and only previously read cell, 1_1). So, $m_0 = 1$ and hence $\mu^{(0)} = (1 + 1) = (2)$. At step (4), we again see that our configuration μ only has the zeroeth partition,

$$\mu = \underset{\mu^{(0)}}{\begin{array}{|c|c|}\hline & \\ \hline \end{array}}.$$

Then, moving to step (6) we see that the rigged configuration corresponding to the partial tableau

$$T' = \begin{array}{|c|c|}\hline 1 & 1 \\ \hline \end{array}$$

is



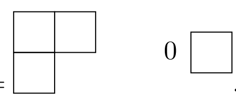
$\nu^{(0)}$

The next cell to be read is 2_3, which occurs in the second row. Hence, $r = 2$. At step (1), we look for the largest singular row in $X^{(1)}$. Because $X^{(1)} = \emptyset$ at this stage, there is no largest singular row here. Hence, $m_1 = 0$. By step (2) we have

$$\begin{aligned} X'^{(1)} &= (X^{(1)} \setminus \{(0, P_0^{(1)}(X))\}) \cup \{(1, g_0)\} \\ &= (\emptyset \setminus \{(0, P_0^{(1)}(X))\}) \cup \{(1, g_0)\} \\ &= \emptyset \cup \{(1, g_1)\} \\ &= \{(1, g_1)\}. \end{aligned}$$

The number of previously read cells which had filling 2 is 0 (because both of the previously read cells had filling 1). So, $m_0 = 1$ and hence $\mu^{(0)} = (2, 0 + 1) = (2, 1)$. Now at step (4), we see that $X'^{(1)}$ has shape 1, and $\mu^{(0)}$ has shape $(2, 1)$ which gives us the configuration

$\mu =$

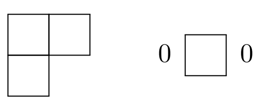


$\mu^{(0)}$ $\mu^{(1)}$

Because $P_{0+1}^{(1)}(\mu) = 0$, we have $g_1 = 0$. Hence, $X'^{(1)} = \{(1, 0)\}$. By step (6), the rigged configuration corresponding to the partial tableau

$$T' = \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 2 & \\ \cline{1-1} \end{array}$$

is



$\nu^{(0)}$ $\nu^{(1)}$

The next cell to be read is 2_4, which also occurs in the second row, so that $r = 2$. At step (1), we see that the largest singular row in $X^{(1)}$ is of length 1. So $m_1 = 1$. For step (2), we have $X^{(1)} = \{(1, 0)\}$ and $P_0^{(1)}(X) = 0$ so that

$$X'^{(1)} = (X^{(1)} \setminus \{(1, 0)\}) \cup \{(2, g_1)\} = \{(2, g_1)\}.$$

At step (3), we realize there is exactly one previously read cell with a 2 inscribed within it. Therefore, $m_0 = 1$ and so $\mu^{(0)} = (2,2)$. Now at step (4), noting that ${X'}^{(1)}$ has shape (2), we have

$$\mu = \underset{\mu^{(0)}}{\begin{array}{|c|c|}\hline & \\ \hline & \\ \hline \end{array}} \qquad \underset{\mu^{(1)}}{0\,\begin{array}{|c|c|}\hline & \\ \hline \end{array}}\ .$$

We see the vacancy number $P^{(1)}_{1+1}(\mu) = 0$ so that $g_1 = 0$. Hence, the rigged configuration corresponding to the partial tableau

$$\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 2 & 2 \\ \hline \end{array}$$

is

$$\mu = \underset{\mu^{(0)}}{\begin{array}{|c|c|}\hline & \\ \hline & \\ \hline \end{array}} \qquad \underset{\mu^{(1)}}{0\,\begin{array}{|c|c|}\hline & \\ \hline \end{array}\,0}\ .$$

Finally, we read the last cell 3_5, which occurs in the first row, so $r = 1$. Because $r = 1$, steps (1) and (2) terminate immediately. For step (3), we note that none of the previously read cells had filling 3. Therefore, $m_0 = 0$ and $\mu^{(0)} = (2,2,0+1) = (2,2,1)$. At step (6), we see that X has configuration

$$\mu = \underset{\mu^{(0)}}{\begin{array}{|c|c|}\hline & \\ \hline & \\ \hline & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array}} \qquad \underset{\mu^{(1)}}{1\,\begin{array}{|c|c|}\hline & \\ \hline \end{array}}\ .$$

and rigged partition $X^{(1)} = \{(2,0)\}$. Hence, the rigged configuration corresponding to

$$T = \begin{array}{|c|c|c|}\hline 1 & 1 & 3 \\ \hline 2 & 2 & \multicolumn{1}{c}{} \\ \cline{1-2} \end{array}$$

is

$$\underset{\mu^{(0)}}{\begin{array}{|c|c|}\hline & \\ \hline & \\ \hline & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array}} \qquad \underset{\mu^{(1)}}{1\,\begin{array}{|c|c|}\hline & \\ \hline \end{array}\,0}\ .$$

For our large example, consider the tableau

$$T = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 1 & 2 & 2 & 3 \\ \hline 2 & 3 & 4 & \multicolumn{3}{c}{} \\ \cline{1-3} 3 & 4 & 5 & \multicolumn{3}{c}{} \\ \cline{1-3} 4 & \multicolumn{5}{c}{} \\ \cline{1-1} \end{array}\ .$$

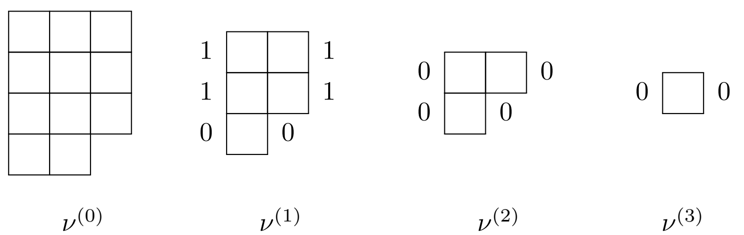
Writing T decorated with subscripts indicating our reading order, we have

$$T = \begin{array}{|c|c|c|c|c|c|} \hline 1_1 & 1_2 & 1_3 & 2_5 & 2_6 & 3_9 \\ \hline 2_4 & 3_8 & 4_{12} \\ \cline{1-3} 3_7 & 4_{11} & 5_{13} \\ \cline{1-3} 4_{10} \\ \cline{1-1} \end{array} .$$

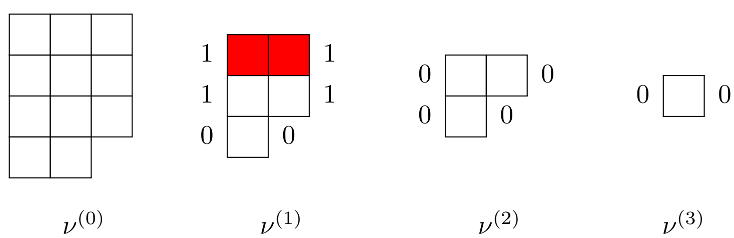
Let us assume that we have already computed the first 11 steps. That is, we know the partial tableau

$$T' = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 2 & 2 & 3 \\ \hline 2 & 3 \\ \cline{1-2} 3 & 4 \\ \cline{1-2} 4 \\ \cline{1-1} \end{array}$$

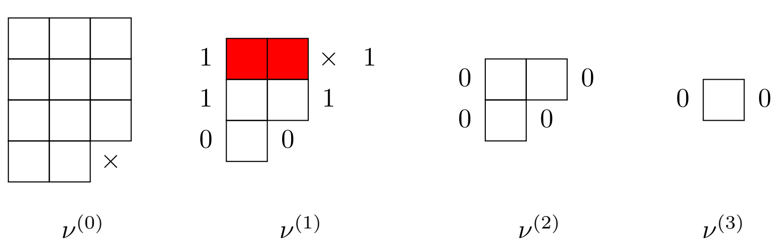
corresponds to the rigged configuration



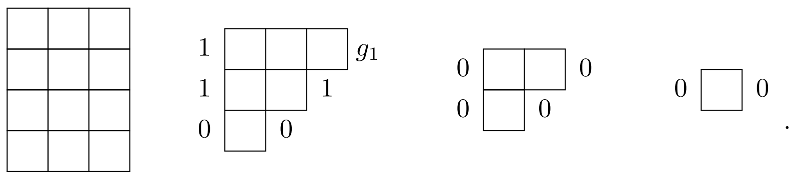
The last cell that was read was 4_{11}. Hence, the next cell to be read is 4_{12}, which occurs in row 2 so that $r = 1$. The largest singular row of $X^{(1)}$ is of length 2 so that $m_1 = 1$. There were also 2 previously read copies of 4, so $m_0 = 2$. We draw the rigged configuration X below, but we use red to indicate the singular rows of length m_i in $X^{(i)}$. In effect, this is a depiction of step (1).



To illustrate steps (2) and (3), we use the filling "×" to indicate the way boxes are added.



Now, we execute the change to the configuration, by replacing the "×" with actual boxes. As described in step (2), we use the placeholder riggings g_i on any row that was lengthened:

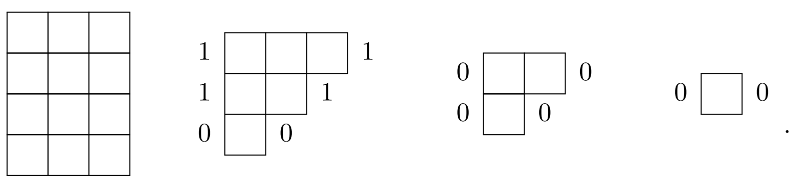


$\nu^{(0)}$ $\nu^{(1)}$ $\nu^{(2)}$ $\nu^{(3)}$

We then notice that the vacancy number to the left of g_1 is 1. So we set $g_1 = 1$. It follows that the rigged configuration corresponding to the partial tableau

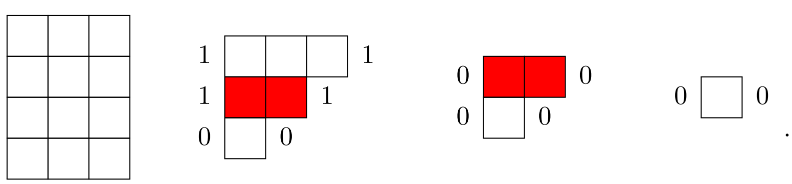
$$T' = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 1 & 2 & 2 & 3 \\ \hline 2 & 3 & 4 \\ \cline{1-3} 3 & 4 \\ \cline{1-2} 4 \\ \cline{1-1}\end{array}$$

is



$\nu^{(0)}$ $\nu^{(1)}$ $\nu^{(2)}$ $\nu^{(3)}$

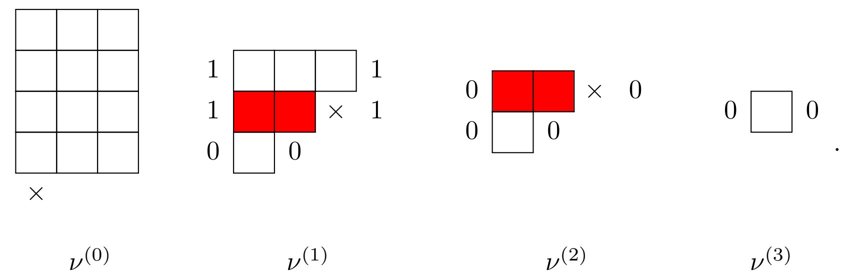
Now we read the final cell 5_{13}, which occurs in the third row so that $r = 3$. The largest singular row in $X^{(2)}$ is of length 2 so that $m_2 = 2$. We see that $X^{(1)}$ has three singular rows of lengths $1, 2, 3$. Note that $m_1 = 2$ here because 2 is the length of the largest singular row at most $m_2 = 2$. This is illustrated below:



$\nu^{(0)}$ $\nu^{(1)}$ $\nu^{(2)}$ $\nu^{(3)}$

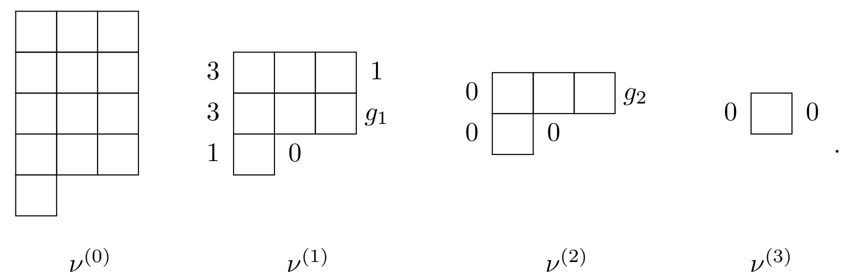
There were zero copies of 5 previously read, so $m_0 = 0$. We again give a depiction of steps (2) and

(3) below:



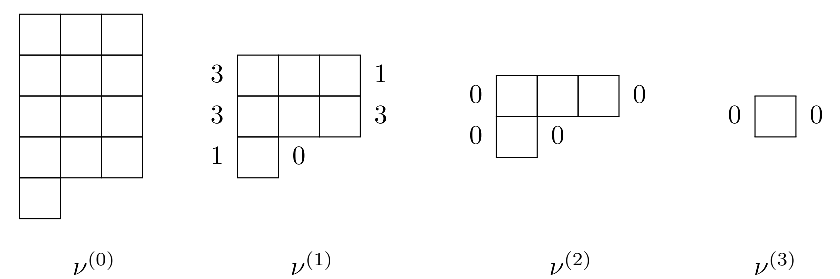
(2.2)

Executing the change to the configuration, we have



(2.3)

We take $g_1 = 3, g_2 = 0$ to make the lengthened rows singular (i.e, take them so that they equal to the corresponding vacancy number to the left). We get,



which is the desired rigged configuration corresponding to T.

The inverse of Φ is also described in [7], which we also describe iteratively.

Algorithm 2.3.4. Let X be a rigged configuration. At each stage of the algorithm, we will have a rigged configuration $Y = (\nu, J)$ and a partial tableau S, where we start with $Y = X$ and $S = \emptyset$. Repeat the following until Y has the empty configuration.

(1) Let m_0 be the smallest row in $\nu^{(0)}$.

(2) Define m_i to be the smallest singular row in $Y^{(i)}$ subject to the condition that $m_i \geq m_{i-1}$.

(3) Repeat step (2) until no such i exists. Then we $m_0 \leq m_1 \leq ... \leq m_r$.

(4) Define $\mu^{(0)}$ to be $\nu^{(0)}$ with a row of length m_0 decreased by one.

(5) For each $1 \leq i \leq r$, if $m_i > 1$, define the rigged partition

$$Y'^{(i)} := (Y^{(i)} \setminus \{(m_i, P^{(i)}_{m_i}(Y))\}) \cup \{(m_i - 1, g_i)\}$$

where g_i is to be determined later. If $m_i = 1$, then define

$$Y'^{(i)} := Y^{(i)} \setminus \{(m_i, P^{(i)}_{m_i}(Y))\}.$$

(6) Let $\mu^{(i)}$ be the shape of $Y'^{(i)}$ for $1 \leq i \leq r$, and $\mu^{(i)} = \nu^{(i)}$ for $i > r$. Define the configuration $\mu = (\mu^{(0)}, \mu^{(1)}, ..., \mu^{(r)}, ...)$.

(7) Set $g_i = P^{(i)}_{m_i - 1}(\mu)$.

(8) Replace Y with the rigged configuration whose configuration is μ, and whose rigged partitions are $Y'^{(i)}$ for $1 \leq i \leq r$ and $Y^{(i)}$ for $i > r$.

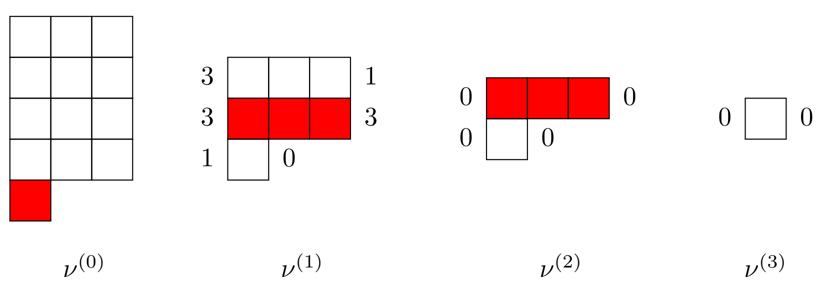
(9) Replace S with the tableau obtained by adding a box with filling $\ell(\nu^{(0)})$ to the $(r+1)$st row, and rearranging the entries of each row in weakly increasing order.

The resulting tableau S that is obtained when Y is the empty configuration is $S = \Phi^{-1}(X)$.

The informal description of this algorithm is at each stage, we reduce our rigged configuration, and build our tableau. To reduce our rigged configuration, we do steps (1), (2), and (3) to find $m_1 \leq ... \leq m_r$. We reduce the smallest row of $\nu^{(0)}$ by one. We reduce the length of a singular row of length m_i in $X^{(i)}$ by one, giving them placeholder riggings g_i. We then set g_i so that the shortened rows remain singular under the new vacancy numbers. All other riggings remain unchanged. To build our tableau, we insert a box at the $(r+1)$st row. The inscribed number is the length of $\nu^{(0)}$ before it is modified and this new rigged configuration corresponds to the tableau obtained by removing largest, right-most entry.

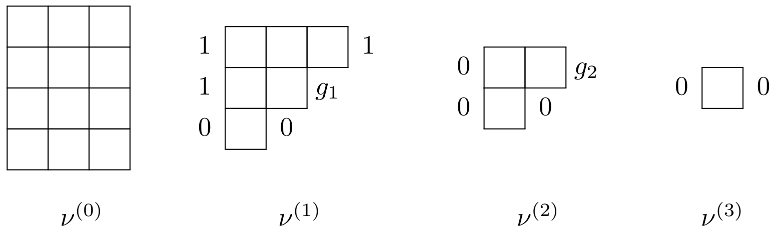
As a worked example, we will compute two steps of the inverse KKR bijection with the large example done above. This should recover the rows in which the last two boxes occur (with respect to the reading order), as well as the number inscribed in them.

We highlight the row in $\nu^{(0)}$ of length m_0 in red. We also highlight a singular row in each $X^{(i)}$ which is of length m_i. This in effect is a pictorial representation of steps (1), (2), (3).

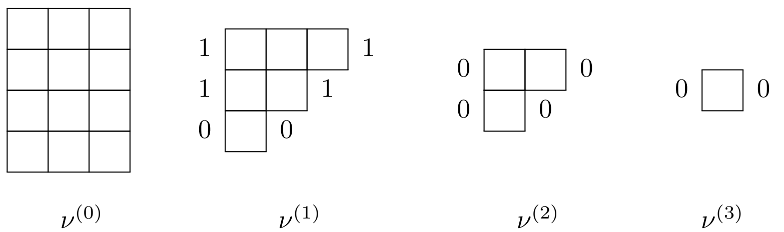


Note that there is no highlighted row in $X^{(3)}$, because there is no singular row of length at least $3 = m_2$ in $X^{(3)}$. This implies that $r = 2$. In particular, the last read box in the tableau occured in row $2 + 1 = 3$. Because the length of $\nu^{(0)}$ is 5, we can also deduce that this last read box must have contained a 5.

Continuing, we decrease each of the highlighted rows by one, and add placeholder riggings on the modified rows:

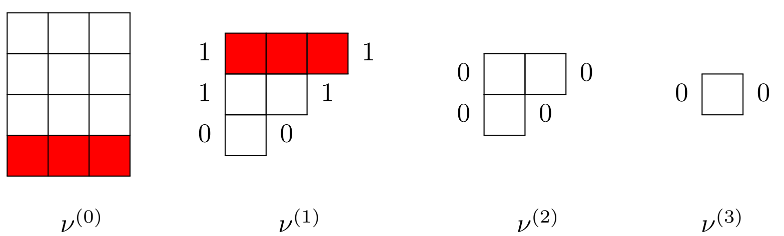


In order to make the shortened rows singular, we set $g_1 = 1$ and $g_2 = 0$. So, we obtain



This rigged configuration should correspond to the tableau obtained by removing the right-most occurence of 5 by Algorithm 2.3.4 (which we can verify it does by an earlier example).

To obtain the second last read cell, along with its filling, we do the same procedure on the resulting rigged configuration above. We find $m_0 \leq m_1 \leq ...$ in accordance with steps (1), (2), (3) and highlight the corresponding rows below:

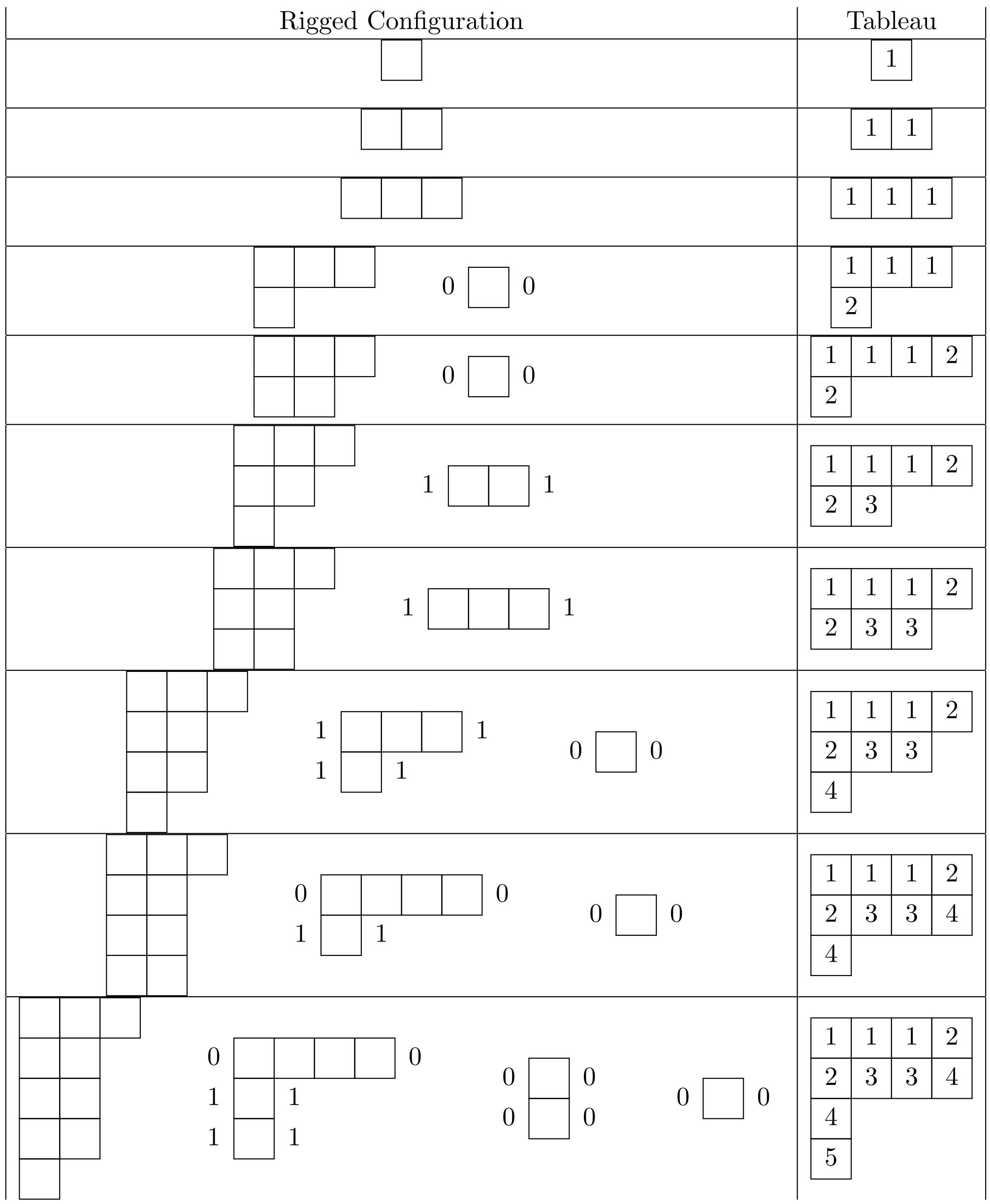


We have $m_0 = 3$ because the last entry of $\nu^{(3)}$ is 3. Note that although every row of $X^{(1)}$ is singular, the only (and hence smallest) singular row of length at least $m_0 = 3$ is the one of length 3, which tells us $m_1 = 3$. We also see that there is no singular row in $X^{(2)}$ of length at least $m_1 = 3$. This implies $r = 1$, so the last second last box with respect to the reading order occurs in row $r + 1 = 2$. Because the length of $\nu^{(0)}$ is 4, we also deduce that this box contains a 4. This is consistent with our "large" example of 2.3.1. To get the row in which the third last box occurs as well as its filling, we reduce the above rigged configuration in the same manner as before (i.e reducing the highlighted rows by one box, and giving them new riggings so that they remain singular under the new vacancy numbers).

Note that our description of Φ does not require the tableau T to have partition content. Hence, we can define it on all semistandard tableaux. In this case, the KKR bijection is no longer a bijection. As such, we will call it the KKR map when the domain are all semistandard tableaux.

Although the KKR map is not a bijection, the "kernel" of this extended map is understood. That is, we understand exactly when two semistandard tableaux yield the same rigged configuration. This result due to Kirillov, Schilling, and Shimozono will be stated in a later section after some more background has been introduced.

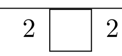
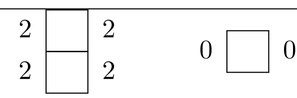
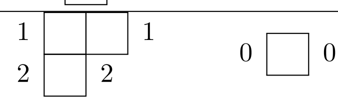
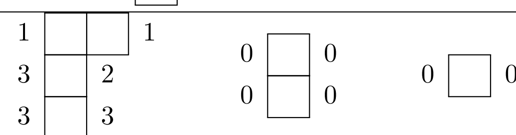
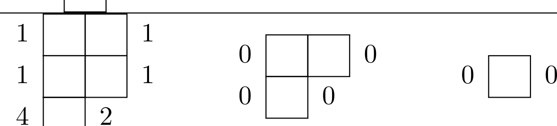
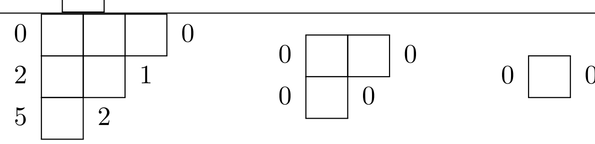
Example 2.3.5. Here is a full example of the KKR bijection being applied to a "large" tableau.



Define $\mathrm{RC}(\lambda) := \mathrm{RC}(\lambda, (1^{|\lambda|}))$. Note that the nature of the KKR bijection allows us to view

$\Phi : \mathrm{SYT}(\lambda) \to \mathrm{RC}(\lambda)$ as a map between Yamanouchi words and rigged configurations, instead of as a map between standard tableaux and rigged configurations.

Example 2.3.6. In this example, we will not draw the Young diagram of the zeroeth partition.

Rigged Configuration	Yamanouchi word
1	1
1^2	11
1^3	111
1^4 	1112
1^5 	11123
1^6 	111232
1^7 	1112324
1^8 	11123243
1^9 	111232432

2.4 Basic Combinatorics of Rigged Configurations

The KKR bijection is an iterative procedure. By a "step" of the KKR bijection, we mean the work done after reading one cell of the tableau. For example, in Examples 2.3.5 and 2.3.6, each row illustrates one "step" of the KKR bijection (reading backwards we also get the "steps" of the inverse KKR bijection). With that in mind, we would like to understand how the steps of the KKR bijection relates to the tableau structure. To do so, we first need a method of talking about the steps of the KKR bijection precisely, which the following definition enables us to do.

Definition 2.4.1. Let T be a semistandard tableau. At step i of the algorithm for computing $\Phi(T)$, the KKR bijection considers a chain $m_0^{(i)} \leq m_1^{(i)} \leq ... \leq m_{w_i-1}^{(i)}$, and adds a box to the ends of singular rows of of lengths $m_j^{(i)}$ in $\nu^{(j)}$. The *chain table* $\mathcal{M}(w)$ is the function $\mathcal{M}(w) : (i,j) \mapsto m_i^{(j)}$.

We can view the chain table as a matrix

$$\begin{bmatrix} m_0^{(1)} & m_0^{(2)} & ... & m_0^{(n)} \\ m_1^{(1)} & m_1^{(2)} & ... & m_1^{(n)} \\ . & . & ... & . \\ . & . & ... & . \end{bmatrix}$$

where if $j \geq w_i$ the entry $m_j^{(i)}$ is the trivial element ϵ.

The entry "ϵ" is not to be interpreted as a number. It is meant to indicate that no change was made to that partition corresponding to the row where the ϵ occured. For example, if $\mathcal{M}(T)_{i,j} = \epsilon$, then at the jth step of the algorithm, there is no change made to $\nu^{(i)}$.

Example 2.4.2. Consider Example 2.3.5. Here is the corresponding chain table (where we write "$\cdot$" for ϵ for ease of reading).

$$\begin{bmatrix} 0 & 1 & 2 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \cdot & \cdot & \cdot & 0 & \cdot & 1 & 2 & 0 & 3 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \end{bmatrix}$$

Example 2.4.3. Consider Example 2.3.6. We write the corresponding chain table and index the columns by letters in the Yamanouchi word.

$$\begin{matrix} 1 & 1 & 1 & 2 & 3 & 2 & 4 & 3 & 2 \end{matrix}$$
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & 0 & 0 & 1 & 0 & 1 & 2 \\ \cdot & \cdot & \cdot & \cdot & 0 & \cdot & 0 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & \cdot & \cdot \end{bmatrix}$$

Remark 2.4.4. The entries in row a of the chain table completely determine the partition $\nu^{(a)}$. We also observe the entries weakly increase top to bottom due to the way the KKR bijection selects singular rows to lengthen.

As stated in the introduction, we restrict ourselves to standard tableaux. So, we will assume all rigged configurations (ν, J) satisfy $\nu^{(0)}$ being a single column unless otherwise stated. Therefore, we can always talk about the columns of the chain table corresponding to letters of the Yamanouchi word.

At each step of the KKR bijection, singular rows are selected in each rigged partition indexed by $1, ..., r$ for some r. These singular rows are then lengthened. Even if we restrict ourselves to looking at the first rigged partition, it is difficult to predict which row in the first rigged partition is lengthened by a step of the KKR bijection. Understanding situations in which the row the KKR bijection chooses to lengthen in the first rigged partition is well behaved is a step towards controlling the chaotic behaviour of the KKR bijection. We make the following definition, which captures the situation when the same row in the first rigged partition is chosen at consecutive steps in the KKR bijection.

Definition 2.4.5 (Connectedness)**.** Let $w = w_1 ... w_n$ be a Yamanouchi word. Consider the consecutive descent sequence $w_a w_{a+1} ... w_b$. We say the subword $w_a w_{a+1} ... w_b$ is *connected* if $\mathcal{M}(w)_{1,a+t} = t$ for $t \in [0, b-a]$. In other words, the entries $\mathcal{M}(w)_{1,x}$ for $x \in [a, b]$ are consecutive.

Example 2.4.6. Consider Example 2.4.3. The Yamanouchi word which indexes the columns is 111232432. We see that the first row (note that we index the rows starting at 0) has consecutive entries at columns 5 to 6. As such, the subword corresponding to the 5th and 6th letter, 32, is connected. Additionally, the first row has consecutive entries $0, 1, 2$ at columns $7, 8, 9$. Thus, the subword corresponding to the 7th, 8th, and 9th letters (432) is also connected.

The main goal of this section will be to find sufficient conditions on Yamanouchi words for connectedness (Theorem 2.4.21).

To give a little credence to the importance of the chain table and tracking the chains of singular rows considered in the KKR bijection, we will explain how keeping track of that information also helps us keep track of the vacancy numbers at each step of the algorithm.

It is tedious to recompute the vacancy numbers at each step of the KKR bijection using equation 2.1. It is far easier to track the way the vacancy numbers change at each step. Recall that each sum in the formula

$$P_r^{(a)}(X) = \sum_{i=1}^{r} \overline{\nu_i^{(a-1)}} - 2\sum_{i=1}^{r} \overline{\nu_i^{(a)}} + \sum_{i=1}^{r} \overline{\nu_i^{(a+1)}}$$

represents the number of boxes in the first r columns of the corresponding Young diagrams. Each step the KKR bijection considers a chain of singular rows $m_0 \leq m_1 \leq ... \leq m_{r-1}$. To track the changes in the vacancy numbers, we look at the three Young diagrams $\nu^{(a-1)}, \nu^{(a)}, \nu^{(a+1)}$. We then check whether or not increasing a row of length m_{a-1}, m_a, m_{a+1} changes the number of boxes in the first r columns (in each of the Young diagrams respectively). This amounts to comparing m_{a-1}, m_a, m_{a+1} with r. Because of the condition $m_{a-1} \leq m_a \leq m_{a+1}$, we only need to consider where r falls in this chain of inequalities. For example, if $m_{a-1} < r \leq m_a \leq m_{a+1}$, then the number of boxes in the first r columns of $\nu^{(a-1)}$ increases by one. However, the number of boxes in the first r columns of $\nu^{(a)}, \nu^{(a+1)}$ do not. So the change in contribution for $\nu^{(a-1)}$ is 1 and the change in contribution for $\nu^{(a)}, \nu^{(a+1)}$ is zero. Hence, we conclude that the vacancy number $P_r^{(a)}(\nu)$ increases by one.

Example 2.4.7. Consider the configurations displayed in 2.2 and 2.3. The chain of singular rows is $m_0 = 0 \leq m_1 = 2 \leq m_2 = 2$. Say we want to track how the vacancy number of rows of length one change in $\nu^{(1)}$. If rows of length m_0, m_1, m_2 are increased in $\nu^{(0)}, \nu^{(1)}, \nu^{(2)}$ by one, then the number of boxes in the first column do not change except for in $\nu^{(0)}$, where it increases by one. Hence, we expect the vacancy number on rows of length one to increase by one, which is exactly what we see.

Using the above discussion we note that we can read off changes to the vacancy numbers of the configuration by looking at the columns of the chain table.

Lemma 2.4.8. Let X be a rigged configuration with corresponding word $w = w_1...w_n$. Fix r and i. Then,

$$P_r^{(i)}(X) = \sum_{j=1}^{n} \begin{cases} 1 & \text{if } \mathcal{M}(w)_{i-1,j} < r \leq \mathcal{M}(w)_{i,j} \\ -1 & \text{if } \mathcal{M}(w)_{i,j} < r \leq \mathcal{M}(w)_{i+1,j} \\ 0 & \text{otherwise} \end{cases}$$

Proof. Follows immediately from the definition of vacancy numbers and chain tables. □

The main application of the above lemma will be to compute the differences in vacancy numbers between two rigged configurations from their chain tables. It will also provide us a paradigm for proving results involving rigged configurations. In particular, it allows us to track how the coriggings of a rigged configuration change at each step of the KKR bijection. As we will see in the proofs of this section, being able to determine which rows are singular, and which rows cannot be singular (i.e have positive corigging) is an indispensable asset.

Let us make some preliminary observations about the chain table. Consider Examples 2.4.3 and 2.4.2. When the length of consecutive columns weakly decrease, the rows restricted to those columns have increasing entries. This no surprise, given how the KKR bijection selects singular rows. Although there is no non-trivial occurence of the following phenomenon in Example 2.4.3 or 2.4.2, when the length of consecutive columns increase, the rows when restricted to these columns have decreasing entries. That is, the entries weakly decrease left to right across columns with increasing length. However, it is infact true that the entries weakly decrease in the south-east direction. Note that this is a stronger statement, as entries of the chain table weakly increase going down the column (see Remark 2.4.4). When we have a standard tableaux, the columns of the chain table are indexed by the corresponding Yamanouchi word, the letters of which also give the lengths of the columns. The above dicussion is the subject of our first result understanding of the interaction between the tableau descent structure and the chain table.

Lemma 2.4.9 (Chain Lemma)**.** Let $w = w_1...w_n$ be a Yamanouchi word and $1 \leq i < n$.

(i) If $w_i < w_{i+1}$ then, $\mathcal{M}(w)_{j,i} \geq \mathcal{M}(w)_{j+1,i+1}$ for $j < w_i$.

(ii) If $w_i \geq w_{i+1}$ then, $\mathcal{M}(w)_{j,i} > \mathcal{M}(w)_{j,i+1}$ for $j < w_{i+1}$.

Proof. First suppose $w_i < w_{i+1}$, and consider the Yamanouchi word $w' = w_1...w_i$. Let $X = (\nu, J)$ be the corresponding rigged configuration. No rows in $X^{(w_i)}$ of size greater than $\mathcal{M}(w)_{w_i-1,i}$ are singular. So we deduce that $\mathcal{M}(w)_{w_i,i+1} \leq \mathcal{M}(w)_{w_i-1,i}$. Suppose $\mathcal{M}(w)_{j+1,i+1} \leq \mathcal{M}(w)_{j,i}$. Note that in $X^{(j)}$, no rows whose length belongs to the interval $I = (\mathcal{M}(w)_{j-1,i}, \mathcal{M}(w)_{j,i}]$ are singular. By the definition of the KKR bijection, $\mathcal{M}(w)_{j,i+1}$ is the length largest singular row in $X^{(j)}$ which is at most $\mathcal{M}(w)_{j+1,i+1}$. Then, $\mathcal{M}(w)_{j,i+1} \leq \mathcal{M}(w)_{j+1,i+1} \leq \mathcal{M}(w)_{j,i}$. Then, because $\mathcal{M}(w)_{j,i+1} \notin I$, we must have $\mathcal{M}(w)_{j,i+1} \leq \mathcal{M}(w)_{j-1,i}$, which proves (i). We note that (ii) follows immediately from the definition of the KKR bijection. □

As a consequence of the chain lemma, we get our first basic result for understanding the structure of $\nu^{(1)}$ in terms of the descents of a Yamanouchi word.

Corollary 2.4.10. Let $w = w_1...w_n$ be a Yamanouchi word. Suppose $1 \leq i < n$. If $w_i = 1$ then $\mathcal{M}(w)_{1,i} = \epsilon$. Otherwise, $\mathcal{M}(w)_{1,i} > 0 \iff w_i \geq w_{i+1}$.

In other words, a new row in $\nu^{(1)}$ is created at step k if and only if $w_{k-1} < w_k$.

Remark 2.4.11. We recover the result [7, Corollary 4.7(v)] that the number of strict ascents in the Yamanouchi word $w_1...w_n$ determines the length of the first column in $\nu^{(1)}$.

Remark 2.4.12. Corollary 2.4.10 tells us that for a subword to be connected, it must be a weakly decreasing sequence.

We can compactify the second statement of the Lemma 2.4.9 by writing

$$\mathrm{Col}_i(\mathcal{M}(w)) > \mathrm{Col}_{i+1}(\mathcal{M}(w))$$

where the comparison is entry-wise whenever it makes sense (excluding row 0, because row 0 of a chain table corresponding to a standard tableau is identically 0).

Example 2.4.13. We have the comparison

$$\begin{bmatrix} 0 \\ 2 \\ 2 \\ 4 \\ \epsilon \\ \epsilon \\ \epsilon \end{bmatrix} > \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ \epsilon \end{bmatrix}.$$

Note that we ignore the entries in row 0. Moreover, we note that the column on the left hand side is still greater despite having less entries. This is due to the fact that when comparing entries that both columns have, the one of the left is greater pointwise.

We can interpret the inequality in (ii) as a "pointwise" inequality, and the diagonal inequality in (i) as a "shifted" inequality. The following lemma tells us that if the chain of singular rows determined by a step of the KKR bijection is "pointwise" less than an existing chain of singular rows in the rigged configuration, then instead of satisfying just a pointwise inequality, it satisfies the stronger "shifted" inequality.

Lemma 2.4.14. Let $w = w_1...w_n$ be a Yamanouchi word and $\Phi(w) = X$. Suppose that there is a chain of singular rows $m_0^{(n+1)} \leq ... \leq m_\ell^{(n+1)}$ where each $m_i^{(n+1)}$ is the length of a singular row in $\nu^{(i)}$. We write C to be the column consisting of entries $m_0^{(n+1)}, ..., m_\ell^{(n+1)}$. Consider the Yamanouchi word $w' = w_1...w_nw_{n+1}$. If $\text{Col}_{n+1}(\mathcal{M}(w)) < C$, then in fact, $\mathcal{M}(w)_{i,n+1} < m_{i-1}^{(n+1)}$ for $i > 1$.

Proof. If $\mathcal{M}(w)_{j,n+1} \geq C_{j-1} = m_{j-1}^{(n+1)}$ for some j, then because $\mathcal{M}(w)_{j-1,n+1}$ is the length of the largest singular row in $X^{(j-1)}$ of length at most $\mathcal{M}(w)_{j,n+1}$ it follows that $\mathcal{M}(w)_{j-1,n+1} \geq m_{j-1}^{(n+1)}$ contradicting $\text{Col}_{n+1}(\mathcal{M}(w)) < C$. □

One thing we would like to be able to know is when certain steps of the KKR bijection are "independent" of some of the steps which came before. That is, we would like to know when the selection of singular rows of lengths $m_1 \leq ... \leq m_r$ does not depend on the previous steps of the KKR bijection. For example, suppose there are two Yamanouchi words $w = w_1...w_ny$ and $w' = w_1...w_nxy$. If the columns of the chain table corresponding to y: $\text{Col}_{n+1}(\mathcal{M}(w))$ and $\text{Col}_{n+2}(\mathcal{M}(w'))$ are equal, then in some sense, the $(n+2)$nd step of the KKR bijection when computing $\Phi(w')$ is "independent" of the $(n+1)$st step (the step corresponding to adding x).

This question is of general interest to one trying to impose control over the KKR bijection. However, it turns out that studying this question will also aid us in our main goal of finding sufficient conditions for connectedness.

To answer this question, it is important to first detect when a chain of singular rows with lengths $m_1 \leq ... \leq m_r$ remain singular after a step of the KKR bijection. The diagonal shifted inequalities provided by Lemma 2.4.14 along with tracking the vacancy numbers with Lemma 2.4.8 prove the following corollary, which says that if we have a chain of singular rows C, and the KKR bijection selects "sufficiently small" chains of singular rows, then C will persist through those steps of the KKR bijection.

Corollary 2.4.15. Let $w = w_1 ... w_n$ be a Yamanouchi word. Let $w' = w_1...w_n x$ and define the chain $C = \mathrm{Col}_{n+1}(\mathcal{M}(w'))$. Let $\tilde{w} = w_1...w_n w_{n+1}...w_{n+k}$. If each $\mathrm{Col}_{n+i}(\mathcal{M}(\tilde{w})) < C$ then the rigged configuration corresponding to $\tilde{w}$ contains the chain C.

Now, we are equipped to give an answer to the posed question regarding which steps of the KKR bijection are independent of each other. However, there is a slight catch. We answer the question assuming some level of control of what happens to the first partition. We will do this by assuming connectedness. This assumption is necessary because the shifted inequalities break down at the first partition. Indeed, we are unable to get a similarly strong bound on m_1 as we are able to get on m_i for $i > 1$.

This assumption turns out to not be too important for us. In particular, if we find recursive conditions for connectedness, then the assumption is mostly given to us for free by recursion.

Lemma 2.4.16. Consider the Yamanouchi word $w = w_1...w_n u_1...u_k$ and $w' = w_1...w_n x u_1...u_k$. Suppose $u_1...u_k$ is connected with respect to w. If $\mathrm{Col}_{n+1}(w') > \mathrm{Col}_{n+i}(w)$ for all $1 \le i \le k$ then each $\mathrm{Col}_{n+i}(w) = \mathrm{Col}_{n+i+1}(w')$.

Proof. Define $m_0 \le m_1 \le m_2 \le ...$ where $m_i = \mathcal{M}(w')_{i,n+1}$ for $x > i \ge 1$, and $m_i := \infty$ if $i \ge x$.

We show that if X, Y are the rigged configurations corresponding to w and w' then

(1) if $k \ge 1$ then $\mathcal{M}(w)_{i,n+k} = \mathcal{M}(w')_{i,n+k+1}$

(2) $\tilde{X}^{(i)}_{\le m_{i-1}} = \tilde{Y}^{(i)}_{\le m_{i-1}}$ for all $x \ge i > 1$

(3) $\tilde{X}^{(i)} = \tilde{Y}^{(i)}$ for $i > x$

(4) $X^{(i)}_{(m_{i-1}, m_i)} = Y^{(i)}_{(m_{i-1}, m_i)}$ for $x \ge i > 1$.

(5) Suppose $x \ge i \ge 1$, and $m_{i-1} < r \le m_i$. Then, $P^{(i)}_r(Y) \ge P^{(i)}_r(X)$.

Proceed by induction on k. Note that the above conditions hold when $k = 0$ (so that $u_1...u_k$ is the empty string). Suppose the result holds for some $k - 1 \ge 0$.

First suppose $u_k \le x$. By Corollary 2.4.15, the rigged configuration X contains the chain $m_0 \le m_1 \le ... \le m_{x-1}$. In particular, the rigged partition $X^{(u_k - 1)}$ contains a singular string of length $m_{u_k - 1}$. But $\mathcal{M}(w)_{u_k - 1, n+k}$ is the largest singular string in $X^{(u_k-1)}$ which must be at least m_{u_k-1}. This implies that $\mathrm{Col}_{n+k}(w) \not< \mathrm{Col}_{n+1}(w') = (m_0, m_1, ...)$ contradicting the hypothesis of the lemma. Therefore $u_k > x$.

Let X', Y' be the rigged configurations corresponding to $w_1...w_n u_1...u_{k-1}$ and $w_1...w_n x u_1...u_{k-1}$. Define $b_i := \mathcal{M}(w)_{i,n+k}$ and $b'_i = \mathcal{M}(w')_{i,n+k+1}$. We wish to show that each $b_i = b'_i$. It is clear that $b_i = b'_i$ for $i > x$ from the fact that $\tilde{X'}^{(i)} = \tilde{Y'}^{(i)}$. Also by Lemma 2.4.14 we know $b_x < m_{x-1}$.

We have $b_{x+1} = b'_{x+1} < m_x$. So by (4) and (5), if there exists a singular row in $\tilde{Y'}^{(x)}$ with length $m_{x-1} < r < b'_{x+1} = b_{x+1} < m_x$ then there is also a singular row of length r in $\tilde{X'}^{(x)}$. However, $b_x < m_{x-1} < r < b_{x+1}$ which contradicts the consturction of the KKR bijection as r is larger than b_x. Hence, there are no singular rows in $\tilde{Y'}^{(x)}$ whose length lies in the interval (m_{x-1}, m_x). Moreover, b'_x is defined as the largest singular row in $\tilde{Y'}^{(x)}$ with $b'_x \le b'_{x+1} = b_{x+1} < m_x$. It follows that $b'_x \le m_{x-1}$.

Then because $\tilde{X'}^{(x)}_{\leq m_{x-1}} = \tilde{Y'}^{(y)}_{\leq m_{x-1}}$ we also obtain $b_x = b'_x$. Now suppose that $b_{y+1} = b'_{y+1}$ for some $1 < y+1 \leq x$. By Corollary 2.4.15, the rigged configuration X' contains the chain of singular rows of lengths $m_0 \leq m_1 \leq ... \leq m_{x-1}$. By the definition of the KKR bijection, b'_y is the largest singular row in $\tilde{Y'}^{(y)}_{\leq b'_{y+1}}$.

Suppose that $\tilde{Y'}^{(y)}$ contains a singular row $(r, 0)$ with $m_{y-1} < r \leq \min(m_y - 1, b_{y+1})$. Let $\alpha = P_r^{(y)}(Y')$ so that $(r, \alpha) \in Y'^{(y)}$. By inductive hypothesis (4), we have $(r, \alpha) \in X'^{(y)}$. By inductive hypothesis (5), it follows that (r, α) is also a singular row in $X'^{(y)}$. But then, as $r \leq b_{y+1}$ it follows that $b_y \geq r > m_{y-1}$. This contradicts Lemma 2.4.14.

We know $b_{y+1} = b'_{y+1} < m_y$ by Lemma 2.4.14. Note that b'_y is the largest singular row of $\tilde{Y'}^{(y)}$ with $b'_y \leq b'_{y+1} = b_{y+1} < m_y$. Hence, from the above discussion, b'_y is at most m_{y-1}. So, b'_y is the largest singular row in $\tilde{Y'}^{(y)}_{\leq m_{y-1}} = \tilde{X'}^{(y)}_{\leq m_{y-1}}$. Note that by Lemma 2.4.14 we have $b_y < m_{y-1}$. Thus, the largest singular row in $\tilde{X'}^{(y)}$ at most b_{y+1} is the largest singular row in $\tilde{X'}^{(y)}_{\leq m_{y-1}} = \tilde{Y'}^{(y)}_{\leq m_{y-1}}$. It follows that $b_y = b'_y$. Hence we have shown that $b_y = b'_y$ for $y > 1$.

A similar argument to above shows the number of singular rows in $\tilde{Y'}^{(1)}$ with length in the interval $m_0 = 0 < r \leq b'_2 = b_2$ is at most the number entries of the chain table satisfying $\mathcal{M}(w)_{1,n+t} = \mathcal{M}(w')_{1,n+t+1} = m_0 = 0$ for $t \in [1, k-1]$. That is, it is at most the number of new boxes added to the first partition during steps appending the letters $u_1, ..., u_{k-1}$. Because $u_1...u_{k-1}$ is connected, there is exactly one such singular row, of length $k-1$. It follows that $b'_1 = k - 1 = b_1$.

The conditions (2) and (4) follow for X, Y because they follow for X', Y', and each $b_i < m_{i-1}$ for $1 < i \leq x$. Condition (3) follows from the fact it holds for X', Y' and because $b_i = b'_i$ for $i \geq x$. The last condition (5) follows from the fact it holds for X', Y' and each $b_i = b'_i$ for $1 < i \leq x$.

□

Using all of the results we have found this section, we are able to give sufficient conditions for connectedness. The conditions are given in the two following definitions, which can be thought of as a generalization of matching opening and closing braces (where opening braces are "small numbers" and closing braces are "big numbers")

Definition 2.4.17. Consider the word $u_1...u_kv_1...v_k$ where $u_k < v_1$, $u_1 \geq ... \geq u_k$ and $v_1 \geq ... \geq v_k$. We say that $u_1...u_kv_1...v_k$ satisfies the *perfect pairing condition* if each $u_i < v_i$

Definition 2.4.18. The word $u_1...u_kv_1...v_t$ where $u_k < v_1$, $u_1 \geq ... \geq u_k$ and $v_1 \geq ... \geq v_t$ satisfies the *pairing condition* if there exists a subsequence $i_1 < i_2 < ... < i_t$ such that $u_{i_1}...u_{i_t}v_1...v_t$ satisfies the perfect pairing condition (of length t).

Example 2.4.19. 322544 satisfies the perfect pairing condition of length 3 because

$$3 < 5$$
$$2 < 4$$
$$2 < 4$$

432253 satisfies the pairing condition of length 2 because the second descent sequence 53 is of length 2, and we can pick the length two subsequence of the first descent sequence 42 such that

4253 satisfies the perfect pairing condition of length 2:

$$4 < 5$$
$$3 < \times$$
$$2 < 3$$
$$2 < \times$$

Remark 2.4.20. Note that if $u_1 ... u_k v_1 ... v_t$ satisfies the pairing condition of length t, then either

(i) $u_1 ... u_{k-1} v_1 ... v_t$ satisfies the pairing condition of length t.

(ii) $u_1 ... u_{k-1} v_1 ... v_{t-1}$ satisfies the pairing condition of length $t - 1$.

Thus, when inducting on descent sequences satisfying the pairing condition, one should induct on the length of the *first* descent sequence. For example, the inductive hypothesis should be of the form, "for some k, if the descent sequences $u_1 ... u_k v_1 ... v_t$ satisfy the pairing condition, then $v_1 ... v_t$ is connected". If we have case (i), the result follows for free. Hence, we only need to worry about case (ii). In effect, this means that an argument for the perfect pairing condition extends easily to an argument for the pairing condition.

We arrive at the main theorem of this section which says that the pairing condition guarantees connectedness.

Theorem 2.4.21 (Connectedness Theorem)**.** Let $w = w_1 ... w_n u_1 ... u_k v_1 ... v_t$ with $k \geq t$ be a Yamanouchi word and suppose $u_1 ... u_k v_1 ... v_t$ satisfies the pairing condition. Then, $v_1 ... v_t$ is connected and $\mathrm{Col}_{n+k+i}(w) \leq \mathrm{Col}_{n+i}(w)$ for each i. In particular, $\mathrm{Col}_{n+k+i}(w) < \mathrm{Col}_{n+k}(w)$ by Lemma 2.4.9.

Proof. Proceed by induction on k. If $k = t = 1$ the result follows by Lemma 2.4.9. Now suppose the result holds for some k and consider $w = w_1 ... w_n u_1 ... u_k v_1 ... v_t$ where $u_1 ... u_k v_1 ... v_t$ satisfies the pairing condition (namely, $k \geq t$). Suppose $x \leq u_k$. Notice that by Lemma 2.4.16 (and induction), $w' = w_1 ... w_n u_1 ... u_k x v_1 ... v_t$ has $v_1 ... v_t$ connected. Moreover, by the Lemma 2.4.9, $\mathrm{Col}_{n+k+1}(w') > \mathrm{Col}_{n+k}(w')$. Also by Lemma 2.4.16, $\mathrm{Col}_{n+k+1+i}(w') = \mathrm{Col}_{n+k+i}(w)$. So, it is also clear that $v_1 ... v_t$ is connected.

Now suppose $y \leq v_t$ with $x < y$ and consider $w'' = w_1 ... w_n u_1 ... u_k x v_1 ... v_t y$. We wish to show $\mathrm{Col}_{n+k+1}(w'') > \mathrm{Col}_{n+k+2+t}(w'')$ and that $\mathcal{M}(w'')_{1,n+k+2+t} = t + 1$. Let $b = n + k + 2 + t$ and define $g_t := \mathcal{M}(w'')_{t,n+k+1}$ if $\mathcal{M}(w'')_{t,n+k+1} \neq \epsilon$ and $g_t := \infty$ otherwise. Define $h_t := \mathcal{M}(w'')_{t,b}$. If $\mathrm{Col}_b(\mathcal{M}(w'')) \not\leq \mathrm{Col}_{n+k+1}(w)$ then consider the largest index i_0 such that $h_{i_0} > g_{i_0}$.

Note that $i_0 < x < y$ (just by the definition of how we defined the comparison between two columns), so $i_0 + 1 \leq x < y$. In particular, $h_{i_0+1} \neq \epsilon$ and $h_{i_0+1} \geq h_{i_0} \geq g_{i_0}$.

Then if Y is the rigged configuration corresponding to $\overline{w} := w_1 ... w_n u_1 ... u_k x v_1 ... v_t$ and X is the rigged configuration corresponding to $w_1 ... w_n u_1 ... u_k v_1 ... v_t$ then $Y^{(i_0+1)}$ contains a singular row of length at least $h_{i_0+1} \geq g_{i_0}$.

Notably, by maximality of i_0, $h_{i_0+1} \leq g_{i_0+1}$ (note this holds even when $i_0 + 1 = x$ due to how we define empty entries of the chain table).

Notice that because each $\mathrm{Col}_{n+k+i}(w) \leq \mathrm{Col}_{n+i}(w) < \mathrm{Col}_{n+k}(w)$, we have $\mathrm{Col}_{n+k+i+1}(w') = \mathrm{Col}_{n+k+i}(w)$ by Lemma 2.4.16. The rigged partitions $\Phi(w_1 ... w_n u_1 ... u_k x)^{(i)}$ is contained in the rigged partition $\Phi(w_1 ... w_n u_1 ... u_k)^{(i)}_{(g_{i-1}, g_i]}$. Indeed, it has exactly one less row of length g_i. Moreover, the fact that each $\mathcal{M}(\overline{w})_{i,n+k+j+1} = \mathcal{M}(w)_{i,n+k+j} < g_{i-1}$ by Lemma 2.4.14 and the inductive hypothesis $\mathrm{Col}_{n+k+i}(w) \leq \mathrm{Col}_{n+i}(w) < \mathrm{Col}_{n+k+1}(\overline{w})$ (where the last strict inequality follows from the chain lemma) shows that in fact $Y'^{(i)}_{(g_{i-1},g_i]} \subseteq X'^{(i)}_{(g_{i-1},g_i]}$ for $i \leq x$.

The fact that we have the equality $\mathcal{M}(\overline{w})_{i,n+k+j+1} = \mathcal{M}(w)_{i,n+k+j}$ implies that each $P_r^{(i)}(Y') = P_r^{(i)}(X') + 1$ for $i < x$ and $m_{i-1} < r < m_i$ by Lemma 2.4.8.

It follows that there cannot be any singular rows in $Y'^{(i_0+1)}_{(g_{i_0},g_{i_0+1}]}$, as it would imply the existence of a row in $X'^{(i_0+1)}_{(g_{i_0},g_{i_0+1}]}$ whose rigging exceeds its vacancy number. This contradicts the fact that $g_{i_0} \leq h_{i_0} \leq h_{i_0+1} < g_{i_0+1}$.

Hence, it follows that each $h_i \leq g_i$. It remains to show that $v_1...v_t y$ is connected.

Note that $Y'^{(1)}_{(t,g_1]} \subseteq X'^{(1)}_{(t,g_1]}$ (which differ by exactly one row of length g_1). This follows from the fact that $\mathcal{M}(w)_{1,n+k+i} = \mathcal{M}(w')_{1,n+k+i+1} = i-1 < t$ for $1 < i \leq t$, and the fact that $\mathcal{M}(w')_{1,n+k+1} = g_1$ (so the first row of the chain tables for w, w' past column $n+k$ and $n+k+1$ respectively do not contain any entries in $(t, g_1]$ with the exception of the chain table of w' having exactly one entry g_1). Also using Lemma 2.4.8, we have $P_r^{(1)}(Y') = P_r^{(1)}(X') + 1$ for $t-2 < r \leq g_1$. Hence, $Y'^{(1)}$ contains no singular rows whose length lies in $(t, g_1]$.

We wish to show that $h_1 = t$. By the chain lemma, we must have $h_1 > \mathcal{M}(\overline{w})_{n+k+t} = t-1$. Thus, because $h_1 \notin (t, g_1]$, either $h_1 = t$ or $h_1 > g_1$. However, we've shown that each $h_i \leq g_i$. Thus, we must have $h_1 = t$ as required.

□

Example 2.4.22. Consider the Yamanouchi word $w = 111232432$. We see that 3243 (which is the subword corresponding to letters $5, 6, 7, 8$) satisfies the pairing condition of length 2. So we expect that the columns corresponding to the letters $4, 3$ (columns 7 and 8) have consecutive entries $0, 1$ in the chain table. We see that this is indeed the case in Example 2.4.3. We moreover see that the pairing condition is only sufficient, not necessary. Indeed, the descent sequence preceeding 432 is 32. However, 32432 does not satisfy the pairing condition of length 3. Despite this, the subword 432 (corresponding to columns $7, 8, 9$) is connected, which we can also see by Example 2.4.3.

We introduce another result on connectedness, which is that the last descent sequence is connected of length equal to the smallest row of $\nu^{(1)}$ if and only if one of $X^{(1)}$'s smallest rows is singular. This proposition will be needed later in Section 2.8.

Proposition 2.4.23. Let $X = (\nu, J)$ be a rigged configuration and $k = \min_{1 \leq i \leq \ell(\nu^{(1)})} \nu_i^{(1)}$. Write $w = w_1...w_n = \Phi^{-1}(X)$. Then $X^{(1)}$ has a singular row of length k if and only if $w_{n-k}w_{n-k-1}...w_n$ is connected (and $w_{n-k-1}...w_n$ is not connected).

Proof. If $w_{n-k}...w_n$ is connected, then $X^{(1)}$ has a singular row of length k because modified rows in the KKR bijection remain singular. For the reverse implication, proceed by induction on k. If $k = 1$, the result follows from Corollary 2.4.10. Suppose all rigged configurations who contains a smallest row (of length $k-1$) that is singular satisfies that the subword induced by the last $k-1$ letters of its corresponding Yamanouchi word is connected. Consider the chain of rows $m_0 = 0 \leq m_1 \leq ... \leq m_{w_n - 1}$ when computing the inverse KKR bijection. By definition, m_1 is the smallest singular row in $X^{(1)}$ of length at least $0 + 1 = 1$, hence $m_1 = k$. So, $\mathcal{M}(w)_{1,n} = k-1$. Note, the length of the smallest row in $\Phi^{-1}(w_1...w_{n-1})^{(1)}$ is $k-1$, which is also singular by the definition of the inverse KKR bijection. By induction, $w_{n-k}...w_{n-1}$ is connected so that $\mathcal{M}(w)_{1,n-k+i} = \mathcal{M}(w_1...w_{n-1})_{1,n-k+i} = i$ for $0 \leq i < k-1$. Then because $\mathcal{M}(w)_{1,n} = k-1$, it follows that $w_{n-k}...w_n$ is connected. □

We end this section with two observations to be used later (in Section 2.7).

Lemma 2.4.24. Suppose that the words $w' = w_1...w_x$ and $v' = v_1...v_y$ with rigged configurations X', Y' (respectively) have the same proper configuration and satisfy $\tilde{X'}^{(i_0)} = \tilde{Y'}^{(i_0)}$ for some $i_0 > 1$. Let $w = w_1...w_x...w_n$ and $v = v_1...v_y...v_m$ with $n - x = m - y$ and suppose that $\mathcal{M}(w)_{i,x+j}(w) = \mathcal{M}(w)_{i,y+j}(w)$ for all i and all $j \geq 0$. Let X, Y be the rigged configurations corresponding to w and v respectively. Then, $\tilde{X}^{(i_0)} = \tilde{Y}^{(i_0)}$.

Proof. The Lemma follows from the fact that the vacancy numbers of rigged partitions with index $i > 1$ are determined completely by the proper configuration, induction, and the way singular rows are chosen at each step of the KKR bijection. □

Lemma 2.4.25. Suppose that the words $w' = w_1...w_x$ and $v' = v_1...v_y$ have corresponding rigged configurations X and Y which satisfy $X'^{(1)} = Y'^{(1)} = Z$. Now let $w = w_1...w_x u_1...u_k$ and $v = v_1...v_y \alpha_1...\alpha_k$ where $u_1...u_k$ and $\alpha_1...\alpha_k$ are connected in w and v respectively. Let X, Y be the rigged configurations corresponding to w and v respectively. Then, $X^{(1)} = Z \cup \{(k, r)\}$ and $Y^{(1)} = Z \cup \{(k, s)\}$ where r is the vacancy number of a row of length k in $X^{(1)}$ and s is the vacancy number of a row of length k in $Y^{(1)}$. In particular, $X^{(1)} = (Y^{(1)} \setminus \{(k, s)\}) \cup \{(k, r)\}$

Proof. The Lemma follows from the definition of connectedness, the fact that if a row's length does not change via a step of the KKR bijection then neither does its rigging, and the fact that the most recently extended row via the KKR bijection is singular. □

2.5 What is the combinatorial R?

For those familiar with crystals, our definition of crystals will require them to be of type A and seminormal. All morphisms will be required to be strict. Also, we follow the opposite tensor product convention to Kashiwara.

We give a brief review of crystals as well as how they relate to tableau combinatorics. See [2] for a more complete reference. For a beginner friendly reference on affine crystals and the combinatorial R, see [18]. The goal of this section is to introduce what the combinatorial R is and explain why it has its characterization in terms of Schensted insertion. We will also explain the typical way to represent standard and semistandard tableaux as paths or (classically) highest weight elements of tensor products of Kirillov-Reshetikhin crystals. After this section, we will primarily be formulating our results in terms of paths rather than tableaux.

The short version of the story is that the set of semistandard tableaux of a given shape, B_λ, is an example of a combinatorially flavoured algebraic object called a crystal. One can define tensor products of crystals, and it turns out the tensor product is commutative. That is, $B_\lambda \otimes B_\mu$ and $B_\mu \otimes B_\lambda$ are isomorphic. One can characterize whether two elements $c \otimes d, d' \otimes c'$ are related by an isomorphism between $B_\lambda \otimes B_\mu$ and $B_\mu \otimes B_\lambda$ by comparing whether the rectifcation of c and d is the same as the rectification of d' and c'.

Much of the combinatorics of tableaux discussed here existed before the discovery of Kashiwara crystals in [5]. However, they will be discussed from the perspective of crystals.

2.5.1 Preliminaries

First let us discuss what a crystal is. A *crystal* of type A_{m-1} is a set C equipped with two operations for each $1 \le i < m$,

$$e_i : C \to C \sqcup \{0\} \tag{2.4}$$

$$f_i : C \to C \sqcup \{0\} \tag{2.5}$$

as well as a weight function $\mathrm{wt} : C \to \mathbb{Z}^m$. The 0 element is a "trivial element" which is used as a certificate that the result of e_i or f_i is "undefined". In particular, it "does not" belong to the crystal. The crystal operators cannot be applied to them. In effect, the 0 element serves as a way to effectively allow the domain to be entirely C, even if e_i, f_i are not defined on some elements.

The operations e_i, f_i are called *raising* and *lowering* operators respectively. They satisfy the conditions:

$$\mathrm{wt}(e_i(x)) = \mathrm{wt}(x) + \alpha_i \tag{2.6}$$

$$\mathrm{wt}(f_i(x)) = \mathrm{wt}(x) - \alpha_i \tag{2.7}$$

where $\alpha_i = e^i - e^{i+1}$, defining e^i to be the ith standard basis vector.

Remark 2.5.1. More generally, a crystal has a root system associated with it (in our case, we use the "type A_{m-1} root system"). In the general case, $\mathbb{Z}^m$ is replaced with the weight lattice, and the α_i are the positive simple roots of the root system. The underlying root system also induces a Weyl group, which is important for many crystal constructions. In type A_{m-1}, the Weyl group is just the symmetric group S_m.

The raising and lowering operators must also satisfy $e_i(f_i(x)) = x$ if $f_i(x) \neq 0$ and $f_i(e_i(x)) = x$ if $e_i(x) \neq 0$. A crystal also comes equipped with functions φ_i, ϵ_i which measure how many times a raising/falling operator can be applied. That is,

$$\varphi_i(x) = \max(k : f_i^k(x) \neq 0) \tag{2.8}$$

$$\epsilon_i(x) = \max(k : e_i^k(x) \neq 0). \tag{2.9}$$

An element $x \in C$ such that $e_i(x) = 0$ for all i is called *highest weight*. An element $x \in C$ with $f_i(x) = 0$ for all i is called *lowest weight*. A crystal is *connected* if there is a sequence of raising and lowering operators which, when applied to x, yields y for any x, y in the crystal. A crystal can be broken up into disjoint connected subcrystals called components or connected components. A *morphism* of crystals C, D is a map from C to $D \sqcup \{0\}$ which preserves the operations $e_i, f_i, \varphi_i, \epsilon_i, \mathrm{wt}$. An *isomorphism* is a morphism such that the extended map $C \sqcup \{0\} \to D \sqcup \{0\}$ (with $0 \mapsto 0$) is a bijection. The precise definition of a morphism and isomorphism becomes tricky due to the trivial element. Adding the trivial elements to the domain and codomain of a morphism somewhat obfuscates the given definition, but is necessary for a precise one. Most of the time, one thinks of a morphism as a map $C \to D$ and only worries about the trivial element when it is immediately necessary.

We wish to eventually define the crystal B_λ. To do so, we must first define the standard crystal, and the $n-$fold tensor product of the standard crystal with itself.

Example 2.5.2. We can define the standard A_{m-1} crystal, B. As a set, B are just the numbers $\{1, ..., m\}$. The raising operator $e_i(x) = x-1$ if $x = i+1$ and $e_i(x) = 0$ otherwise. Also, $f_i(x) = x+1$

if $x = i$ and $f_i(x) = 0$ otherwise. In particular, $\epsilon_i(i+1) = 1$ and $\epsilon_i(j) = 0$ otherwise. Similarly, $\varphi_i(i) = 1$ and $\varphi_i(j) = 0$ otherwise. The elements of B are typically written inscribed in boxes (i.e $\boxed{1}, \boxed{2}, ...$). However, for notational reasons we will often omit these boxes where it is understood.

We can define the tensor product of two crystals C and D as follows. As a set, $C \otimes D$ consists of elements of the form $x \otimes y$ with $x \in C, y \in D$. The weight $\mathrm{wt}(x \otimes y) = \mathrm{wt}(x) + \mathrm{wt}(y)$. The operator $e_i(x \otimes y)$ is $e_i(x) \otimes y$ if $\epsilon_i(x) > \varphi_i(y)$ and $x \otimes e_i(y)$ otherwise. The operator $f_i(x \otimes y)$ is $f_i(x) \otimes y$ if $\epsilon_i(x) \geq \varphi_i(y)$ and $x \otimes f_i(y)$ otherwise. We take $0 \otimes y = x \otimes 0$ to be the trivial element. The tensor product is known to be associative.

Remark 2.5.3. We can view elements of the $n-$fold tensor product B^n as words $w_1...w_n$ of length n with the identification

$$w_1...w_n \mapsto w_1 \otimes ... \otimes w_n \in B^n.$$

The *signature rule* (see [2, section 2.4]) is a combinatorial rule for the raising and lowering operators on tensor products of crystals, which we review here.

Algorithm 2.5.4 (Signature rule)**.** Let $u_1 \otimes ... \otimes u_k \in C_1 \otimes ... \otimes C_k$. Draw coloured brackets

$$)^{\varphi_i(u_1)}(^{\epsilon_i(u_1)})^{\varphi_i(u_2)}(^{\epsilon_i(u_2)}...)^{\varphi_i(u_k)}(^{\epsilon_i(u_k)}$$

where $)^{\varphi_i(u_j)}(^{\epsilon_i(u_j)}$ is given colour j. Consider the left-most unmatched left opening brace and say it has colour j. Then, $e_i(u_1 \otimes ... \otimes u_k) = u_1 \otimes ... \otimes u_{j-1} \otimes e_i(u_j) \otimes u_j \otimes ... \otimes u_k$. If the right-most unmatched closing brace has colour j then, $f_i(u_1 \otimes ... \otimes u_k) = u_1 \otimes ... \otimes u_{j-1} \otimes f_i(u_j) \otimes u_{j+1} \otimes ... \otimes u_k$.

Notice that the signature rule agrees with the original description of the tensor product in the case where $k = 2$.

Example 2.5.5. In the particular case of B^n, using the expressions for ϵ, φ in Example 2.5.2, the signature rule becomes eaiser to understand. Indeed, to compute e_i, f_i of $w_1 \otimes ... \otimes w_n \in B^n$, only consider the letters $w_{j_1}, ..., w_{j_k}$ that are equal to either $i+1$ or i. Replace the $i+1$ with opening braces, and the i with closing braces. Then,

$$e_i(w_1 \otimes ... \otimes w_n) = w_1 \otimes ... \otimes w_{j_x - 1} \otimes e_i(w_{j_x}) \otimes w_{j_x+1} \otimes ... \otimes w_n$$

where w_{j_x} is the left-most unmatched opening brace and

$$f_i(w_1 \otimes ... \otimes w_n) = w_1 \otimes ... \otimes w_{j_y - 1} \otimes f_i(w_{j_y}) \otimes w_{j_y+1} \otimes ... \otimes w_n$$

where w_{j_y} is the right-most unmatched closing brace. We give an explicit computation below, where the red opening brace represents the left-most unmatched opening brace.

$$\begin{aligned}
& e_1(\boxed{3} \otimes \boxed{2} \otimes \boxed{2} \otimes \boxed{3} \otimes \boxed{3} \otimes \boxed{1} \otimes \boxed{1} \otimes \boxed{2} \otimes \boxed{2}) \\
& \to \boxed{\times} \otimes \boxed{(} \otimes \boxed{(} \otimes \boxed{\times} \otimes \boxed{\times} \otimes \boxed{)} \otimes \boxed{)} \otimes \boxed{\color{red}{(}} \otimes \boxed{(} \\
& \to \boxed{3} \otimes \boxed{2} \otimes \boxed{2} \otimes \boxed{3} \otimes \boxed{3} \otimes \boxed{1} \otimes \boxed{1} \otimes e_1(\boxed{2}) \otimes \boxed{2} \\
& \to \boxed{3} \otimes \boxed{2} \otimes \boxed{2} \otimes \boxed{3} \otimes \boxed{3} \otimes \boxed{1} \otimes \boxed{1} \otimes \boxed{1} \otimes \boxed{2}
\end{aligned}$$

Remark 2.5.6. Using the signature rule, we see that $w_1 ... w_n$ is Yamanouchi if and only if the element $w_n \otimes ... \otimes w_1 \in B^n$ is highest weight. Note in particular that it's the *reverse* Yamanouchi words that are highest weight.

Now we can define B_λ. We put a crystal structure on semistandard tableaux, where the raising/falling operators are induced by the raising/falling operators on their row-reading words.

Definition 2.5.7. Define the crystal B_λ of type A_{m-1} to have elements be semistandard tableau of shape λ and largest entry at most m. Take the row-reading word and realize it as an element $w \in B^n$. The words $e_i(w) \in B^n$ and $f_i(w) \in B^n$, if non-zero, are also row-reading words for a tableau of shape λ. We take $e_i(T), f_i(T)$ to be these tableaux respectively.

Example 2.5.8. Consider the semistandard tableau

$$\begin{array}{|c|c|c|c|}\hline 1 & 1 & 2 & 2 \\ \hline 2 & 2 & 3 & 3 \\ \hline 3 \\ \cline{1-1}\end{array}$$

The row-reading word is 322331122. By Example 2.5.5, we have

$$\begin{aligned} & e_1(\boxed{3}\otimes\boxed{2}\otimes\boxed{2}\otimes\boxed{3}\otimes\boxed{3}\otimes\boxed{1}\otimes\boxed{1}\otimes\boxed{2}\otimes\boxed{2}) \\ = & \boxed{3}\otimes\boxed{2}\otimes\boxed{2}\otimes\boxed{3}\otimes\boxed{3}\otimes\boxed{1}\otimes\boxed{1}\otimes\boxed{1}\otimes\boxed{2}. \end{aligned}$$

Identifying the result as a word, we obtain 322331112. The tableau corresponding to this reading word is

$$\begin{array}{|c|c|c|c|}\hline 1 & 1 & 1 & 2 \\ \hline 2 & 2 & 3 & 3 \\ \hline 3 \\ \cline{1-1}\end{array}$$

Hence,

$$e_1\left(\begin{array}{|c|c|c|c|}\hline 1 & 1 & 2 & 2 \\ \hline 2 & 2 & 3 & 3 \\ \hline 3 \\ \cline{1-1}\end{array}\right) = \begin{array}{|c|c|c|c|}\hline 1 & 1 & 1 & 2 \\ \hline 2 & 2 & 3 & 3 \\ \hline 3 \\ \cline{1-1}\end{array}$$

Remark 2.5.9. The standard crystal B, and the crystal $B_{(1)}$ are the same.

Remark 2.5.10. The row-reading word gives an embedding (that is, an injective morphism) $\mathrm{RR} : B_\lambda \to B^n$.

Remark 2.5.11. All crystals come with *crystal reflection operators* s_i, which induce an action of the Weyl group corresponding to the root system on the crystal [2, Theorem 11.14]. In type A_{m-1}, this means the s_i induce an action of S_m on the crystal. The operation s_i maps x to the unique element y such that $\epsilon_i(x) = \varphi_i(y)$ and $\varphi_i(x) = \epsilon_i(y)$. They are realized explicitly by either applying raising or falling operators until the desired element is achieved. The fact that the s_i indeed induce an action of the underlying Weyl group (or even in the special case of the symmetric group) is non-trivial. The operators s_i on B_λ are also called Lascoux-Schützenberger involutions.

Crystal reflection operators behave nicely with the KKR map as the following theorem demonstrates.

Theorem 2.5.12 ([8] Lemma 8.5)**.** Let $T, T' \in \mathrm{SSYT}(T)$ be two tableaux who are equivalent modulo the action of the symmetric group. Then the image of T, T' under the KKR map is the same.

Remark 2.5.13. The crystal B_λ (in type A_{m-1} of semistandard young tableaux of shape λ) has unique highest weight element, y_λ of weight λ which is the tableaux of shape and content both λ. These tableaux y_λ are called Yamanouchi tableaux. Any crystal automorphism of B_λ has to map $y_\lambda \mapsto y_\lambda$. It follows that the entire automorphism is determined, as it is connected and must preserve crystal operators. It moreover follows that any multiplicity free direct sum $\oplus_\lambda B_\lambda$ has a unique automorphsim.

2.5.2 Commutativity of the Tensor Product

In the previous subsection, we saw the tensor product construction of crystals. One might ask whether the tensor product of crystals is "commutative". Indeed, it is not immediately clear from the signature rule. There are varying levels of "commutativity" of the tensor product we may have. It could be the case that $C \otimes D$ and $D \otimes C$ are not necessarily isomorphic, in which case the tensor product is not commutative. We could have that $C \otimes D$ and $D \otimes C$ are isomorphic, but there are many different isomorphisms. Lastly, there may exist some canonical, or even unique isomorphism between $C \otimes D$ and $D \otimes C$.

It turns out, that $C \otimes D$ and $D \otimes C$ are indeed always isomorphic, but there can be many different isomorphisms. We will demonstrate in the specific case of B_λ that tensor products commute. The way we will do this is by showing that $B_\lambda \otimes B_\mu$ and $B_\mu \otimes B_\lambda$ decompose into the same smaller crystals (connected components) with the same multiplicity. This will follow by interpreting the "character" of a crystal as a Schur function.

The *character* of a crystal is defined as $\chi(C) = \sum_{c \in C} {x_1}^{\mathrm{wt}(c)_1} \ldots {x_m}^{\mathrm{wt}(c)_m}$. In particular, we note that

$$\chi(B_\lambda) = \sum_T {x_1}^{\alpha(T)_1} \ldots {x_m}^{\alpha(T)_m} = s_\lambda(x_1, \ldots, x_m).$$

Remark 2.5.14. It is a standard result in crystal theory [2, Proposition 2.17] that for any crystal C, the number of elements with weight μ and weight ν are the same modulo the action of the Weyl group. Hence, the polynomial $\chi(C)$ is invariant under the action of the Weyl group. In our type A_{m-1} case, the Weyl group is the symmetric group S_m. Hence, our above discussion shows that the character of a type A_{m-1} crystal is a symmetric function. In particular, $\chi(B_\lambda) = s_\lambda(x_1, \ldots, x_m)$ is a symmetric function. Therefore the Schur functions are symmetric functions, which proves the assertion made in Example 2.2.11.

It is a non-trivial fact that the family of crystals (B_λ) (which consists of the crystals B_λ ranging over all partitions) is closed under tensor product. That is, $B_\lambda \otimes B_\mu$ is the union of crystals of the form B_ν for some partitions ν (with multiplicities potentially greater than one). From the definition of tensor product we immediately see $\chi(B_\lambda \otimes B_\mu) = \chi(B_\lambda)\chi(B_\mu) = s_\lambda(x_1, \ldots, x_m)s_\mu(x_1, \ldots, x_m)$. Then, because (B_λ) is closed under tensor product, the number of copies of B_ν in $B_\lambda \otimes B_\mu$ is $[s_\nu](s_\lambda s_\mu) = c^\nu_{\lambda\mu}$ which are the Littlewood-Richardson coefficients.

Additionally, we note that the ring of symmetric functions is commutative. This implies that the decomposition of $B_\lambda \otimes B_\mu$ and $B_\mu \otimes B_\lambda$ into connected components is the same. Hence, $B_\lambda \otimes B_\mu$ and $B_\mu \otimes B_\lambda$ are isomorphic.

2.5.3 The Plactic Product and the Combinatorial R

Now that we know $B_\lambda \otimes B_\mu$ and $B_\mu \otimes B_\lambda$ are isomorphic, we would like to know what the isomorphisms look like.

Another consequence of the B_λ being closed under tensor product is that $B^n = B^n_{(1)}$ is a union of B_λ. In Definition 2.5.7, we defined B_λ in terms of their embedding into B^n. Because B^n is a union of B_λ, this shows that B^n can be entirely covered via these embeddings.

Definition 2.5.15. Let $T_1, ..., T_k$ be semistandard tableaux with largest entry at most m. Also, let $\sum_{i=1}^k |T_i| = n$. Let T_i be of shape $\lambda^{(i)}$. Each T_i has an embedding into $B^{|\lambda^{(i)}|}$ via the row-reading word, RR. In fact, the tensor product $B_{\lambda^{(1)}} \otimes ... \otimes B_{\lambda^{(k)}}$ naturally embeds into B^n via $\mathrm{RR} : T_1 \otimes ... \otimes T_k \mapsto \mathrm{RR}(T_1) \otimes ... \otimes \mathrm{RR}(T_k)$. Note that because B^n is a union of B_λ's, that the image of RR of the connected component containing $T_1 \otimes ... \otimes T_k$ is isomorphic to B_λ for some λ. This isomorphism is unique by Example 2.5.7. Let $T \in B_\lambda$ be the tableau which corresponds to $\mathrm{RR}(T_1) \otimes ... \otimes \mathrm{RR}(T_k)$ under this unique isomorphism. We call T the *plactic product* of $T_1, ..., T_k$ and write $T = T_1 \cdot T_2 \cdot ... \cdot T_k$. In the literature, the plactic product is known as *rectification*.

Remark 2.5.16. We note that because the tensor product of crystals is associative, the plactic product is also associative. Hence, we can define a monoid structure on semistandard tableaux called the *plactic monoid*.

Define

$$\pi_{\lambda^{(1)},...,\lambda^{(k)}} : \otimes_{i=1}^k B_{\lambda^{(i)}} \to \oplus_\lambda B_\lambda$$
$$\pi_{\lambda^{(1)},...,\lambda^{(k)}} : T_1 \otimes ... \otimes T_k \mapsto T_1 \cdot ... \cdot T_k.$$

Definition 2.5.15 implies π is an isomorphism onto its image when restricted to any connected component.

Remark 2.5.17. Consider a word $w_1...w_n$. By Remark 2.5.6, the reversed word $w_n...w_1$ is Yamanouchi if and only if the crystal element $w = w_n \otimes ... \otimes w_1 \in B^n$ is highest weight. The map $\pi_{(1),...,(1)}$ is an isomorphism onto its image when restricted to the component containing w. Hence, $\pi_{(1),...,(1)}(w)$ is also highest weight. This gives us the alternate characterization that a word $w_1...w_n$ is Yamanouchi if and only if $w_n \cdot ... \cdot w_1$ is a Yamanouchi tableaux. Indeed, each B_λ has a unique highest weight element, y_λ. Additionally, the weight of w is $\mathrm{wt}(w) = \mathrm{wt}(\pi_{(1),...,(1)}(w)) = \mathrm{wt}(y_\lambda) = \lambda$. Hence, $w_n \cdot ... \cdot w_1$ is the unique semistandard tableau with shape and content equal to the content of $w_1...w_n$.

As a consequence of Definition 2.5.15, we see that if we have an isomorphism $F : B_\lambda \otimes B_\mu \to B_\mu \otimes B_\lambda$, then $T_1 \otimes T_2, F(T_1 \otimes T_2) =: T_2' \otimes T_1'$ belong to the same isomorphic component. Moreover, they are in the same relative position in this component. It follows that we must have equality of their plactic products,

$$T_1 \cdot T_2 = T_2' \cdot T_1.$$

Thus, we can detect when we have an isomorphism by computing the plactic product of the tensor factors. It may seem difficult to compute the plactic product, given its description in Definition

2.5.15. However, there are many different well known combinatorial models for the plactic product, which make computation possible. The two main ones being Schensted insertion and jeu-de-taquin slides. The one we will describe and use is Schensted insertion.

Definition 2.5.18. Let T be a semistandard tableau, and $x \in \mathbb{N}$. First assume that T consists of a single row, $v_1v_2...v_\ell$. Then, we define $T \leftarrow x$ to be $v_1...v_\ell x$ if $x \geq v_\ell$, and otherwise we define $T \leftarrow x$ to be the two row tableau with first row $v_1...v_{i_0-1}xv_{i_0+1}...v_\ell$ and second row v_{i_0} where v_{i_0} is the leftmost letter that is larger than x. We think of the letter v_{i_0} as being "bumped". Now, suppose that T has k rows and that we have already defined $T \leftarrow x$ for all tableaux with at most $k-1$ rows. Consider the $S = \mathrm{Row}_1(T) \leftarrow x$. If S has one row, then $T \leftarrow x = T'$ where $\mathrm{Row}_1(T') = S$ and $\mathrm{Row}_i(T') = \mathrm{Row}_i(T)$ for $i > 1$. If S has two rows, let y be the unique element in its second row. Then $T \leftarrow x = T'$ where $\mathrm{Row}_1(T') = \mathrm{Row}_1(S)$ and $\mathrm{Row}_i(T') = \mathrm{Row}_i(\overline{T} \leftarrow y)$ for $i > 1$ defining $\overline{T}$ to be the tableau obtained from T by removing the first row. We define the Schensted insertion (or row insertion) $T \leftarrow w_1...w_k := (...((T \leftarrow w_1) \leftarrow w_2)... \leftarrow w_k)$.

Example 2.5.19. We give an example of Schensted insertion below.

$$\begin{array}{|c|c|c|c|}\hline 1&1&4&5\\ \hline 3&4\\ \cline{1-2} 4\\ \cline{1-1}\end{array} \leftarrow 237$$

$$= \begin{array}{|c|c|c|c|}\hline 1&1&2&5\\ \hline 3&4&4\\ \cline{1-3} 4\\ \cline{1-1}\end{array} \leftarrow 37$$

$$= \begin{array}{|c|c|c|c|}\hline 1&1&2&3\\ \hline 3&4&4&5\\ \hline 4\\ \cline{1-1}\end{array} \leftarrow 7$$

$$= \begin{array}{|c|c|c|c|c|}\hline 1&1&2&3&7\\ \hline 3&4&4&5\\ \cline{1-4} 4\\ \cline{1-1}\end{array}$$

The well known interpretation of the plactic product in terms of Schensted insertion is given by $T_1 \cdot T_2 = T_1 \leftarrow \mathrm{RR}(T_2)$.

Recall that by Remark 2.5.13 if a crystal can be written as a multiplicity free direct sum of B_λ, then it has a unique automorphism. The Pieri rule tells us that $s_\lambda s_{(r)}$ is multiplicity free (all Littlewood-Richardson coefficients are at most 1). Hence, $B_\lambda \otimes B_{(r)}$ has a unique automorphism, which implies that there is a unique isomorphism $B_\lambda \otimes B_{(r)} \cong B_{(r)} \otimes B_\lambda$.

We are now ready to describe the combinatorial R. Recall that for any two crystals C and D, the two tensor products $C \otimes D \cong D \otimes C$ are isomorphic. However, there is not necessarily a unique isomorphism between them. An affine crystal is a generalization of crystals, which have raising and lowering operators at index 0. It is a fact that all affine crystals possess unique automorphisms. In particular, if C and D are *affine*, then there *is* a unique isomorphism

$$C \otimes D \overset{\text{affine}}{\cong} D \otimes C.$$

This unique isomorphism is called the *combinatorial* $R-$*matrix* (which we will call the combinatorial R). One might ask why would we want to discuss affine crystals, as we have so far only considered classical crystals. The reason is that, the crystals we will mostly be concerned with are *Kirillov-Reshetikhin* (KR) crystals (denoted $B^{r,s}$), as well as tensor products of them. Any affine crystal can be viewed as classical crystal by "forgetting" the affine structure. It turns out, that if one forgets the affine strucrue of $B^{r,s}$, then one obtains the classical crystal $B_{(s^r)}$. The KR crystals we will be concerned with are $B^{1,s}$ (which have the classical structure of $B_{(s)}$).

Therefore, we can define the "combinatorial R" on $B_{(r)} \otimes B_{(s)}$. It is the unique isomorphism $B_{(r)} \otimes B_{(s)} \cong B_{(s)} \otimes B_{(r)}$. Indeed, we cannot define the combinatorial R on classical crystals. However, $B_{(r)}, B_{(s)}$ both have realizations as the affine KR crystals $B^{1,r}, B^{1,s}$. The combinatorial R descends to a classical crystal isomorphism, and because $B_{(r)} \otimes B_{(s)}$ only has one automorphism, any isomorphism $B_{(r)} \otimes B_{(s)} \cong B_{(s)} \otimes B_{(r)}$ must be the combinatorial R. We can characterize it using Schensted insertion (and our discussion of the plactic product) as follows.

Definition 2.5.20. Let T_1, T_2 be semistandard tableaux of shapes $(r), (s)$ respectively. We define

$$R(T_1 \otimes T_2) = T_1' \otimes T_2' \tag{2.10}$$

where T_1', T_2' are the unique semistandard tableaux of shapes $(s), (r)$ respectively such that $T_1 \leftarrow T_2 = T_1' \leftarrow T_2'$. Equation 2.10 is also written as

$$T_1 \otimes T_2 \simeq T_1' \otimes T_2'.$$

If

$$T_1 \otimes ... \otimes R(T_i \otimes T_{i+1}) \otimes ... \otimes T_k = T_1 \otimes ... \otimes T_i' \otimes T_{i+1}' \otimes ... \otimes T_k$$

then we also write

$$T_1 \otimes ... \otimes T_i \otimes T_{i+1} \otimes ... \otimes T_k \simeq T_1 \otimes ... \otimes T_i' \otimes T_{i+1}' \otimes ... \otimes T_k.$$

2.5.4 RSK, Recording Tableaux, and Paths

Note that B_λ gives a crystal theoretic way to encode tableaux. However, for our purposes, we would like a different way to describe tableaux crystal theoretically. This will be in terms of highest weight elements of tensor products of KR crystals (identifying KR crystals with their classical structure). These are also known in the literature as *paths*.

Remark 2.5.21. Schensted insertion is invertible if we know how the shape of the tableau $T' = T \leftarrow a$ changed. In particular, the location of the box added corresponds to the last step of the insertion process. Suppose this box has entry α and occurs in row r. We can then reverse the procedure by considering the largest entry in the immediate row above which is smaller than α and call it β. We then remove α from row r, and replace β with α. We then repeat this procedure with β instead of α. If α is in the first row, then we simply remove it, and conclude $a = \alpha$.

Example 2.5.22. Suppose we have the semistandard tableau

$$T \leftarrow a = \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 3 & 5 & 7 \\ \hline 2 & 2 & 5 & 5 & \\ \cline{1-4} 3 & 4 & & & \\ \cline{1-2} \end{array}$$

and we know that row insertion added a new box to the third row. The last step of row insertion always appends a box to a row. In particular, it is the largest entry of that row. Then, initially $\alpha = 4$ so that

$$\begin{array}{ccccc} 1 & 1 & 3 & 5 & 7 \\ 2 & 2 & 5 & 5 & \\ 3 & & & & \end{array}, \alpha = 4$$

$$\rightarrow \begin{array}{ccccc} 1 & 1 & 3 & 5 & 7 \\ 2 & 4 & 5 & 5 & \\ 3 & & & & \end{array}, \alpha = 2$$

$$\rightarrow \begin{array}{ccccc} 1 & 2 & 3 & 5 & 7 \\ 2 & 4 & 5 & 5 & \\ 3 & & & & \end{array}, \alpha = 1$$

We therefore conclude that $a = 1$. Indeed,

$$\begin{array}{ccccc} 1 & 2 & 3 & 5 & 7 \\ 2 & 4 & 5 & 5 & \\ 3 & & & & \end{array} \leftarrow 1 = \begin{array}{ccccc} 1 & 1 & 3 & 5 & 7 \\ 2 & 2 & 5 & 5 & \\ 3 & 4 & & & \end{array}.$$

Note that the plactic product is very non-injective. For example, by Remark 2.5.17, two different Yamanouchi words are the same under taking the plactic product of all their letters. So, by itself, the plactic product is unable to distinguish two Yamanouchi words. However, with some additional bookkeeping, it becomes possible to do so.

This bookkeeping is done using the Robinson-Schensted-Knuth (RSK) algorithm. On a crystal theoretic level, the RSK algorithm gives the plactic product along with an additional label which distinguishes which copy of B_λ the plactic product lives in. For example, consider a sequence of rows $(r_1), ..., (r_k)$ and consider $B_{(r_1)} \otimes ... \otimes B_{(r_k)} \equiv \oplus_\lambda B_\lambda^{\oplus c_\lambda}$. The map $\pi_{(r_1),...,(r_k)}$ gives a morphism from the tensor product to its decomposition in terms of B_λ. However, it does not care about whether there are multiple copies of B_λ (that is if $c_\lambda > 1$). The label the RSK algorithm uses is another tableau (of the same shape as the plactic product).

Definition 2.5.23. We define $\mathrm{RSK}_{(r_1),...,(r_k)}$ recursively on k. If $k = 1$, then

$$\mathrm{RSK}_{(r_1)}(T_1) = (T_1, R_1)$$

where $R_1 = \boxed{1}\boxed{...}\boxed{1} \in B_{(r_1)}$. Suppose $\mathrm{RSK}_{(r_1),...,(r_{k-1})}$ has already been defined and write

$$\mathrm{RSK}_{(r_1),...,(r_{k-1})}(T_1 \otimes ... \otimes T_{k-1}) = (P', Q')$$

where P', Q' have shape μ. Then $\mathrm{RSK}_{(r_1),...,(r_k)}(T_1 \otimes ... \otimes T_k) = (P, Q)$ where $P = T_1 \cdot ... \cdot T_k$ is of shape λ and Q is the tableau of shape λ with entries equal to Q' on the subshape μ, and whose remaining entries are k.

The RSK algorithm is in fact a bijection between rows $T_1, ..., T_k$ of shape $(r_1), ..., (r_k)$ and pairs of tableaux (P, Q) of the same shape such that the content of P is the union of contents of $T_1, ..., T_k$ and the content of Q is $(r_1, ..., r_k)$. Indeed, invertibility follows from Remark 2.5.21. We call the first entry of the image of RSK the "P" (or "insertion") tableau, and the second entry the "Q" (or "recording") tableau.

The insertion tableau is what records the crystal theoretic information, and the recording tableau is essentially just a label. However, it is often the case that we care about structure on the Q tableau, even if no structures are "supposed to exist". For example, a change in the recording tableau amounts to a different factorization of the P tableau. Note that in particular, this observation gives us a real procedure to compute the combinatorial R: we apply the content permuting crystal reflection operator to the recording tableau, and then compute inverse RSK. We will also later be interested in computing other types of refactorizations, which also correspond to crystal operators on the Q tableau.

As such, we wish to understand how changes in our crystal element induce and recording tableaux affect each other. Note that this is well understood in the case of the insertion tableau, as $\pi_{(r_1),...,(r_k)}$ is a crystal isomorphism onto its image when restricted to any connected component.

One way to assign meaning to the empty names of recording tableaux is to find the equivalence class of crystal elements with the same insertion tableau, and then compare them. This is accomplished via *Knuth relations* (for example see [2] or [20] for a reference). But what if we want to perform crystal operations on the recording tableau? The following concept will help us understand crystal operators on the recording tableau.

A useful model for tensor products of rows are *biwords*. A biword is simply a two row matrix with each biletter being regarded as a column. The bottom row of a biword represents the entries in the tensor product of rows. The top row represents which tensor factor each of the entries in the second row belong to. The top row is sorted in increasing order.

Example 2.5.24. The following is an example of a biword.

$$\begin{pmatrix} 1 & 2 & 2 & 2 & 3 & 3 \\ 1 & 2 & 1 & 1 & 2 & 2 \end{pmatrix}$$

The biletters are

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}.$$

The interaction of the combinatorics of biwords and the RSK algorithm is well studied. For example consider the combinatorial procedure of switching the top and bottom rows of the biword, and then sorting biletters by their first entry and then reverse sorting the biletters by their second entry among biletters with common first letter. It is known that this switches the insertion and recording tableaux of the biword. In particular, this implies that to give meaning to crystal operations on the recording tableau, we can conjugate the desired crystal operator with the operation of switching the rows of a biword.

Recall that the set of Yamanouchi words is in bijection with standard tableaux. We would like to understand how this bijection fits within the cyrstal theoretic framework. Indeed, in Section 2.4, Yamanouchi words were our primary data structure for representing standard young tableaux.

It is worth noting that although the Q tableau of Definition 2.5.23 can distinguish Yamanouchi words, the Q tableau of a (reverse) Yamanouchi word is not necessarily equal to the image under the bijection between Yamanouchi words and standard tableaux. It turns out that if we had defined RSK with *column insertion* instead of Schensted insertion, then the Q tableau would be equal.

The modern interpretation of the KKR bijection is not between tableaux and rigged configurations. It is between (classically) highest weight elements of tensor products of KR crystals and rigged configurations. To recover the original description of the KKR bijection, we pass through this column insertion variant of the RSK algorithm. The notion of column insertion is "dual" to row insertion, in the sense that the order of the tensor factors switch.

Definition 2.5.25. Let T be a semistandard tableau and x a letter. We write $x \to T$ to be the plactic product $x \cdot T$. The column insertion $w_n...w_1 \to T := (w_n \to (... \to (w_1 \to T)...))$. The insertion $x \to T$ can be given a direct construction: if x is greater than every element of the first column of T, then append x to the bottom of the first column of T. Otherwise, take the upper most element α which is greater or equal to x and replace it with x. Then column insert α into the remaining tableau.

Example 2.5.26. We compute the column insertion $342 \to \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 4 \\ \cline{1-1} 5 \\ \cline{1-1}\end{array}$.

$$342 \to \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 4 \\ \cline{1-1} 5 \\ \cline{1-1}\end{array}$$

$$= 34 \to \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 2 & 4 \\ \cline{1-2} 5 \\ \cline{1-1}\end{array}$$

$$= 3 \to \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 2 & 4 \\ \cline{1-2} 4 & 5 \\ \cline{1-2}\end{array}$$

$$= \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 2 & 4 & 4 \\ \hline 3 & 5 \\ \cline{1-2}\end{array}$$

Let $w_1...w_n$ be a Yamanouchi word. The Q tableau of the reverse word $w_n...w_1$ (using the column insertion variant of RSK) is the standard tableau corresponding to the Yamanouchi word $w_1...w_n$. In other words, the highest weight elements of $w \in B^n$ are in bijection with standard tableaux of size n, via the Q tableau of RSK using column insertion.

We can also check what the analogue of this correspondence is for semistandard tableaux by using the inverse of RSK. It turns out that highest weight elements in $B_{(\mu_1)} \otimes ... \otimes B_{(\mu_k)}$ are similarly in bijection with semistandard tableaux of content $(\mu_1, ..., \mu_k)$ under the Q tableau of RSK using column insertion. These highest weight elements of tensor products $B_{(\mu_1)} \otimes ... \otimes B_{(\mu_k)}$ are sometime called *paths*. Note that this includes the case where each $\mu_i = 1$ so that $B_{(\mu_1)} \otimes ... \otimes B_{(\mu_k)} = B^k$. To be more explicit about the correspondence, we give the following description.

Proposition 2.5.27. For a standard tableaux T, the corresponding crystal element is

$$\boxed{w_n} \otimes ... \otimes \boxed{w_1} \in B^n$$

where $w_1 ... w_n$ is the Yamanouchi word of T. If T is semistandard then the corresponding crystal element is $\boxed{w_1^{(1)} \mid ... \mid w_{\ell_1}^{(1)}} \otimes ... \otimes \boxed{w_1^{(k)} \mid ... \mid w_{\ell_k}^{(k)}} \in B_{(\ell_1)} \otimes ... \otimes B_{(\ell_k)}$ where

$$w_1^{(1)} \otimes ... \otimes w_{\ell_1}^{(1)} \otimes ... \otimes w_1^{(k)} \otimes ... \otimes w_{\ell_k}^{(k)}$$

is the crystal element corresponding to the standardization of T, and the letters grouped in the same tensor factor have the same content. Specifically, letters in the ith tensor factor have content $k - i + 1$.

Example 2.5.28. Consider the semistandard tableau

$$\begin{array}{|c|c|c|c|} \hline 1 & 1 & 3 & 4 \\ \hline 2 & 2 & 5 & 5 \\ \hline 4 \\ \cline{1-1} \end{array}.$$

The standardization is

$$\begin{array}{|c|c|c|c|} \hline 1 & 2 & 5 & 7 \\ \hline 3 & 4 & 8 & 9 \\ \hline 6 \\ \cline{1-1} \end{array}.$$

The crystal element corresponding to this standard tableau is

$$2 \otimes 2 \otimes 1 \otimes 3 \otimes 1 \otimes 2 \otimes 2 \otimes 1 \otimes 1.$$

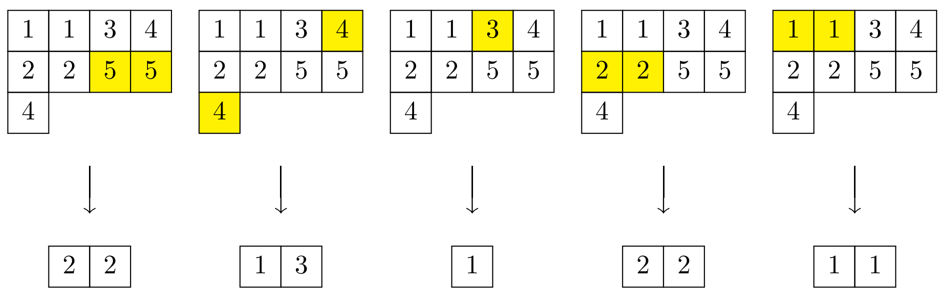
Grouping together letters whose corresponding cells (in the semistandard tableau) have the same content we get

$$\boxed{2 \mid 2} \otimes \boxed{1 \mid 3} \otimes \boxed{1} \otimes \boxed{2 \mid 2} \otimes \boxed{1 \mid 1}.$$

Indeed, the last two cells with respect to the standardization reading have content 5. So the first two letters get grouped together. The third and fourth last cells have content 4, so the third and fourth letters get grouped together. The fifth last cell has content 3, which no other cell has. So, it is not grouped together with any other letter. The sixth and seventh last cells have content 2. So they get grouped together. Finally, the eigth and ninth last read cells in the standardization reading both have content 1, so they are grouped together.

Alternatively, one can think of each row in the path as a reverse "Yamanouchi" word for each letter. That is, start with the largest letter in the semistandard tableau. Read the cells where this letter occurs, from right to left, top to bottom. Record the row numbers of these cells with respect to this reading order, and suppose we have $a_1, a_2, ..., a_x$. Then, the first tensor factor is $\boxed{a_1 \mid ... \mid a_x}$. We perform this procedure, reading the cells where the second largest letter occurs. If the row numbers are $b_1, ..., b_y$, then the second tensor factor is $\boxed{b_1 \mid ... \mid b_y}$. We continue this procedure until all cells have been read.

Example 2.5.29. Let us consider the same tableau as in the previous example. There are two fives which occur in rows 2 and 2 reading right to left, so the first tensor factor is $\begin{array}{|c|c|}\hline 2 & 2\\\hline\end{array}$. The two 4s occur in rows 1 and 3 reading right to left, top to bottom. So the second tensor factor is $\begin{array}{|c|c|}\hline 1 & 3\\\hline\end{array}$. There is only one 3, which occurs in row 1. Hence, the third tensor factor is $\begin{array}{|c|}\hline 1\\\hline\end{array}$. Both twos occur in rows 2 and 2 reading right to left, so the fourth tensor factor is $\begin{array}{|c|c|}\hline 2 & 2\\\hline\end{array}$. Finally, both ones occur in row one, so the last tensor factor is $\begin{array}{|c|c|}\hline 1 & 1\\\hline\end{array}$. This agrees with what we had before. The following diagram demonstrates the procedure.



As alluded to earlier, we remark that the action of the combinatorial R on paths is equivalent to the action of crystal reflection operators on semistandard tableaux through the identification given in Proposition 2.5.27.

2.6 Plactic Factorizations

Definition 2.6.1. Let $T \in \text{SSYT}(\lambda)$. If T_1, T_2 are tableaux such that $T_1 \cdot T_2 = T$, then we say $T_1 \cdot T_2$ is a plactic factorization of T.

Using the discussion above, we can calculate precisely how many factorizations a tableau may have. The following theorem can be thought of as an alternative description of the Littlewood-Richardson rule. It can also be thought of as a "element wise description" of the decomposition of $B_\mu \otimes B_\nu$ discussed in the previous section.

Theorem 2.6.2. Let $T \in \text{SSYT}(\lambda)$. The number of pairs $T_1 \in \text{SSYT}(\mu), T_2 \in \text{SSYT}(\nu)$ such that $T_1 \cdot T_2 = T$ is $[s_\lambda](s_\mu s_\nu) = c^\lambda_{\mu\nu}$. In particular, the number of such factorizations only depends on the shapes λ, μ, ν.

Proof. Each such factorization $T_1 \cdot T_2$ implies that $T_1 \otimes T_2$ lies in a connected component isomorphic to B_λ. If $T'_1 \cdot T'_2$ is also a factorization of T and is in the same connected component as $T_1 \otimes T_2$ then because $\pi_{\mu,\nu}$ is (an isomorphism hence) injective on each connected component, we must have $T'_1 \otimes T'_2 = T_1 \otimes T_2$ hence $T_1 = T'_1$ and $T_2 = T'_2$. This shows that the number of factorizations is at most $c^\lambda_{\mu\nu}$. Moreover, for any connected component C of $B_\mu \otimes B_\nu$ with $C \cong B_\lambda$, there is a unique crystal isomorphism $\varphi : B_\lambda \to C$. Note that the inverse of an isomorphism is an isomorphism. Because there is a unique crystal isomoprhism $C \to B_\lambda$ it follows that $\pi_{\mu,\nu} \circ \varphi = \text{Id}_{B_\lambda}$. In particular, if $\varphi(T) = T_1 \otimes T_2 \in C \subseteq B_\mu \otimes B_\nu$ then $T_1 \cdot T_2 = \pi_{\mu,\nu}(T_1 \otimes T_2) = \pi_{\mu,\nu}(\varphi(T)) = T$. Hence, there is also at least one factorization per copy of B_λ. The result follows. $\square$

Corollary 2.6.3. Let $T \in \mathrm{SSYT}(\lambda)$. Suppose either μ or ν is a single row. Then there is at most one factorization $T_1 \in \mathrm{SSYT}(\mu), T_2 \in \mathrm{SSYT}(\nu)$ with $T = T_1 \cdot T_2$

Proof. If μ is a single row, the result immediately follows from 2.6.2 and the Pieri rule. If ν is a single row, the result follows from 2.6.2 and commutativity of the ring Λ_n. □

Corollary 2.6.4. Let $T \in \mathrm{SSYT}(\lambda)$. Suppose either μ or ν is a single column. Then there is at most one factorization $T_1 \in \mathrm{SSYT}(\mu), T_2 \in \mathrm{SSYT}(\nu)$ with $T = T_1 \cdot T_2$

Proof. The same proof as above, but using the dual Pieri rule instead of the Pieri rule. □

Remark 2.6.5. The above two corollaries can also be proved using RSK, because the shape of $T \leftarrow w_1...w_n$ is well understood when $w_1 \leq ... \leq w_n$ using the Pieri rule. However, the more general Theorem 2.6.2 is easier using the crystal perspective. Theorem 2.6.2 is not a characterization of the Littlewood-Richardson rule which is suited for computational purposes. However the interpretation using plactic factorizations is a perspective that will prove useful later, when we control for "local products" of tensor factors staying the same.

Theorem 2.6.2 implies that we can "refactor" a product of two tableaux in at most one way if we fix the shapes of the new factors, and one of the factors is a row or column. In this light, we can view the combinatorial R as a refactorization given that one of the shapes is a row or column.

Definition 2.6.6. Let $w := W_1 \otimes ... \otimes W_k \in \otimes_{i=1}^k B_{(R_i)}$ where each (R_i) is a single row of length $R_i \in \mathbb{N}$ and each $W_i \in B_{(R_i)}$. Suppose that $W_i \cdot W_{i+1}$ can be refactored as $W_i' \cdot W_{i+1}'$ where $W_i' \in B_{(R_i-1)}, W_{i+1}' \in B_{(R_{i+1}+1)}$. Then, W_i', W_{i+1}' are uniquely determined and we define

$$\mathrm{ref}_i(w) = W_1 \otimes ... \otimes W_{i-1} \otimes W_i' \otimes W_{i+1}' \otimes W_{i+2} \otimes ... \otimes W_k$$

If no such refactorization exists, then we define $\mathrm{ref}_i(w) = \emptyset$.

Remark 2.6.7. *If $W_1 \otimes ... \otimes W_k$ corresponds to a semistandard tableau T, then we can interpret ref_i as the raising operator e_{k-i}.*

Example 2.6.8. Consider

$$T = \begin{array}{|c|c|c|c|}\hline 1 & 1 & 2 & 2 \\ \hline 2 & 2 & 3 & 3 \\ \hline 3 \\ \cline{1-1}\end{array}$$

which corresponds to

$$w = \begin{array}{|c|c|c|}\hline 2 & 2 & 3 \\ \hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1 & 1 & 2 & 2 \\ \hline\end{array} \otimes \begin{array}{|c|c|}\hline 1 & 1 \\ \hline\end{array}.$$

Notice,

$$\mathrm{ref}_2(w) = \begin{array}{|c|c|c|}\hline 2 & 2 & 3 \\ \hline\end{array} \otimes \begin{array}{|c|c|c|}\hline 1 & 2 & 2 \\ \hline\end{array} \otimes \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline\end{array}$$

which corresponds to the tableau

$$\begin{array}{|c|c|c|c|}\hline 1 & 1 & 1 & 2 \\ \hline 2 & 2 & 3 & 3 \\ \hline 3 \\ \cline{1-1}\end{array} = e_1(T).$$

We define an operation on B^n which groups together the maximal ascent sequences.

Definition 2.6.9. We define a map $\tau : B^n \to \oplus_{\lambda\vdash n} \otimes_{i=1}^{\ell(\lambda)} B_{(\lambda_i)}$ as follows. Let $w_1 \otimes ... \otimes w_n \in B^n$. Then,

$$\tau(w_1 \otimes ... \otimes w_n) = \boxed{w_1}\boxed{...}\boxed{w_{a_1}} \otimes \boxed{w_{a_1+1}}\boxed{...}\boxed{w_{a_2}} \otimes ... \otimes \boxed{w_{a_k+1}}\boxed{...}\boxed{w_n}$$

Where $w_i \leq w_j$ if $i, j \in [w_{a_t-1} + 1, w_{a_t}]$ for some t defining $a_0 = 0$. We moreover insist that each $w_{a_t} > w_{a_t} + 1$.

Example 2.6.10. Consider $w = 1 \otimes 2 \otimes 3 \otimes 2 \otimes 2 \otimes 4 \otimes 2 \otimes 5$. Then

$$\tau(w) = \boxed{1}\boxed{2}\boxed{3} \otimes \boxed{2}\boxed{2}\boxed{4} \otimes \boxed{2}\boxed{5}$$

A particular refactorization that will be important to us is the following

Definition 2.6.11. Define $\text{raise}_i = \text{RR} \circ \text{ref}_i \circ \tau : B^n \to B^n$. If $\text{ref}_i(\tau(w)) = \emptyset$, then take $\text{raise}_i(w) = \emptyset$.

We would like to realize raise_i as a combinatorial operation on words, which will be accomplished in Theorem 2.6.14. First, we must define the combinatorial operation which we claim is equal to raise_i.

Definition 2.6.12. Let $w = w_1...w_n$ be a word with $w_1 \leq ... \leq w_k$, $w_{k+1} \leq ... \leq w_n$, and $w_k > w_{k+1}$. Define descent shift right (DSR):

$$\text{DSR}_i := w_1...w_{i-1}w_{i+1}...w_k w_{k+1}...w_{k+j}w_i w_{k+j+1}...w_\ell...w_n$$

where $j \in [1, \ell - k]$ is the largest number such that $w_j \leq w_i$.

It may seem strange to call the operation descent shift right because it is shifting ascent sequences rather than descent sequences. However, the words we will often encounter are reversed Yamanouchi words so the ascent and descent sequences are swapped.

Example 2.6.13. Consider $w = 12322425$. The ascent sequence containing index 3 is 123. The next ascent sequence is 224. The only place we can insert a 3 into 224 to keep it ascending is after the second 2. So it becomes 2234 where as the first ascent sequence loses the 3 and becomes 12. Hence, we obtain $\text{DSR}_3(w) = 12223425$

We can clearly extend this definition to elements of the crystal B^n by using the identification $w_1...w_n = \boxed{w_1} \otimes ... \otimes \boxed{w_n}$ discussed in Section 2.5.

From the Pieri rule, we know that the plactic product of two rows yields a tableau with at most two rows. We can calculate certain factorizations explicitly using RSK. In particular, we can calcualte raise_i directly using RSK. However, DSR gives us a combinatorial model for computing raise_i directly on the word as the following theorem demonstrates.

Theorem 2.6.14. Let $\tau(w) = W_1 \otimes ... \otimes W_k \in \otimes_{i=1}^k B_{(\lambda_i)}$ for some $\lambda \vdash n$. Suppose $\text{raise}_i(w) \neq \emptyset$. Then $\text{raise}_i(w) = \text{DSR}_j(w)$ where w_j is the largest entry in W_i such that $w_j \in \text{Row}_1(W_i \leftarrow W_{i+1})$.

Proof. Write $\tau(w) = W_1 \otimes ... \otimes W_k$ where each $W_i \in B_{(\lambda_i)}$. Note that the computation of raise_i is local. That is, it only depends on W_i, W_{i+1} so

$$\text{raise}_i(W_1 \otimes ... \otimes W_k) = W_1 \otimes ... \otimes W_{i-1} \otimes \text{raise}_1(W_i \otimes W_{i+1}) \otimes W_{i+2} \otimes ... \otimes W_k$$

Write $W_i = \boxed{u_1}\boxed{...}\boxed{u_{\lambda_i}}$ and $W_{i+1} = \boxed{v_1}\boxed{...}\boxed{v_{\lambda_{i+1}}}$. Consider the biword

$$w = \binom{1}{u_1}, ..., \binom{1}{u_{\lambda_i}}, \binom{2}{v_1}, ..., \binom{2}{v_{\lambda_{i+1}}}.$$

The insertion tableau of w is the plactic product $W_i \cdot W_{i+1}$. The recording tableau Q is a two row tableau with content $(\lambda_i, \lambda_{i+1})$. Applying the falling operator $f_1(Q)$, we note that the biword $w' = \binom{1}{u'_1}, ..., \binom{1}{u'_{\lambda_i - 1}}, \binom{2}{v'_1}, ..., \binom{2}{v'_{\lambda_{i+1}+1}}$ corresponding to $(W_i \cdot W_{i+1}, f_1(Q))$ under RSK yields a refactorization $W'_i \cdot W'_{i+1} = W_i \cdot W_{i+1}$ where W_i is the row with entries $u'_1, ..., u'_{\lambda_i - 1}$ and W_{i+1} is the row with entries $v'_1, ..., v'_{\lambda_{i+1}-1}$. By Corollary 2.6.3, this is the unique factorization of $W_i \cdot W_{i+1}$ into two rows of lengths $\lambda_i - 1, \lambda_{i+1}$. Hence, $\text{raise}_1(W_i \otimes W_{i+1}) = W'_i \otimes W'_{i+1}$. Moreover we have that under RSK, $(W_i \cdot W_{i+1}, f_1(Q))$ corresponds to some $\text{DSR}_{j'}$ on $u_1 ... u_{\lambda_i} v_1 ... v_{\lambda_{i+1}}$. In particular, j' is the column of the right-most 1 in Q. Hence it corresponds to some DSR_j on w. □

The characterization of j given in the proof of Theorem 2.6.14 implies that we can determine the letter that gets moved over in the following way. Insert $W_i \leftarrow W_{i+1}$. Consider the right-most element in the first row that was not bumped. This is the letter that gets moved over.

Example 2.6.15. Consider $\text{raise}_1(11233122)$. We have

$$\tau(211233122) = \boxed{1}\boxed{1}\boxed{2}\boxed{3}\boxed{3} \otimes \boxed{1}\boxed{2}\boxed{2}.$$

Also,

$$11233 \leftarrow 122 = \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & 2 & 2 \\ \hline 2 & 3 & 3 \\ \cline{1-3}\end{array}.$$

However, note that both 2s in the first row actually come from 122 and not 11223. Therefore, the right most unbumped element is actually a 1. Hence, $\text{raise}_1(11233122) = \text{DSR}_1(11233122) = \text{DSR}_2(11233122) = 12331122$.

2.7 Inverse Time Evolution, T_∞

In this section we will prove our core results: Theorem 2.7.43, the closely related Theorem 2.7.42, and Theorem 2.7.35.

Theorem 2.7.43 and Theorem 2.7.42 are both results which understand how certain perturbations of the riggings of a rigged configuration induce changes on the corresponding tableau. Theorem 2.7.43 is the more powerful of the two, and is more widely applicable. However, Theorem 2.7.42 allows one to occasionally circumvent the careful considerations that need to be made in Section 2.8 which are used to extend Theorem 2.7.43 to control all riggings in the first rigged partition. As we will see later, Theorem 2.7.42 will be used as the "half-way" point to Theorem 2.7.43.

Theorem 2.7.35 gives a characterization of the smallest row of the first partition of a rigged configuration, in terms of a statistic we introduce called "descent coenergy" (DCE). This statistic is capable of measuring the extent to which adjacent descent sequences satisfy the pairing condition. In light of this remark, Theorem 2.7.35 provides a sort of partial converse to Theorem 2.4.21. Descent coenergy also measures the partial shapes of plactic products of tensor factors in $\tau(w)$, which turns out to be very important for some of the arguments we make.

The main tool we use in this section is the multicolour box-ball system, as well as its interaction with rigged configurations. In subsection 2.7.1, we review the multicolour box-ball system, its time evolution rule, as well as the crystal formulation of the inverse time evolution rule. In subsection 2.7.2 we give an "elementary proof" of the fact the asymptotic state of the multicolour box-ball system exists and is described by rigged configurations. The proof is "elementary" from the perspective of combinatorics as it only relies on the crystal formulation of time evolution, rigged configurations, and the KKR bijection rather than the origins of the multicolour box-ball system in integrable systems.

In subsection 2.7.3, we introduce the DCE statistic, and investigate how it behaves under inverse time evolution. We also investigate how the DCE statistic and soliton decompositions are related to each other. In particular, we give a characterization of certain stable states in the multicolour box-ball system using the DCE statistic. This section contains many of the important lemmas which are used in subsection 2.7.4. In subsection 2.7.4 we prove Theorems 2.7.43, 2.7.42, and 2.7.35.

Before we begin, let us introduce some notational conventions we will use in this section. We write $\boxed{1}^N = 1^N$ to be the n-fold tensor product $\boxed{1} \otimes ... \otimes \boxed{1}$. For the sake of brevity, we will often omit the boxes. A word $v_1...v_k$ is assumed to be interpreted as $\boxed{v_1} \otimes ... \otimes \boxed{v_k}$. In contrast, rows are written explicitly to avoid confusion, or we use τ if a word is increasing. That is, the row containing elements $v_1...v_k$ will be written as $\boxed{v_1 \mid ... \mid v_k}$ or $\tau(v_1...v_k)$ except when it is clear from context that a word is meant to be interpreted as a row (for example, an equation with a single row on the left hand side and a single word on the right hand side). One other exception we will mention is with respect to Schensted insertion. We write

$$w_1...w_n \leftarrow v_1...v_m := \boxed{w_1 \mid ... \mid w_n} \leftarrow v_1...v_m.$$

2.7.1 The Multicolour Box-Ball System

The box-ball system was introduced by Takahashi and Satsuma in [22]. It is an "ultradiscretization" of the Korteweg de-Vries (KdV) equation, which is reflected in its dynamics. What we will be concerned with is the *multicolour box-ball system*. The multicolour box-ball system is a natural generalization of the traditional box-ball system, and was introduced in [21].

Imagine an infinite one dimensional lattice, and place a finite number of positive integers (greater than one) on this lattice. We place a one for every unused cell, and represent them as a "." diagrammatically. We call each of these configurations a *state*.

Example 2.7.1. An example of a state is depicted below. The dots on the left and right are implied to extend out to infinity.

```
..........368...24.....9843.45...........
```

We think of the positive integers (greater than one) as coloured balls, and each of the ones (or dots) are thought of as boxes (or vacuum). We can define a rule to *evolve* the system, known as *time evolution*. This can be described with *Takahashi's algorithm* as follows.

Algorithm 2.7.2 ([21])**.** Let $w \in B^n$. Initially set a to be the largest letter of w. Take the right-most copy of a, and exchange it with the closest 1 to its left. Then repeat this for all other copies of a starting from the right and moving to the left. After all copies of a have been moved, set $a = a - 1$. Repeat until $a = 1$. The resulting word is $T_\infty(w)$.

Example 2.7.3. We apply Algorithm 2.7.2, listing out each step where a letter is moved.

```
..........368...24.....9843.45...........
..........368...24....9.843.45...........
..........368...24....98.43.45...........
.........836....24....98.43.45...........
........683.....24....98.43.45...........
........683.....24....98.4354...........
........683.....24....984435............
........683.....24...4984.35.............
........683....42....4984.35.............
........683...24.....4984.35.............
```

We note that the definition of time evolution does not strictly require that the coloured balls be placed in an infinite lattice. Indeed, it suffices for there to be a large number of ones to the left of the left-most coloured ball, and a large number of ones to the right of the right-most coloured ball.

This observation is important in explaining the crystal reformulation of the multicolour box-ball system. The multicolour box-ball system and its corresponding time evolution rule was reformulated in [4] using elements in B^n (beginning and ending with a large number of ones) as states, and the combinatorial R as time evolution.

The crystal description of the time evolution rule turns out to be much more useful theoretically. In particular, the crystal description of *inverse time evolution* will be important to us.

Definition 2.7.4. Consider a word $w \in B^n$. Let N be sufficiently large ($N \geq n$ suffices) and consider the element $v := \boxed{1}\boxed{1}\boxed{...}\boxed{1} \in B_{(N)}$. Let $v \in B_{(N)}$. Under combinatorial R, we have $v \otimes w \simeq w' \otimes v'$. We define $T^{-1}_{\infty,v}(w) := w'$. If $v = u$ we define $T^{-1}_\infty := T^{-1}_{\infty,v}$. We also define $\mathrm{RF}_{\infty,v}(w) = v'$. We call T^{-1}_∞ the *inverse time evolution.*

Remark 2.7.5. Suppose $w \in B^n$ is highest weight. Then, via combinatorial R,

$$\boxed{1}\boxed{...}\boxed{1} \otimes w \simeq w' \otimes U.$$

Because the combinatorial R is a crystal isomorphism, $w' \otimes U$ is a highest weight element. We see from the definition of the tensor product that U must be highest weight. Indeed, using the signature rule on two elements we see that if $\varepsilon_i(U) > 0$ for any i then there is at least one unmatched left brace in

$$)^{\varphi_i(w')}(^{\varepsilon_i(w')})^{\varphi_i(U)}(^{\varepsilon_i(U)}.$$

Hence, $U = \boxed{1}\boxed{...}\boxed{1}$ is again a row consisting entirely of ones.

In light of the above remark, we can define the (ordinary non-inverse) time evolution by having the large row of ones start at the right in Definition 2.7.4. This is denoted T_∞. However, this is only a true inverse if $w \in B^n$ begins and ends with a large number of ones, as otherwise we don't have a version of Remark 2.7.5 for regular time evolution. Indeed, note that $\begin{array}{|c|}\hline 2\\\hline\end{array} \otimes \begin{array}{|c|}\hline 1\\\hline\end{array}$ is highest weight (it is a reverse Yamanouchi word). However, we see

$$
\begin{aligned}
&\begin{array}{|c|}\hline 2\\\hline\end{array} \otimes \begin{array}{|c|}\hline 1\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1 & ... & 1 & 1\\\hline\end{array}\\
\simeq &\begin{array}{|c|}\hline 2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1 & ... & 1 & 1\\\hline\end{array} \otimes \begin{array}{|c|}\hline 1\\\hline\end{array}\\
\simeq &\begin{array}{|c|c|c|c|}\hline 1 & ... & 1 & 2\\\hline\end{array} \otimes \begin{array}{|c|}\hline 1\\\hline\end{array} \otimes \begin{array}{|c|}\hline 1\\\hline\end{array}
\end{aligned}
$$

and that $\begin{array}{|c|c|c|c|}\hline 1 & ... & 1 & 2\\\hline\end{array}$ is not highest weight. There is a symmetrical problem for inverse time evolution. That is, if the highest weight state $w \in B^n$ does not begin and end with a large number of ones, then $T_\infty^{-1}(w)$ is not necessarily highest weight.

We note that the crystal theoretic definition of time evolution and inverse time evolution can be applied to elements $w \in B^n$ which do not necessarily begin and end with a large number of ones. As stated earlier, in this case, the two operations are not inverses of each other (and do not necessarily map highest weight elements to highest weight elements). Nevertheless, this is sometimes a useful perspective as it lets us prove more general results about the combinatorial R and apply them to time evolution/inverse time evolution.

The reason we are interested in the inverse of time evolution rather than time evolution itself, is because the inverse interacts very nicely with ascent sequences as the following two lemmas demonstrate.

Lemma 2.7.6. Let $v_1 \leq ... \leq v_k$. Let $x \in \mathbb{N}$ and consider $u = \begin{array}{|c|c|c|c|c|c|c|}\hline 1 & 1 & ... & 1 & v_1 & ... & v_k\\\hline\end{array}$. Suppose $u \otimes x \simeq x' \otimes u'$ so that $x' = T_{\infty,u}^{-1}(x)$ and $u' = \mathrm{RF}_{\infty,u}(x)$. Compute the insertion $T = u \leftarrow x$. If T has one row, then $x' = 1$ and u' is T with a box with filling one removed. If T has two rows, then the second row of T has exactly one entry which is equal to x'. The first row is again equal to u'.

Proof. First suppose T has one row. Let T' be T with a box with filling one removed. Then clearly $T = u \leftarrow x = 1 \leftarrow T'$ and the result follows. If T has two rows, with second row having single entry y, then we notice $T = u \leftarrow x = y \leftarrow \mathrm{Row}_1(T)$ and the result follows. $\square$

Lemma 2.7.7. Let $v_1 \leq ... \leq v_k$ (with $v = v_1...v_k$) and $u_1 \leq ... \leq u_\ell$ (with $u = u_1...u_\ell$). Consider the plactic product of the two one row rectangles $T := \begin{array}{|c|c|c|}\hline v_1 & ... & v_k\\\hline\end{array} \cdot \begin{array}{|c|c|c|}\hline u_1 & ... & u_\ell\\\hline\end{array}$. Then, $T_{\infty,v}^{-1}(u) = \mathrm{RR}(\mathrm{Row}_2(T) \otimes 1^{\ell - |\mathrm{Row}_2(T)|})$. $\mathrm{RF}_{\infty,v}(u)$ is $\mathrm{RR}(\mathrm{Row}_1(T))$ with some ones appended to the left.

Proof. Follows immediately from Lemma 2.7.6. $\square$

We can interpret the above lemmas as follows. We (Schensted) insert the letters of our state into an infinite row. If there is a bump, then the bumped element replaces the letter we inserted. If there is no bump, then we replace the inserted letter with a 1. If we have an entire ascent sequence, we treat that ascent sequence as a row, T. We then (Schensted) insert T into the infinite row. The second row of the infinite row after insertion (along with some additional 1s) replace T.

2.7.2 The Multicolour Box-Ball System, Rigged Configurations, and the Soliton Decomposition

In what follows, we will assume all elements $w \in B^n$ begin and end with a large number of ones. Because we will be using the crystal theoretic formulation of time evolution, the ordering of our "sufficiently large values" is important. Let H be the length of the row we use in the Definition 2.7.4. While picking $H \geq n$ certainly suffices, it is far from necessary. If we consider the space of all words we care about, H only needs to be at least the maximum number of letters greater than one.

Let M, N be the padding of ones we put on every word (to the left and right respectively). We want $M, N \gg H \geq \#\{w_i : w_i > 1\}$ for all words $w_1...w_n \in B^n$ we consider. We want to be able to fit the entirety of (the entries greater than one of) w into the row used in time evolution, and we want to be able to apply time evolution as many times as we like (where it comes out as a row of ones on the other side, which is why we need the padding).

From the non-combinatorial origins of the box-ball system, it should be clear that under enough time evolutions, any state eventually "stabilizes". In particular, we can think of the blocks of numbers in the multicolour box-ball system as "colliding" with each other and we expect that given enough time, there will be no more interaction between the blocks.

Example 2.7.8. Consider the state,

```
.......344.233.44..........
```

There are two blocks: $233, 44$. Under time evolution, these blocks "collide":

```
..........233.44..........
......2334...4............
..2334......4.............
```

The collision results in two blocks: $2334, 4$. It is clear that under any further time evolutions, the state remain a union of the two blocks $2334, 4$, which will move further and further apart.

This asymptotic state is known as a *soliton decomposition.* It is not obvious that any state eventually becomes a soliton decomposition from a combinatorial/crystal perspective. The goal of this subsection is to precisely define a soliton decomposition as well as prove its existence for highest weight $w \in B^n$.

Because we are technically working with a finite number of boxes, it becomes difficult to compare states which are "essentially the same".

Example 2.7.9. The states 111233111 and 112331111 are different. They differ by the finite number of ones to the left and right of 233.

As such, for $w \in B^n$, we use $\mathrm{ess}(w)$ to denote the elements w' with all the ones on the left and right removed. We call this the *essential part* of w.

Example 2.7.10. Consider $w = 1111123445113441111$. Then, $\mathrm{ess}(w) = 2344511344$.

To describe the non-interacting parts of a soliton decomposition we make the following definition.

Definition 2.7.11. Suppose $\mathrm{ess}(w) = \mathrm{RR}(S_1) \otimes 1^{a_1} \otimes \mathrm{RR}(S_2) \otimes ... \otimes 1^{S_{\ell-1}} \otimes \mathrm{RR}(S_\ell) \in B^n$ with each $a_i \geq 0$, each $S_1 \in B_{(k)}$ with every entry greater than one. We say w is a homogeneous state of degree k. If $\mathrm{ess}(T_\infty(w)) = \mathrm{ess}(w)$, we say that w is a *homogeneous stable state of degree* k. We say the $\mathrm{RR}(S_i)$ are the solitons of w so that ℓ is the number of solitons of w.

Example 2.7.12. Consider $w = 11111334122311111$. Computing one step of time evolution (with each line showing the movement of each letter).

```
.....334.223.....
....433..223.....
....433.322......
...343..322......
..334...322......
..334..232.......
..334.223........
```

Hence, $T_\infty(w) = 11334122311111111$ which has the same essential part as w. So w is a homogeneous stable state of degree 3.

With this definition, we can describe a soliton decomposition as a bunch of homogeneous stable states which only move further apart from each other as time evolution continues. More precisely,

Definition 2.7.13. Let $w \in B^n$. Suppose there are homogeneous stable states $R_1, ..., R_k$ of degrees $d_1 > ... > d_k$ such that

$$\text{ess}(w) = \text{ess}(R_1) \otimes 1^{a_1} \otimes \text{ess}(R_2) \otimes ... \otimes 1^{a_{k-1}} \text{ess}(R_k).$$

Moreover assume that for any $m \geq 1$, there exists $b_1, ..., b_{k-1}$ satisfying $b_i \geq a_i$ such that

$$\text{ess}(T_\infty(w)^{(m)}) = \text{ess}(R_1) \otimes 1^{b_1} \otimes \text{ess}(R_2) \otimes ... \otimes 1^{b_{k-1}} \text{ess}(R_k).$$

Then we say that w is a *soliton decomposition.*

Definition 2.7.14. Let $w \in B^n$. Suppose that there exists some m such that $T_\infty^{(m)}(w)$ is a soliton decomposition. We say that $T_\infty^{(m)}(w)$ is the soliton decomposition of w.

For a highest weight $w \in B^n$, we can prove that a soliton decomposition of w always exists. Moreover, it is an amazing fact that we can describe the soliton decomposition by using the corresponding rigged configuration. To prove this, we must first recall some facts about how rigged configurations behave under time evolution.

Theorem 2.7.15 ([9] Proposition 2.6)**.** Let $w \in B^n$ be a highest weight element and $X = (\nu, J)$ be its corresponding rigged configuration. Let $X' = \Phi(T_\infty(w))$. Then, $X^{(i)} = {X'}^{(i)}$ for $i > 1$, and ${X'}^{(1)} = \{(x, \alpha + x) : (x, \alpha) \in X^{(1)}\}$.

The proof of the above theorem is actually very simple, and largely follows from the fact that the KKR map is invariant under the combinatorial R along with the crystal formulation of time evolution.

Theorem 2.7.15 explains how the riggings change under time evolution. We will also need to track the coriggings.

Corollary 2.7.16. Let $w \in B^n$ with $\Phi(w) = X$ and $\Phi(T_\infty(w)) = X'$. Then, the coriggings decrease faster on larger rows. In particular, $x < y$ and for any $k > 0$ there exists $M > 0$ so that for all $m \geq M$,

$$\min(\alpha : (x, \alpha) \in \tilde{X}^{(1)}) - \max(\beta : (y, \beta) \in \tilde{X'}^{(1)}) \geq k$$

Moreover, the relative coriggings among rows is invariant under time evolution. More precisely, if $\alpha_0 = \min(\alpha : (x, \alpha) \in \tilde{X}^{(1)})$ and $\beta_0 = \min(\beta : (x, \beta) \in \tilde{X'}^{(1)})$ then we have the equality of multisets,

$$\{\alpha - \alpha_0 : (x, \alpha) \in \tilde{X}^{(1)}\} = \{\beta - \beta_0 : (x, \beta) \in \tilde{X'}^{(1)}\}$$

Proof. Follows from Theorem 2.7.15 along with the fact that X, X' have the same configuration (hence the same vacancy numbers). □

We are ready to prove the existence of a soltion decomposition for all highest weight $w \in B^n$. The following Theorem is a folklore result in rigged configurations, soliton theory, and crystal theory. We give an explicit statement and proof of this result. Our proof only relies on the definition of the KKR map, and Theorem 2.7.15 (which recall is [9, Proposition 2.6]). The proof of [9, Proposition 2.6] only relies on the definition of the KKR map and [8, Lemma 8.5].

Theorem 2.7.17 (Folklore — Soliton Decomposition of Reverse Yamanouchi Words)**.** Let $w \in B^n$ be highest weight and let $(\nu, J) = \Phi(w)$ be the corresponding rigged configuration. Then there exists M such that for all $m \geq M$, $T_\infty^{(m)}(w)$ is a soliton decomposition. Moreover, if $R_1, ..., R_k$ are the corresponding homogeneous stable states of degrees $d_1 > ... > d_k$ then the d_i are the distinct parts of $\nu^{(1)}$. The number of solitons in each R_i is the multiplicity of d_i in $\nu^{(1)}$.

Proof. Let $X = (\nu, J) = \Phi(w)$. Take $k = \ell(\nu^{(1)})$ and pick any $b_1, ..., b_{k-1} \geq 0$. Let $\lambda_1 > ... > \lambda_k$ be the distinct parts of $\nu^{(1)}$. Let $\mathcal{M}_i$ be the multiplicity of λ_i in $\nu^{(1)}$. Let $\mu_{\geq i}$ be the partition $\nu^{(1)}$ with all parts of sizes $\lambda_1, ..., \lambda_{i-1}$ removed. Pick M large enough so that if $X' = \Phi(T_\infty^M(w))$ then for each j,

$$\begin{aligned}&\min_{1\leq i\leq \mathcal{M}_j}(\alpha : (x,\alpha) \in \tilde{X}'^{(1)}, x = \lambda_j) - \min_{1\leq i\leq \mathcal{M}_{j-1}}(\alpha : (x,\alpha) \in \tilde{X}'^{(1)}, x = \lambda_{j-1})\\ &\geq b_{k-1} + 2\lambda_j \mathcal{M}_{j-1}. + \sum_{(\lambda_{j-1},\alpha)\in \tilde{X}^{(1)}} (\alpha - \alpha_0)\end{aligned}$$

where $\alpha_0 = \min(\beta : (\lambda_{j-1}, \beta) \in \tilde{X}^{(1)}$. We can do this by Corollary 2.7.16 because the RHS is constant (also note that α_0 is invariant under time evolution by Corollary 2.7.16). Consider the inverse KKR bijection on X'. The above construction implies the first row of the chain table is of the form

$$012...\lambda_k\epsilon...\epsilon 012...\lambda_k\epsilon...\epsilon 012...\lambda_1$$

where the tuple comprised of the lengths of the connected sequences is

$$\{\lambda_k\}^{\mathcal{M}_k} \times \{\lambda_{k-1}\}^{\mathcal{M}_{k-1}} \times ... \times \{\lambda_1\}^{\mathcal{M}_1}$$

and these connected sequences comprise exactly the non-ϵ valued entries. The length of each string of ϵs between $012...\lambda_j\epsilon...\epsilon 012...\lambda_{j-1}$ is of length at least b_{j-1}. Each ϵ corresponds to a 1 in the Yamanouchi word (because this is the first row, any letter at least 2 would result in a value which isn't ϵ). Moreover, all these connected sequences must be descent sequences by Corollary 2.4.10 (comprised of letters greater than 1). This shows that $T_\infty^{(M)}(w)$ is broken up into homogeneous states of degrees $\lambda_1, ..., \lambda_k$ with number of solitons $\mathcal{M}_1, ..., \mathcal{M}_k$ (recall that $T_\infty^{(M)}(w)$ and its corresponding Yamanouchi word are reverses of each other).

It remains to show these homogeneous states are stable, and that they are the same for any $m > M$. The fact they are the same for any $m > M$ follows from the fact that the relative coriggings among rows with the same length remain unchanged under time evolution. So the part of the chain table concerning the construction of rows of length λ_j is the same. In particular, the length of the chains are the same (which determines the letters in the corresponding highest weight crystal element).

It then follows that they are stable. If we pick $b_i > \sum_{j=1}^{k} |\mathrm{ess}(R_j)| =: S$ we note that each R_i is surrounded by more than S ones. It follows that the computation of time evolution is local on the subword $1^{S+1} \otimes R_i \otimes 1^{S+1}$. Indeed, note that any $\begin{array}{|c|c|c|}\hline v_1 & ... & v_t \\ \hline \end{array} \in B_{(t)}$ that for any $T > t$, we have the following combinatorial R computation.

$$\begin{array}{|c|}\hline 1 \\ \hline \end{array}^{T} \otimes \begin{array}{|c|c|c|}\hline v_1 & ... & v_t \\ \hline \end{array} \simeq \begin{array}{|c|c|c|}\hline 1 & ... & 1 \\ \hline \end{array} \otimes \begin{array}{|c|}\hline u_1 \\ \hline \end{array} \otimes ... \otimes \begin{array}{|c|}\hline u_t \\ \hline \end{array} \in B_{(t)} \otimes B^{T}$$

which follows because the insertion $1^T \leftarrow v_1 ... v_t$ is a single row. So the fact the R_i are independent of the $m > M$ implies that they are also homogeneous stable states in their own right. $\square$

What the above proof is doing is picking an M large enough so that the first partition of the rigged configuration is built row by row, starting with the smallest (the interpretation with inverse KKR is that we remove one row at a time, starting with the largest).

2.7.3 Descent Coenergy and Time Evolution

We introduce a new statistic as follows.

Definition 2.7.18. Let $w \in B^n$. Suppose $\tau(w) = W_1 \otimes ... \otimes W_k$. Define the *descent coenergy*

$$\mathrm{DCE}_i(w) = |\mathrm{Row}_2(W_i \leftarrow W_{i+1})|.$$

Example 2.7.19. Let $w = 55434123211$. Then,

$$\tau(w) = \begin{array}{|c|c|}\hline 5 & 5 \\ \hline \end{array} \otimes \begin{array}{|c|}\hline 4 \\ \hline \end{array} \otimes \begin{array}{|c|c|}\hline 3 & 4 \\ \hline \end{array} \otimes \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline \end{array} \otimes \begin{array}{|c|}\hline 2 \\ \hline \end{array} \otimes \begin{array}{|c|c|}\hline 1 & 1 \\ \hline \end{array}$$

Then

$$W_1 \leftarrow W_2 = \begin{array}{|c|c|}\hline 4 & 5 \\ \hline 5 \\ \cline{1-1} \end{array}$$

$$W_2 \leftarrow W_3 = \begin{array}{|c|c|}\hline 3 & 4 \\ \hline 4 \\ \cline{1-1} \end{array}$$

$$W_3 \leftarrow W_4 = \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 3 & 4 \\ \cline{1-2} \end{array}$$

$$W_4 \leftarrow W_5 = \begin{array}{|c|c|c|}\hline 1 & 2 & 2 \\ \hline 3 \\ \cline{1-1} \end{array}$$

$$W_5 \leftarrow W_6 = \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 2 \\ \cline{1-1} \end{array}$$

Therefore, $\mathrm{DCE}_1(w) = 1$, $\mathrm{DCE}_2(w) = 1$, $\mathrm{DCE}_3(w) = 2$, $\mathrm{DCE}_4(w) = 1$, $\mathrm{DCE}_5(w) = 1$.

Remark 2.7.20. Suppose $w = u_1 ... u_t v_1 ... v_\ell$ with $u_1 \geq ... \geq u_t$, $v_1 \geq ... \geq v_\ell$ and $u_1 < v_1$. Lemma 2.7.27 implies that if w' is the reverse word of w, then $\mathrm{DCE}_1(w') = k$ if and only if k is the largest number such that there exists $i_1 < ... < i_k$ and $j_1 < ... < j_\ell$ such that $v_{j_\ell} v_{j_{\ell-1}} ... v_{j_1} u_{i_k} u_{i_{k-1}} ... u_{i_1}$ satisfies the perfect pairing condition. Moreover, we can always take $j_x = x$.

Example 2.7.21. Remark 2.7.20 implies that the DCE statistic can be used to measure the extent to which a pair of descent sequences satisfy the pairing condition. For example, consider the pair of descent sequences 322542. Note that this does not satisfy the pairing condition of length 3, despite the second descent sequence (542) having length 3. However, if we restrict the second descent sequence to only contain the first two letters, we have 32254 which satisfies the pairing condition of length 2. This is reflected in the fact that $\mathrm{DCE}_1(245223) = 2$. In other words, the DCE value of the reverse word gives the maximum length of the pairing condition that can be satisfied if we truncate the second descent sequence.

DCE stands for "descent coenergy" as it is the coenergy of $\tau(w)$. The reason we say it's descent coenergy instead of ascent coenergy is because typically the "real" object corresponding to a highest weight $w \in B^n$ is its reversed (Yamanouchi) word.

Remark 2.7.22. A corollary of Theorem 2.7.15 is that by Corollary 2.4.10, the number of ascent sequences of a highest weight $w \in B^n$ (resp descent sequences of the reverse word) is the same as the number of ascent sequences of $T_\infty(w)$. In particular, if $\tau(w) = W_1 \otimes ... \otimes W_x$ then we can write $\tau(T_\infty(w)) = W'_1 \otimes ... \otimes W'_x$. Similarily, we can write $\tau(T_\infty^{-1}(w)) = W''_1 \otimes ... \otimes W''_x$. This implies that the set of indices that DCE can be applied on is invariant under time evolution.

Many of our results and arguments will require the DCE statistic. As such, this subsection is dedicated to studying how the statistic behaves under inverse time evolution and related operations. The main results of this subsection that will be useful to us later are Theorem 2.7.23, Lemma 2.7.29, and Corollary 2.7.31.

In our proof of existence and uniqueness (modulo arbitrary distances between homogeneous stable states) of soliton decompositions for reverse Yamanouchi words, our construction shows that every ascent sequence (respectively descent sequence in the Yamanouchi word) is connected. Note that the connectedness Theorem 2.4.21 only gives sufficient conditions for connectedness. Thus, from Theorem 2.7.17 alone, we cannot conclude that adjacent solitons (including the space in between) in a homogeneous stable state satisfy the pairing condition.

However, it turns out that adjacent solitons in a homogeneous stable state do in fact satisfy the pairing condition as our next Theorem (2.7.23) demonstrates.

Theorem 2.7.23. Let $w \in B^n$ be a state. Write $\tau(w) = W_1 \otimes ... \otimes W_x$. Then $\mathrm{DCE}_i(w) = k$ for all $1 \le i < x$ if and only if w is a homogeneous stable state of degree k.

Proof. Note that $w \in B^n$ is a homogeneous stable state if and only if its essential part is preserved by inverse time evolution. Proceed by induction on the number of solitons, ℓ of w. If $\ell = 2$ the result is immediate. Suppose the result is true for some $\ell - 1$. Write $\mathrm{ess}(w) = V_1 \otimes ... \otimes V_\ell$. Let $u = \boxed{1}\boxed{...}\boxed{1}$ be the row with a large number of ones used in the definition of inverse time evolution. Consider the combinatorial R computation,

$$\begin{aligned}
&\boxed{1}\boxed{...}\boxed{1} \otimes V_1 \otimes ... \otimes V_{k-1} \\
&\simeq 1^{|V_1|} \otimes \boxed{1}\boxed{...}\boxed{1}\boxed{V_1} \otimes V_2 \otimes ... \otimes V_{k-1} \\
&\simeq 1^{|V_1|} \otimes \mathrm{RR}(\mathrm{Row}_2(V_1 \leftarrow V_2)) \otimes \tau(1...1\mathrm{RR}(\mathrm{Row}_1(V_1 \leftarrow V_2))) \otimes V_3 \otimes ... \otimes V_{k-1}
\end{aligned}$$

Note that if w is a homogeneous stable state, then $\mathrm{Row}_2(V_1 \leftarrow V_2) = V_1$. In particular,

$$|\mathrm{Row}_2(V_1 \leftarrow V_2)| = k.$$

This also implies that $\mathrm{Row}_1(V_1 \leftarrow V_2) = V_2$. Then we see that the rest of the combinatorial R computation is equivalent to computing $T_\infty^{-1}(\iota_{M,N}(\mathrm{RR}(V_2 \otimes ... \otimes V_{k-1})))$ which has less solitons. By induction, every $\mathrm{DCE}_i(w) = k$.

Conversely, if each $\mathrm{DCE}_i(w) = k$, then $\mathrm{DCE}_1(w) = k$ so that $|\mathrm{Row}_2(V_1 \leftarrow V_2)| = |V_1|$. This is only possible if every letter of V_1 was bumped. Hence we in fact have $\mathrm{Row}_2(V_1 \leftarrow V_2) = V_1$. The result then follows similarly to the above using induction. □

To proceed with our study of the DCE statistic and prove Lemmas 2.7.29 and 2.7.31, we need some technical lemmas which confirm some intuitions about Schensted insertion. These are Lemmas 2.7.24, 2.7.25, 2.7.26, and 2.7.27.

Lemma 2.7.24. Let $v_1 \leq ... \leq v_k$ and $u_1 \leq ... \leq u_\ell$. Consider the insertion

$$T = \begin{array}{|c|c|c|}\hline v_1 & ... & v_k \\ \hline\end{array} \leftarrow u_1 ... u_\ell.$$

The indices $i_1 < ... < i_r$ of bumped elements $v_{i_1}, ..., v_{i_r}$ satisfy the properties

(1) $u_j < v_{i_j}$

(2) If $u_j < v_x$ with $x < i_j$ then $x = i_p$ for some $p < j$.

Proof. Follows immediately from the definition of Schensted row insertion. □

It is important to note that there may be different index sets $i_1 < ... < i_r$ which give the same second row $\begin{array}{|c|c|c|}\hline v_{i_1} & ... & v_{i_r} \\ \hline\end{array}$ (in particular when v has non-standard content). The above lemma assumes the i_j are "leftmost" following the definition of Schensted insertion. We make the following observation

Lemma 2.7.25. Let $v_1 \leq ... \leq v_k$ and $u_1 \leq ... \leq u_\ell$. Also let $q_1 \leq ... \leq q_t$ such that there exists $i_1 < ... < i_k$ with each $q_{i_j} = v_j$. Consider the two tableaux $T = \begin{array}{|c|c|c|}\hline q_1 & ... & q_t \\ \hline\end{array} \leftarrow u_1 ... u_\ell$ and $T' = \begin{array}{|c|c|c|}\hline v_1 & ... & v_k \\ \hline\end{array} \leftarrow u_1 ... u_\ell$. Then, $|\mathrm{Row}_2(T)| \geq |\mathrm{Row}_2(T')|$. Moreover, for $i \leq |\mathrm{Row}_2(T')|$ we have $\mathrm{Row}_2(T)_i \leq \mathrm{Row}_2(T')_i$.

Proof. Let the second row of T be the elements $q_{x_1}, ..., q_{x_r}$ and the second row of T' be the elements $v_{y_1}, ..., v_{y_t}$ where the indices x_i, y_i satisfy the conditions in Lemma 2.7.24. Define

$$T'_m := \begin{array}{|c|c|c|}\hline q_1 & ... & q_t \\ \hline\end{array} \leftarrow u_1 ... u_{m-1}.$$

Note that $T'_1 = \begin{array}{|c|c|c|}\hline q_1 & ... & q_t \\ \hline\end{array}$. We show the stronger statement that if $j < t$ then each $x_j \leq i_{y_j}$ so that $q_{x_j} \leq q_{i_{y_j}} = v_{y_j}$. First note that if $t \geq 1$ then there exists $q_{i_{y_1}} > u_1$ and hence the second row of $T'_1 \leftarrow u_1$ has length 1 which demonstrates $r \geq 1$. Both $u_1 < q_{x_1}$ and $u_1 < v_{y_1}$. We have $v_{y_1} = q_{i_{y_1}}$. We cannot have $i_{y_1} < x_1$ due to condition (2) in Lemma 2.7.24. So $i_{y_1} \geq x_1$. Suppose that $x_j \leq i_{y_j}$ for all $1 \leq j \leq m-1 < t$. First note that $m \leq t$ so we can compute the inequality $x_{m-1} \leq i_{y_{m-1}} < i_{y_m}$. We have $u_m < v_{y_m} = q_{i_{y_m}}$. Also note that the first row of T'_m contains the letters q_s for $s > x_{m-1}$. In particular, it contains a copy of $v_{y_m} = q_{i_{y_m}}$ in its first row so that the second row of $T_m \leftarrow u_m$ has length one more than the second row of T_m. Hence, $r \geq m$. Note that both $u_m < q_{x_m}$ and $u_m < v_{y_m} = q_{i_{y_m}}$. By condition (2) of Lemma 2.7.24, if $i_{y_m} < x_m$ then $i_{y_m} = x_p$ for some $p < m$. We also see that $x_{m-1} \leq i_{y_{m-1}} < i_{y_m} = x_p$. Hence $x_{m-1} < x_p \implies m \leq p$. But we also have $m > p$. This is a contradiction. So $x_m \leq i_{y_m}$. The result follows. □

This lemma allows us to reduce Lemmas 2.7.26 and 2.7.27 to straightforward computations.

Lemma 2.7.26. Let $v_1 \leq ... \leq v_k$, $u_1 \leq ... \leq u_\ell$ and $q_1 \leq ... \leq q_\ell$. Moreover assume that each $q_i \leq u_i$. Define $T := v_1...v_k \leftarrow u_1...u_\ell$ and $T' := v_1...v_k \leftarrow q_1...q_\ell$. Then $|\text{Row}_2(T)| \leq |\text{Row}_2(T')|$.

Proof. Let $i_1 < ... < i_r$ be the indices corresponding to the bumped entries $v_{i_1}, ..., v_{i_r}$ when inserting $v_1...v_k \leftarrow u_1...u_\ell$ that satisfy the conditions of Lemma 2.7.24. Then we see that each $q_j \leq u_j < v_{i_j}$ for $j \leq r$. Therefore,

$$\begin{aligned} |\text{Row}_2(T)| &= |\text{Row}_2(v_1...v_k \leftarrow u_1...u_\ell)| \\ &= r \\ &= |\text{Row}_2(v_{i_1}...v_{i_r} \leftarrow u_1...u_\ell)| \\ &= |\text{Row}_2(v_{i_1}...v_{i_r} \leftarrow q_1...q_\ell)| \\ \text{(Lemma 2.7.25)} &\leq |\text{Row}_2(v_1...v_k \leftarrow q_1...q_k)| \\ &= |\text{Row}_2(T')| \end{aligned}$$

The result follows. □

Lemma 2.7.27. Let $v_1 \leq ... \leq v_k$ and $u_1 \leq ... \leq u_\ell$. Let $1 \leq r \leq k$. Then,

$$|\text{Row}_2(v_r v_{r+1}...v_k \leftarrow u_1...u_\ell)| = \min(|\text{Row}_2(v_1...v_k \leftarrow u_1...u_\ell)|, k - r + 1)$$

Proof. Let $h = |\text{Row}_2(v_1...v_k \leftarrow u_1...u_\ell)|$. We will show the result holds when $r = k-h+1$ (so that $h = k-r+1$). Let $i_1 < ... < i_h$ be the sequence of bumped indices when inserting $v_1...v_k \leftarrow u_1...u_\ell$. We note that then each $u_j < v_{i_j} \leq v_{r+j-1}$ for $j \leq h$ (due to $i_j \leq r + j - 1$). We then see using Schensted insertion that $|\text{Row}_2(v_r...v_k \leftarrow u_1...u_\ell)| = h$.

For $r < k - h + 1$ we note that from Lemma 2.7.25 we have,

$$|\text{Row}_2(v_r v_{r+1}...v_k \leftarrow u_1...u_\ell)| \geq |\text{Row}_2(v_{k-h+1}...v_k \leftarrow u_1...u_\ell)|$$

Hence,

$$\begin{aligned} |\text{Row}_2(v_r v_{r+1}...v_k \leftarrow u_1...u_\ell)| &\geq |\text{Row}_2(v_{k-h+1}...v_k \leftarrow u_1...u_\ell)| \\ &= h \\ &= |\text{Row}_2(v_1...v_k \leftarrow u_1...u_\ell)| \end{aligned}$$

However, another application of Lemma 2.7.25 (noting that $v_1...v_k$ contains $v_r...v_k$) gives us

$$|\text{Row}_2(v_1...v_k \leftarrow u_1...u_\ell)| \geq |\text{Row}_2(v_r...v_k \leftarrow u_1...u_\ell)|$$

Therefore

$$|\text{Row}_2(v_r...v_k \leftarrow u_1...u_\ell)| = |\text{Row}_2(v_1...v_k \leftarrow u_1...u_\ell)|$$

We also note that when $r \leq k - h + 1$ that $k - r + 1 \geq h$ so

$$\text{Row}_2(v_r...v_k \leftarrow u_1...u_\ell)| = \min(h, k - r + 1)$$

Now suppose that $r > k - h + 1$ so that $k - r + 1 < h$. We note that each $u_j < v_{i_j} \leq v_{r+j-1}$ for $r + j - 1 \leq k$. So by the definition of Schensted insertion, each v_{r+j-1} (with$r \leq r + j - 1 \leq k$) is bumped when inserting $v_r...v_k \leftarrow u_1...u_\ell$. Therefore, $|\text{Row}_2(v_r...v_k \leftarrow u_1...u_\ell)| = k - r + 1 < h$ as required. □

Note that Lemma 2.7.27 implies that the DCE statistic can be used to measure to what degree a pair of descent sequences satisfy the pairing condition (see Remark 2.7.20, Example 2.7.21).

Before proving Lemma 2.7.29, we need one more technical lemma.

Lemma 2.7.28. Let $q_1 \leq ... \leq q_t$ and $v_1 \leq ... \leq v_k$. Let $T = q_1...q_t \leftarrow v_1...v_k$. Suppose $v_1, ..., v_r$ induce a bump, and each v_j, $j > r$ does not. Now let $1 \leq i_1 < ... < i_x \leq k$. Then,

$$|\mathrm{Row}_2(\mathrm{Row}_2(T) \leftarrow v_{i_1}...v_{i_x})| \geq \#\{i_j \leq r\}$$

Proof. Let $\mathrm{Row}_2(T) = q_{l_1}...q_{l_y}$ satisfying the conditions of Lemma 2.7.24. Suppose $\{i_j \leq r\} = \{i_1, ..., i_\alpha\}$. Because each v_{i_j} induces a bump, there exists s_j with $v_{i_j} < q_{l_{s_j}}$. Moreover, $l_{s_1} < l_{s_2} < ... < l_{s_\alpha}$. Note that then $q_{l_{s_1}}...q_{l_{s_\alpha}}$ is a subword of $q_{l_1}...q_{l_y}$. So by Lemma 2.7.25,

$$\begin{aligned}\#\{i_j \leq r\} &= \alpha \\ &= |\mathrm{Row}_2(q_{l_{s_1}}...q_{l_{s_\alpha}} \leftarrow v_{i_1}...v_{i_\alpha})| \\ &\leq |\mathrm{Row}_2(q_{l_1}...q_{l_y} \leftarrow v_{i_1}...v_{i_\alpha})| \\ &\leq |\mathrm{Row}_2(q_{l_1}...q_{l_y} \leftarrow v_{i_1}...v_{i_x})| \\ &= |\mathrm{Row}_2(\mathrm{Row}_2(T) \leftarrow v_{i_1}...v_{i_x})|\end{aligned}$$

as required. □

Now we can prove the second main result of this subsection.

Lemma 2.7.29. Let $v_1 \leq ... \leq v_k$, $u_1 \leq ... \leq u_\ell$ and $q_1 \leq ... \leq q_t$. Consider the image under combinatorial R,

$$\begin{aligned}&\begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & q_1 & ... & q_t \\ \hline\end{array} \otimes v_1 \otimes ... \otimes v_k \otimes u_1 \otimes ... \otimes u_\ell \\ &\simeq v'_1 \otimes ... \otimes v'_{k'} \otimes 1 \otimes ... \otimes 1 \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & q'_1 & ... & q'_{t'} \\ \hline\end{array} \otimes u_1 \otimes ... \otimes u_\ell \\ &\simeq v'_1 \otimes ... \otimes v'_{k'} \otimes 1 \otimes ... \otimes 1 \otimes u'_1 \otimes ... \otimes u'_{\ell'} \otimes 1 \otimes ... \otimes 1 \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & q''_1 & ... & q''_{t''} \\ \hline\end{array}\end{aligned}$$

Let $T := v_1...v_k \leftarrow u_1...u_\ell$, $T' := v'_1...v'_{k'} \leftarrow 1...1u'_1...u'_{\ell'}$ and $T'' := 1...1q_1...q_t \leftarrow v_1...v_k$. Then, $|\mathrm{Row}_2(T')| \geq \min(|\mathrm{Row}_2(T)|, |\mathrm{Row}_2(T'')|)$.

Proof. Consider the second row of the insertion $\begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & q_1 & ... & q_t \\ \hline\end{array} \leftarrow v_1...v_k$, and say it has entries $q_{i_1}, ..., q_{i_r}$ where $i_1 < ... < i_r$ satisfy the conditions in Lemma 2.7.24. Suppose the insertion of v_j for $j > \alpha$ does not induce a bump. Let $L = |\mathrm{Row}_2(T')|$ and define $T''' = 1...1q'_1...q'_{t'} \leftarrow u_1...u_\ell$. By Lemma 2.7.7, we have

$$L = |\mathrm{Row}_2(q_{i_1}...q_{i_r} \leftarrow 1^{k-\alpha}\mathrm{Row}_2(T'''))|$$

Note that $q'_1...q'_{t'}$ contains $v_1...v_k$. Hence, by Lemma 2.7.25, we have $|\mathrm{Row}_2(T''')| \geq |\mathrm{Row}_2(T)|$. Also, each entry $\mathrm{Row}_2(T''')_j \leq \mathrm{Row}_2(T)_j$ for $j \leq |\mathrm{Row}_2(T)|$. Then, by Lemma 2.7.26 we have

$$|\mathrm{Row}_2(q_{i_{k-\alpha+1}}...q_{i_r} \leftarrow \mathrm{Row}_2(T'''))| \geq |\mathrm{Row}_2(q_{i_{k-\alpha+1}}...q_{i_r} \leftarrow \mathrm{Row}_2(T))|$$

Therefore,

$$\begin{aligned}
|\mathrm{Row}_2(q_{i_1}...q_{i_r} \leftarrow 1^{k-\alpha}\mathrm{Row}_2(T'''))| &= k-\alpha+|\mathrm{Row}_2(q_{i_{k-\alpha+1}}...q_r \leftarrow \mathrm{Row}_2(T'''))| \\
&\geq k-\alpha+|\mathrm{Row}_2(q_{i_{k-\alpha+1}}...q_{i_r} \leftarrow \mathrm{Row}_2(T))| \\
(\text{Lemma 2.7.27}) &\geq k-\alpha+\min(|\mathrm{Row}_2(q_{i_1}...q_{i_r} \leftarrow \mathrm{Row}_2(T))|, r-(k-\alpha))
\end{aligned}$$

Let the second row $\mathrm{Row}_2(T) = v_{l_1}...v_{l_s}$ where $l_1 < ... < l_s$ satisfy the conditions of Lemma 2.7.24. Let $\gamma = \#\{l_j \leq \alpha\}$. Note that $s-\gamma = \#\{l_j > \alpha\} \leq \#\{j > \alpha\} = k-\alpha$ so that $\gamma \geq \alpha - k + s$. By Lemma 2.7.28, we have

$$|\mathrm{Row}_2(q_{i_1}...q_{i_r} \leftarrow \mathrm{Row}_2(T))| = |\mathrm{Row}_2(\mathrm{Row}_2(T'') \leftarrow v_{l_1}...v_{l_s})| \geq \gamma$$

Hence,

$$\begin{aligned}
&|\mathrm{Row}_2(q_{i_1}...q_{i_r} \leftarrow 1^{k-\alpha}\mathrm{Row}_2(T'''))| \\
&\geq k-\alpha+\min(|\mathrm{Row}_2(q_{i_1}...q_{i_r} \leftarrow \mathrm{Row}_2(T))|, r-(k-\alpha)) \\
&\geq \min(k-\alpha+\gamma, r-k+\alpha+k-\alpha) \\
&\geq \min(k-\alpha+\alpha-k+s, r) \\
&= \min(s, r)
\end{aligned}$$

The result follows. $\square$

As a consequence of the above Lemma 2.7.29, we see that under inverse time evolution, the minimum DCE value cannot decrease. Respectively, under time evolution the minimum DCE value cannot increase. The last main result of this section is Corollary 2.7.31. First we prove Lemma 2.7.30 from which Corollary 2.7.31 is derived.

Lemma 2.7.30. Let $w \in B^n$. Suppose $\mathrm{DCE}_i(w) > M$ for $i \leq r$. Then, $\mathrm{DCE}_i(T_\infty^{-1}(w)) > M$ for $i < r$.

Proof. Write $w = w_1^{(1)}w_2^{(1)}...w_{\ell_1}^{(1)}...w_1^{(x)}...w_{\ell_x}^{(x)}$ where

$$\tau(w) = \begin{array}{|c|c|c|}\hline w_1^{(1)} & ... & w_{\ell_1}^{(1)} \\ \hline\end{array} \otimes ... \otimes \begin{array}{|c|c|c|}\hline w_1^{(x)} & ... & w_{\ell_x}^{(x)} \\ \hline\end{array} = W_1 \otimes ... \otimes W_x$$

That is each $w_i^{(a)} < w_j^{(a)}$ for $i < j$ and $w_{\ell_i}^{(i)} > w_1^{(i+1)}$. Note that by Lemma 2.7.7,

$$\begin{aligned}
T_\infty^{-1}(w) &= \overbrace{1...1w'^{(1)}_1...w'^{(1)}_{\ell'_1}}^{W'_1}\overbrace{1...1w'^{(2)}_1...w'^{(2)}_{\ell'_2}}^{W'_2}\overbrace{1...1}^{W'_3}\ \ ...\ \overbrace{1...1w'^{(x-1)}_1...w'^{(x-1)}_{\ell'_{x-1}}}^{W'_{x-1}}\overbrace{1...1}^{W'_x} \\
&= \underbrace{1...1}_{W_1}\underbrace{w'^{(1)}_1...w'^{(1)}_{\ell'_1}1...1}_{W_2}\underbrace{w'^{(2)}_1...w'^{(2)}_{\ell'_2}1...1}_{W_3}\ \ ...\ \underbrace{1...1}_{W_{x-1}}\underbrace{w'^{(x-1)}_1...w'^{(x-1)}_{\ell'_{x-1}}1...1}_{W_x}
\end{aligned}$$

The under braces represent the positions of the old ascent sequences and the over braces show the ascent sequences of $T_\infty^{-1}(w)$ itself. Now, let $i < r$. We wish to show that

$$|\mathrm{Row}_2(1...1w'^{(i)}_1...w'^{(i)}_{\ell'_i} \leftarrow 1...1w'^{(i+1)}_1...w'^{(i+1)}_{\ell'_{i+1}})| > M$$

Note that

$$|\mathrm{Row}_2(1...1w'^{(i)}_1...w'^{(i)}_{\ell'_i} \leftarrow 1...1w'^{(i+1)}_1...w'^{(i+1)}_{\ell'_{i+1}})| = |\mathrm{Row}_2(w'^{(i)}_1...w'^{(i)}_{\ell'_i} \leftarrow 1...1w'^{(i+1)}_1...w'^{(i+1)}_{\ell'_{i+1}})|$$

Consider the state (identifying W_j with $\mathrm{RR}(W_j)$ for compactness of notation)

$$\begin{array}{|c|c|c|}\hline 1 & ... & 1 \\ \hline\end{array} \otimes w$$
$$\simeq 1...1{w'}_1^{(1)}...{w'}_{\ell'_1}^{(1)}...1...1{w'}_1^{(i-1)}...{w'}_{\ell'_{i-1}}^{(i-1)} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & q_1 & ... & q_t \\ \hline\end{array} \otimes W_{i+1} \otimes W_{i+2} \otimes ... \otimes W_x.$$

Note that $q_1...q_t$ contains the subsequence $w_1^{(i)}...w_{\ell_i}^{(i)}$ so that by Lemma 2.7.25,

$$\begin{aligned}|\mathrm{Row}_2(1...1q_1...q_t \leftarrow w_1^{(i+1)}...w_{\ell_{i+1}}^{(i+1)})| &\geq |\mathrm{Row}_2(w_1^{(i)}...w_{\ell_i}^{(i)} \leftarrow w_1^{(i+1)}...w_{\ell_{i+1}}^{(i+1)})| \\ &= |\mathrm{Row}_2(W_i \leftarrow W_{i+1})|.\end{aligned}$$

Then, by Lemma 2.7.29,

$$\begin{aligned}&|\mathrm{Row}_2({w'}_1^{(i)}...{w'}_{\ell'_i}^{(i)} \leftarrow 1...1{w'}_1^{(i+1)}...{w'}_{\ell'_{i+1}}^{(i+1)})| \\ &\geq \min(|\mathrm{Row}_2(W_{i+1} \leftarrow W_{i+2})|, |\mathrm{Row}_2(1...1q_1...q_t \leftarrow W_{i+1})|) \\ &\geq \min(|\mathrm{Row}_2(W_{i+1} \leftarrow W_{i+2})|, |\mathrm{Row}_2(W_i \leftarrow W_{i+1})|) \\ &= \min(\mathrm{DCE}_{i+1}(w), \mathrm{DCE}_i(w)) \\ &> M\end{aligned}$$

where the last line follows because $i < i+1 \leq r$. □

Now we can prove the last main result of this subsection.

Corollary 2.7.31. Let $w \in B^n$ with $\tau(w) = W_1 \otimes ... \otimes W_x$. Suppose $\mathrm{DCE}_i(w) > M$ for all $1 \leq i < x$. Then, $\mathrm{DCE}_i(T_\infty^{-1}(w)) > M$ for all $1 \leq i < x$.

Proof. By Lemma 2.7.30, it suffices to show that $\mathrm{DCE}_{x-1}(T_\infty^{-1}(w)) > M$. Note that the length of the subsequence of W'_{x-1} which does not contain any ones is at least the length by Lemma 2.7.25 $|\mathrm{Row}_2(W_{x-1} \leftarrow W_x)| = \mathrm{DCE}_{x-1}(w) > M$. Because W'_x consists of a large number of ones, we have $|\mathrm{Row}_2(W'_{x-1} \leftarrow W'_x)| = |W'_{x-1}| > M$. □

2.7.4 Main Results of Section 2.7

The main theorems we wish to prove are Theorems 2.7.43 (as well as the related Theorem 2.7.42) and 2.7.35 and We remark that we will need Theorem 2.7.35 to prove Theorem 2.7.43. So we must prove Theorem 2.7.35 first. To prove Theorem 2.7.35, we first prove a version of it that assumes $w \in B^n$ has a large number of ones at the beginning and end.

Theorem 2.7.32. Let $w \in B^n$ (that begins and ends with a large number of ones) be highest weight with $\tau(w) = W_1 \otimes ... \otimes W_x$. Let $X = (\nu, J)$ be its corresponding rigged configuration. Then,

$$\min_{1 \leq i < x} \mathrm{DCE}_i(w) = \min_{1 \leq i \leq \ell(\nu^{(1)})} \nu_i^{(1)}$$

Proof. By Theorem 2.7.17, there exists some soliton decomposition $w' := T_\infty^{(m)}(w)$ of w. By Lemma 2.7.30 we must have $\min_{1 \leq i < x} \mathrm{DCE}_i(w') \leq \min_{1 \leq i < x} \mathrm{DCE}_i(w)$. The degrees of the homogeneous stable states agree with the rows of $\nu^{(1)}$. Moreover, by Theorem 2.7.23, writing $\tau(w') = W_1 \otimes ... \otimes W_x$

and W_i' to be the part only containing the letters greater than one, we have each $\mathrm{DCE}_i(w') = |W_i'|$. Note that because we have a soliton decomposition (hence arbitrary distance between homogeneous stable states) that we get $\mathrm{DCE}_i(w') = |W_i'|$ for even the last ascent sequence in each homogeneous stable state. We also have $\mathrm{DCE}_1(w') \geq ... \geq \mathrm{DCE}_{x-1}(w')$. We have that $\mathrm{DCE}_{x-1}(w')$ is equal to the degree of the right most homogeneous stable state, which is equal to the smallest row of $\nu^{(1)}$ by Theorem 2.7.17. It hence follows that

$$\min_{1\leq i\leq \ell(\nu^{(1)})} \nu_i^{(1)} = \min_{1\leq i<x} \mathrm{DCE}_i(w') \leq \min_{1\leq i<x} \mathrm{DCE}_i(w)$$

We also note that by Theorem 2.4.21 that

$$\min_{1\leq i<x} \mathrm{DCE}_i(w) \leq \min_{1\leq i\leq \ell(\nu^{(1)})} \nu_i^{(1)}$$

The result follows. □

To prove the case for general $w \in B^n$ requires a bit of care. As far as rigged configurations are concerned, appending and prepending a large number of ones to both ends of the corresponding crystal element does not change the rigged configuration in a non-trivial way. Adding ones to the beginning changes the first rigged partition, and leaves the others the same. However, the first corigged partition stays the same. Adding ones to the end has the opposite effect: the first corigged partition changes, but the first rigged partition does not. All other rigged/corigged partitions do not change. This observation is recorded in the following lemma.

Lemma 2.7.33. Let $w \in B^n$ be highest weight and $M, N \in \mathbb{N}$. Consider $X' = \Phi(\boxed{1}^{M} \otimes w \otimes \boxed{1}^{N})$ and $X = \Phi(w)$. Then, the proper configurations of X and X' are the same. Moreover, $X^{(i)} = X'^{(i)}$ for $i > 1$ and, $X'^{(1)} = \{(x, \alpha + N) : (x, \alpha) \in X^{(1)}\}$. If $M = 0$, then $\tilde{X}^{(1)} = \tilde{X'}^{(1)}$.

Remark 2.7.34. The type of reasoning done in Lemma 2.7.33 is how Theorem 2.7.15 is proved.

Define $\iota_{M,N} : w \mapsto 1^M \otimes w \otimes 1^N$. Then define $\jmath_{M,N} = \Phi \circ \iota_{M,N} \circ \Phi^{-1}$. Note that Lemma 2.7.33 gives a description of $\jmath_{M,N}$. Clearly, both $\iota_{M,N}$ and $\jmath_{M,N}$ are invertible if M, N are known.

Theorem 2.7.35. Theorem 2.7.32 is true for any highest weight $w \in B^n$. In particular, it does not need to start and end with a large number of ones.

Proof. Write $\tau(w) = W_1 \otimes ... \otimes W_x$. Let M, N be large. Because w is highest weight, the reverse word is Yamanouchi. In particular, $W_x = \tau(1^\ell)$ for some ℓ. Note that $\tau(\iota_{M,N}(w)) = W_1' \otimes W_2 \otimes ... \otimes W_{x-1} \otimes W_x'$ where $W_1' = 1^M \leftarrow W_1$ and W_x' consists of $\ell + N$ ones. We observe that, $\mathrm{DCE}_i(w) = \mathrm{DCE}_i(\iota_{M,N}(w))$ for $i < x-1$. Because w is highest weight, $W_{x-1} = \tau(1^a 2^b)$ for some a, b. Also, we must have $|W_x| = \ell \geq b$. Then,

$$\begin{aligned} \mathrm{DCE}_{x-1}(w) &= |\mathrm{Row}_2(W_{x-1} \leftarrow \tau(1^\ell))| \\ &= b \\ &= |\mathrm{Row}_2(W_{x-1} \leftarrow \tau(1^{\ell+N})| \\ &= \mathrm{DCE}_{x-1}(\iota_{M,N}(w)) \end{aligned}$$

The result follows from the fact that the proper configuration is preserved under $\jmath_{M,N}$. □

The argument in the proof of Theorem 2.7.32 essentially uses the fact that it follows from Theorem 2.4.21 if the word is an asymptotic state of time evolution (in other words, a soliton decomposition).

We wish to apply the results of subsection 2.7.3 to prove Theorems 2.7.43 and 2.7.42. Similar to Theorem 2.7.32, it is helpful to adopt the assumption in subsection 2.7.2 that words $w \in B^n$ begin and end with a large number of ones. This assumption also implies that $\mathrm{raise}_1(w)$ is always defined if w has any letter greater than 1. It also lets us assume that time evolution, and inverse time evolution are in fact inverses. However, we wish for Theorem 2.7.43 to hold for any highest weight $w \in B^n$. Therefore, we must justify why we can make it.

Suppose $F_n : B^n \to B^n$ is any family of functions satisfying $F_{n+M+N} \circ \iota_{M,N} = \iota_{M,N} \circ F_n$. The above lemma implies that if G_n is a family of functions on rigged configurations which commutes with $\jmath_{M,N}$ in the same manner as above and $\Phi \circ F_{n+N+M} = G_{n+N+M} \circ \Phi$ on any highest weight word $w \in B^{n+M+N}$ where the first M letters and last N letters are 1 (and it is still highest weight when these 1s are removed), then in fact $\Phi \circ F_n = G_n \circ \Phi$ for all n.

Indeed, note that

$$\begin{aligned}
&\Phi \circ F_{n+N+M} = G_{n+M+N} \circ \Phi \\
&\implies \Phi \circ \iota_{N,M} \circ F_n \circ \iota_{N,M}^{-1} = \jmath_{N,M} \circ G_n \circ \jmath_{N,M}^{-1} \circ \Phi \\
&\implies \jmath_{M,N} \circ \Phi \circ F_n \circ \iota_{N,M}^{-1} = \jmath_{N,M} \circ G_n \circ \Phi \circ \iota_{N,M}^{-1} \\
&\implies \Phi \circ F_n = G_n \circ \Phi
\end{aligned}$$

As a result of the above discussion, in this subsection we assume all words $w \in B^n$ have a large number of ones appended to both the left and the right (unless otherwise stated) as doing so does not affect Theorems 2.7.43 or 2.7.42.

Let us give a general game plan for how we will prove Theorems 2.7.43 and 2.7.42. First let us understand what we are trying to prove. Define the following perturbations on rigged configurations.

Definition 2.7.36. Let $X = (\nu, J)$ be a rigged configuration. Let $\rho \in \nu^{(1)}$, and let

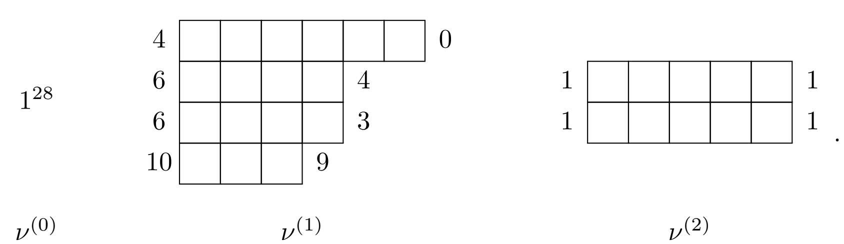
$$\alpha = \max_{1 \le i \le \ell(\nu^{(1)})} (x : (\rho, x) \in X^{(1)}).$$

Define $X^{\sharp(\rho)} := Y$ where X, Y have the same configuration, $Y^{(i)} = X^{(i)}$ for $i > 1$ and

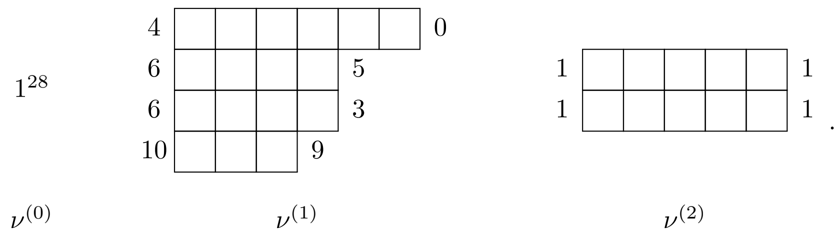
$$Y^{(1)} = (X^{(1)} \setminus \{(\rho, \alpha)\}) \cup \{(\rho, \alpha + 1)\}.$$

If $\rho = \min_{1 \le i \le \ell(\nu^{(1)})} \nu_i^{(1)}$, then we write $X^{\sharp} := X^{\sharp(\rho)}$.

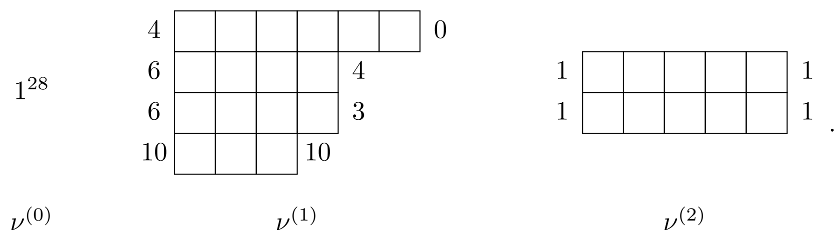
Example 2.7.37. Suppose X is the following rigged configuration:



Then $X^{\sharp(4)}$ is depicted below,



We also depict $X^{\sharp}$ below,



Theorem 2.7.42 says that if $\mathrm{DCE}_1(w) \leq \mathrm{DCE}_2(w)$, then $\Phi(\mathrm{raise}_1(w)) = \Phi(w)^{\sharp(\mathrm{DCE}_1(w))}$. To describe the more powerful Theorem 2.7.43 we first define the specific raise operation as follows.

Definition 2.7.38. Let $w \in B^n$ be a word with $\tau(w) = W_1 \otimes \ldots \otimes W_x$. Let $i_0 = \min_{1 \leq i < x}(i : \mathrm{DCE}_i(w) = \min_{1 \leq j < x} \mathrm{DCE}_j(w))$. Then define $\mathrm{raise}(w) := \mathrm{raise}_{i_0}(w)$.

In other words, $\mathrm{raise} = \mathrm{raise}_{i_0}$ where i_0 is the left-most minimizing index of DCE. Theorem 2.7.43 says that $\Phi(\mathrm{raise}(w)) = \Phi(w)^{\sharp}$. In particular, note that Theorem 2.7.43 does not require $\mathrm{DCE}_1(w) \leq \mathrm{DCE}_2(w)$.

Now we are ready to describe our game plan for proving these theorems. Using tableau combinatorics, we show Theorem 2.7.43 for when DCE is minimized at index 1. The general result then follows from a rigged configuration argument. Additionally, we remark that the proof of Theorem 2.7.42 is similar to that of Theorem 2.7.43, except the follow-up rigged configuration argument at the end is not needed. In a sense, Theorem 2.7.42 is an easy generalization of the halfway point to proving Theorem 2.7.43. Thus, we will largely discuss the strategy for proving Theorem 2.7.43.

The general strategy to prove the result when DCE is minimized at 1 is to show that raise_1 commutes with inverse time evolution (Lemma 2.7.40). The idea is that after applying inverse time evolution sufficiently many times (we will see later that only one application is necessary), then raise_1 "obviously" does what we want it to on the corresponding rigged configuration (Lemma 2.7.41). The result then follows by noting how time evolution and inverse time evolution change riggings in a way which completely depends on the configuration (and that they are inverses of each other). This is the most difficult step, and hence it would not be wrong to consider Lemma 2.7.40 the main theorem of this section with Theorem 2.7.43 being the corollary.

To show raise_1 commutes with inverse time evolution, we want to show that the insertion of the first two tensor factors of $\tau(T_\infty^{-1}(w))$ and $\tau(T_\infty^{-1}(\mathrm{raise}_1(w)))$ are the same. This is because raise_1 is defined as a refactorization of the first two tensor factors. After this, our goal is to apply Corollary 2.6.3. This amounts to showing that the length of the first tensor factor of $\tau(T_\infty^{-1}(\mathrm{raise}_1(w)))$ is one less than the first tensor factor of $\tau(T_\infty^{-1}(w))$ and that the length of the second tensor factor of $\tau(T_\infty^{-1}(\text{raise}_1(w)))$ is one more than the second tensor factor of $\tau(T_\infty^{-1}(w))$.

The rigged configuration argument is simple. If we perform raise_{i_0} where i_0 is the smallest minimizing index of DCE, then all smaller indices $i < i_0$ satisfy $\text{DCE}_i(w) > \text{DCE}_{i_0}(w)$. In particular, the behaviour of the KKR bijection is controlled until a row of length $\text{DCE}_i(w) > \text{DCE}_{i_0}(w)$ is added by the connectedness Theorem 2.4.21. Adding a row of this length makes the row of length $\text{DCE}_{i_0}(w)$ non-singular. Hence, it is not considered in the KKR bijection. It follows that the remaining steps of the KKR bijection are "essentially the same" before and after applying raise_{i_0}.

Now, we proceed with our plan. We wish to show that if DCE is minimized at 1, then the plactic product of the first two tensor factors of w and the plactic product of the first two tensor factors of $T_\infty^{-1}(w)$ are the same. This information is partially encoded by $\text{DCE}_1(w)$ (in particular, this gives the length of the second row of the plactic product). With some additional information, we can show the shape of the plactic product of the first two tensor factors stay the same. It turns out that this is sufficient for showing the fillings are also identical. Hence, we first prove the following lemma.

Lemma 2.7.39. Let $w = w_1...w_n$ be a word. Write $\tau(w) = W_1 \otimes ... \otimes W_x$. If $t \leq 2$ or $t \geq 3$ with $\text{DCE}_1(w) \leq \text{DCE}_2(w)$ then $\text{DCE}_1(w) = \text{DCE}_1(T_\infty^{-1}(w))$.

Proof. Let $\tau(w) = W_1 \otimes ... \otimes W_x$ and $\tau(T_\infty^{-1}(w)) = W_1' \otimes ... \otimes W_x'$. First suppose that $x \leq 2$. Then because w has a large number of 1s at the beginning and end, we have $W_x = 11...1$. It follows that $\text{DCE}_1(w)$ is equal to the number of entries of W_1 greater than 1. Using Lemma 2.7.7, we see that W_x' is also compirsed of only ones, and the number of entries of W_1' greater than 1 is the same as the number of entries of W_1 greater than 1. Hence, the result follows if $x \leq 2$.

Now suppose that $x > 2$ and suppose $W_1 = w_1...w_a$. Then,

$$\begin{array}{|c|c|c|}\hline 1 & ... & 1 \\ \hline \end{array} \otimes \text{RR}(W_1) \simeq 1^{\otimes|W_1|} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & w_1 & ... & w_a \\ \hline \end{array}$$

So, by Lemma 2.7.29, we have $\text{DCE}_1(T_\infty^{-1}(w)) \geq \min(\text{DCE}_2(w), \text{DCE}_1(w)) = \text{DCE}_1(w)$. However, we also note that the number of entries in W_1' greater than 1 is equal to $\text{DCE}_1(w)$. So,

$$\text{DCE}_1(w) \geq \text{DCE}_1(T_\infty^{-1}(w)) \geq \text{DCE}_1(w)$$

which shows that $\text{DCE}_1(w) = \text{DCE}_1(T_\infty^{-1}(w))$ as required. □

In particular, note the above lemma holds when $\text{DCE}_1(w) = \min_{1 \leq i < x} \text{DCE}_i(w)$. Now, we execute the most difficult part of our game plan by proving the following lemma.

Lemma 2.7.40. Let $w \in B^n$ with $\tau(w) = W_1 \otimes ... \otimes W_x$ and either $x \leq 2$ or $x > 2$ with $\text{DCE}_1(w) \leq \text{DCE}_2(w)$. Then

$$\text{raise}_1(T_\infty^{-1}(w)) = T_\infty^{-1}(\text{raise}_1(w))$$

Proof. First suppose that $x = 2$ (when $x = 1$ there can be no refactorization, for w consists entirely of ones). Let $w_1...w_\ell$ be the entries of W_1 greater than 1. Write $w = 1^N w_1...w_\ell 1^M$. We see that $\text{raise}_1(w) = 1^{N-1} w_1...w_\ell 1^{M+1}$. Also, with Lemma 2.7.7 we have $T_\infty^{-1}(w) = 1^{N+\ell} w_1...w_\ell 1^{M-\ell}$ so that $\text{raise}_1(T_\infty^{-1}(w)) = 1^{N+\ell-1} w_1...w_\ell 1^{M-\ell+1}$. But we also see that

$$T_\infty^{-1}(\text{raise}_1(w)) = 1^{N-1+\ell} w_1...w_\ell 1^{M+1-\ell}.$$

So the result holds when $x = 2$. Now suppose that $x \geq 3$.

Write $W_1 = v_1...v_k$, $W_2 = u_1...u_\ell$ and $W_3 = \alpha_1...\alpha_s$. Consider the combinatorial R computation

$$\begin{aligned}
&\begin{array}{|c|c|c|}\hline 1 & ... & 1 \\ \hline \end{array} \otimes v_1...v_k u_1...u_\ell \alpha_1...\alpha_s \\
&\simeq 1^k \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & v_1 & ... & v_k \\ \hline \end{array} \otimes u_1...u_\ell \alpha_1...\alpha_s \\
&\simeq 1^k \otimes v_1'...v_{k'}' 1...1 \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & q_1 & ... & q_t \\ \hline \end{array} \otimes \alpha_1...\alpha_s \\
&\simeq 1^k \otimes v_1'...v_{k'}' 1...1 u_1'...u_{\ell'}' 1...1 \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & q_1' & ... & q_{t'}' \\ \hline \end{array}
\end{aligned}$$

Where $v_1' > 1, u_1' > 1$. Let $\tau(\text{raise}_1(w)) = \overline{W}_1 \otimes ... \otimes \overline{W}_x$ with $\overline{W}_1 = \overline{v}_1...\overline{v}_{k-1}$ and $\overline{W}_2 = \overline{u}_1...\overline{u}_{\ell+1}$. Note that $\overline{W}_j = W_j$ for $j > 2$. We again compute using combinatorial R

$$\begin{aligned}
&\begin{array}{|c|c|c|}\hline 1 & ... & 1 \\ \hline \end{array} \otimes \overline{v}_1...\overline{v}_{k-1} \overline{u}_1...\overline{u}_{\ell+1} \alpha_1...\alpha_s \\
&\simeq 1^{k-1} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & \overline{v}_1 & ... & \overline{v}_{k-1} \\ \hline \end{array} \otimes \overline{u}_1...\overline{u}_{\ell-1} \alpha_1...\alpha_s \\
&\simeq 1^{k-1} \otimes \overline{v}_1'...\overline{v}_{\overline{k}'}' 1...1 \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & \overline{q}_1 & ... & \overline{q}_t \\ \hline \end{array} \otimes \alpha_1...\alpha_s \\
&\simeq 1^{k-1} \otimes \overline{v}_1'...\overline{v}_{\overline{k}'}' 1...1 \overline{u}_1'...\overline{u}_{\overline{\ell}'}' 1...1 \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & \overline{q}_1' & ... & \overline{q}_{\overline{t}'}' \\ \hline \end{array}
\end{aligned}$$

Where $\overline{v}_1' > 1, \overline{u}_1' > 1$. We observe that by Lemma 2.7.7, both $k' = \overline{k}' = \text{DCE}_1(w)$ $(*)$. However, there are fewer 1s to the left of $\overline{v}_1'$ than there are to the left of v_1'. Note that up to appending some 1s to the left we have,

$$\begin{aligned}
\begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & \overline{q}_1 & ... & \overline{q}_t \\ \hline \end{array} &= 1...1\text{Row}_1(\overline{v}_1...\overline{v}_{k-1} \leftarrow \overline{u_1}...\overline{u}_{\ell+1}) \\
&= 1...1\text{Row}_1(\overline{W}_1 \leftarrow \overline{W}_2) \\
(\text{due to } W_1 \leftarrow W_2 = W_1' \leftarrow W_2') &= 1...1\text{Row}_1(W_1 \leftarrow W_2) \\
&= 1...1\text{Row}_1(v_1...v_k \leftarrow u_1...u_\ell) \\
&= \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & q_1 & ... & q_t \\ \hline \end{array}
\end{aligned}$$

By Lemma 2.7.7 it follows that

$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & q_1' & ... & q_{t'}' \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & \overline{q}_1' & ... & \overline{q}_{\overline{t}'}' \\ \hline \end{array}$$

and that the number of ones following $u_{\ell'}'$ is equal to the number of ones following $\overline{u}_{\overline{\ell}'}'$. Let this number be N_0. This also implies that the number of letters in $1^k v_1'...v_{k'}' 1...1 u_1'...u_{\ell'}'$ and $1^{k-1}\overline{v}_1'...\overline{v}_{\overline{k}'}' 1...1 \overline{u}_1'...\overline{u}_{\overline{\ell}'}'$ are the same (because $1...1q_1'...q_{t'}' = 1...1\overline{q}_1'...\overline{q}_{t'}'$). Let the common number of letters in each string be r. Define

$$\begin{aligned}
T &:= \begin{array}{|c|c|c|}\hline 1 & ... & 1 \\ \hline \end{array} \leftarrow v_1...v_k u_1...u_\ell \alpha_1...\alpha_s \\
&= \begin{array}{|c|c|c|}\hline 1 & ... & 1 \\ \hline \end{array} \leftarrow \overline{v}_1...\overline{v}_{k-1} \overline{u}_1...\overline{u}_{\ell-1} \alpha_1...\alpha_s
\end{aligned}$$

By the definition of combinatorial R, we have

$$\begin{aligned}
T &= (\emptyset \leftarrow 1^k v_1'...v_{k'}' 1...1 u_1'...u_{\ell'}' 1^{N_0}) \leftarrow \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & q_1' & ... & q_{t'}' \\ \hline \end{array} \\
&= (\emptyset \leftarrow 1^k v_1'...v_{k'}' 1...1 u_1'...u_{\ell'}') \leftarrow (1^{N_0} \leftarrow \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & q_1' & ... & q_{t'}' \\ \hline \end{array})
\end{aligned}$$

Where the second line follows from associativity of the plactic monoid. Similarily, we have

$$\begin{aligned} T &= (\emptyset \leftarrow 1^{k-1}\overline{v}'_1...\overline{v}'_{\overline{k}'}1...1\overline{u}'_1...\overline{u}'_{\overline{\ell}'}1^{N_0}) \leftarrow \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & \overline{q}'_1 & ... & \overline{q}'_{\overline{t}'} \\ \hline\end{array} \\ &= (\emptyset \leftarrow 1^{k-1}\overline{v}'_1...\overline{v}'_{\overline{k}'}1...1\overline{u}'_1...\overline{u}'_{\overline{\ell}'}) \leftarrow (1^{N_0} \leftarrow \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & \overline{q}'_1 & ... & \overline{q}'_{\overline{t}'} \\ \hline\end{array}) \end{aligned}$$

Note that

$$Q := 1^{N_0} \leftarrow \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & q'_1 & ... & q'_{t'} \\ \hline\end{array} = 1^{N_0} \leftarrow \begin{array}{|c|c|c|c|c|c|}\hline 1 & ... & 1 & \overline{q}'_1 & ... & \overline{q}'_{\overline{t}'} \\ \hline\end{array}$$

is a single row, say of length M. Note that by Lemma 2.7.39 we have

$$|\mathrm{Row}_2(\emptyset \leftarrow 1^k v'_1...v'_{k'}1...1u'_1...u'_{\ell'})| = |\mathrm{Row}_2(\emptyset \leftarrow 1^{k-1}\overline{v}'_1...\overline{v}'_{\overline{k}'}1...1\overline{u}'_1...\overline{u}'_{\overline{\ell}'})|$$

Moreover, because $1^k v'_1...v'_{k'}1...1u'_1...u'_{\ell'}$ and $1^{k-1}\overline{v}'_1...\overline{v}'_{\overline{k}'}1...1\overline{u}'_1...\overline{u}'_{\overline{\ell}'}$ have the same number of letters, the sizes satisfy

$$|\emptyset \leftarrow 1^k v'_1...v'_{k'}1...1u'_1...u'_{\ell'}| = |\emptyset \leftarrow 1^{k-1}\overline{v}'_1...\overline{v}'_{\overline{k}'}1...1\overline{u}'_1...\overline{u}'_{\overline{\ell}'}|$$

It then follows that

$$|\mathrm{Row}_1(\emptyset \leftarrow 1^k v'_1...v'_{k'}1...1u'_1...u'_{\ell'})| = |\mathrm{Row}_1(\emptyset \leftarrow 1^{k-1}\overline{v}'_1...\overline{v}'_{\overline{k}'}1...1\overline{u}'_1...\overline{u}'_{\overline{\ell}'})|$$

Therefore, $\emptyset \leftarrow 1^k v'_1...v'_{k'}1...1u'_1...u'_{\ell'}$ and $\emptyset \leftarrow 1^{k-1}\overline{v}'_1...\overline{v}'_{\overline{k}'}1...1\overline{u}'_1...\overline{u}'_{\overline{\ell}'}$ share a common shape λ. By Corollary 2.6.3, T has at most factorization $T = G \cdot H$ where $G \in \mathrm{SSYT}(\lambda)$ and $H \in \mathrm{SSYT}((M))$. It follows that we in fact have the equality

$$\emptyset \leftarrow 1^k v'_1...v'_{k'}1...1u'_1...u'_{\ell'} = G = \emptyset \leftarrow 1...1^{k-1}\overline{v}'_1...\overline{v}'_{\overline{k}'}1...1\overline{u}'_1...\overline{u}'_{\overline{\ell}'}$$

We further note that

$$G = 1^k v'_1...v'_{k'} \cdot 1...1u'_1...u'_{\ell'}$$

is a factorization of G into two rows of length $k + \mathrm{DCE}_1(w)$ and $r - k - \mathrm{DCE}_1(w)$. Also,

$$G = 1...1^{k-1}\overline{v}'_1...\overline{v}'_{\overline{k}'} \cdot 1...1\overline{u}'_1...\overline{u}'_{\overline{\ell}'}$$

is a factorization of G into two rows of length $k - 1 + \mathrm{DCE}_1(w)$ and $r - k - \mathrm{DCE}_1(w) + 1$. Write $\tau(T_\infty^{-1}(w)) = W'_1 \otimes ... \otimes W'_x$. Note that $W'_1 = 1^k v'_1...v'_{k'}$ and $W'_2 = 1...1u'_1...u'_{\ell'}$. Write $\tau(T_\infty^{-1}(\mathrm{raise}_1(w))) = V_1 \otimes ... \otimes V_x$. Note that because the number of ones following $u'_{\ell'}$ and $\overline{u}'_{\overline{\ell}'}$ is the same, and because $1...1q'_1...q'_{t'} = 1...1\overline{q}'_1...\overline{q}'_{\overline{t}'}$, it follows that $V_i = W'_i$ for $i > 2$. We have $V_1 = 1^{k-1}\overline{v}'_1...\overline{v'}_{\overline{k}'}$ and $V_2 = 1...1\overline{u}'_1...\overline{u}'_{\overline{\ell}'}$ so that $G = W'_1 \leftarrow W'_2 = V_1 \leftarrow V_2$. We see that $\tau(\mathrm{raise}_1(T_\infty^{-1}(w))) = U_1 \otimes U_2 \otimes W'_3 \otimes ... \otimes W'_x$ where $G = U_1 \leftarrow U_2$ with $|U_1| = |W'_1| - 1 = k + \mathrm{DCE}_1(w) - 1 = |V_1|$ and $|U_2| = |W'_2| + 1 = r - k - \mathrm{DCE}_1(w) + 1 = |V_2|$. By Corollary 2.6.3, it follows that $U_1 = V_1$ and $U_2 = V_2$. Therefore, $\tau(\mathrm{raise}_1(T_\infty^{-1}(w))) = \tau(T_\infty^{-1}(\mathrm{raise}_1(w)))$ and so $\mathrm{raise}_1(T_\infty^{-1}(w)) = T_\infty^{-1}(\mathrm{raise}_1(w))$ as required. □

We remark that the refactorization of the first two ascent sequences of $T_\infty^{-1}(w)$ actually just amounts to moving a 1 over from the left side of v'_1 to the right side of $v'_{k'}$. This can be seen using Lemma 2.6.14 and the fact that $k' = \mathrm{DCE}_1(w)$ yet $\mathrm{DCE}_1(T_\infty^{-1}(w)) = k'$. Hence, the only unbumped letter must be a one. This can be seen in the computations of the proof, as the number of 1s to the left of the first non-trivial element of $v'_1...v'_{k'}$ is k but the number of ones to the left of the first non-trivial element of $\overline{v}'_1...\overline{v}'_{\overline{k}'}$ is $k-1$.

The fact that refactorization is easy to understand on $T_\infty^{-1}(w)$ allows us to prove a special case of Theorem 2.7.43, that we will need to prove Theorem 2.7.43. This special case is what we described as an instance where raise_1 and $X^\sharp$ are "obviously" the same thing in our game plan.

Lemma 2.7.41. Let $w \in B^n$ be highest weight with $\tau(w) = W_1 \otimes ... \otimes W_\ell$ and (ν, J) be the corresponding rigged configuration. Suppose either $\ell = 2$ or $\ell > 2$ with $\text{DCE}_1(w) \leq \text{DCE}_2(w)$ and the number of entries in W_1 greater than 1 is exactly equal to $\rho := \text{DCE}_1(w)$. Then $\Phi(\text{raise}_1(w)) = \Phi(w)^{\sharp(\rho)}$.

Proof. If $\ell = 2$, then W_1 is of the form $1^a 2^b$ and $W_2 = W_\ell = 1^M$ for some M satisfying $a+b+M = n$. In this case, the rigged configuration has a single rigged partition $(\nu, J)^{(1)}$ with single row of length b. Indeed, note that the rigged configuration corresponding to $1^{a-1}2^b 1^{M+1} = \text{raise}_1(1^a 2^b 1^M)$ is of the same form with the single rigging being one larger. Now asssume that $\ell \geq 3$.

Let $\text{DCE}_1(w) = \rho$. We let u be the reversed word of w. Write $u = u_1...u_x u_{x+1}...u_n...u_N$ where u_n is the last letter greater than 1 and $u_x \geq u_{x+1} \geq ... \geq u_n$ with $u_{x-1} < u_x$. Note that the first partition of $\Phi(u_1...u_n)$ contains a singular row of length ρ. In particular, it has the largest rigging among all rows of length ρ due to the fact that riggings are bounded above by their vacancy numbers. The rigged configuration of u has the same proper configuration and riggings as the rigged configuration of $u_1...u_n$. However, the coriggings of all rows in the first partition are increased by $N - n$. Note the proper configuration and riggings of $u_1...u_{x-1}1$ and $u_1...u_{x-1}$ are the same. Using Lemmas 2.4.24 and 2.4.25, we see that the rigged configurations X', Y' corresponding to $u_1...u_{x-1}1u_x...u_n$ and $u_1...u_x...u_n$ have the same proper configuration, satisfy $\tilde{X'}^{(i)} = \tilde{Y'}^{(i)}$ for $i > 1$, and $X'^{(1)} = Z^{(1)} \cup \{(\rho, q)\}, Y'^{(1)} = Z^{(1)} \cup \{(\rho, q')\}$ where q, q' are the vacancy numbers of a row of length ρ in $X'^{(1)}, Y'^{(1)}$ respectively and $Z = \Phi(u_1...u_{x-1})$. We see that $\nu^{(0)}(X')$ and $\nu^{(0)}(Y')$ differ by exactly one box. Hence, $q = q' + 1$. Moreover, because the proper configurations of X', Y' are the same and $\tilde{X'}^{(i)} = \tilde{Y'}^{(i)}$ for $i > 1$, we in fact have $X'^{(i)} = Y'^{(i)}$ for $i > 1$. The result follows by noting that the addition of trailing 1s do not affect the riggings. ☐

Theorem 2.7.15 implies time evolution intertwines with a configuration preserving map which changes riggings in a way completely determined from the configuration. This lets us prove the following Theorem.

Theorem 2.7.42. Let $w \in B^n$. Let w satisfy $\tau(w) = W_1 \otimes ... \otimes W_x$ with either $x = 2$ or $x \geq 3$ with $\text{DCE}_1(w) \leq \text{DCE}_2(w)$. Then, $X^{\sharp(\text{DCE}_1(w))} = \Phi(\text{raise}_1(w))$.

Proof. Note that because $\text{DCE}_1(w) \leq \text{DCE}_2(w)$, we have $\text{DCE}_1(T_\infty^{-1}(w)) = \text{DCE}_1(w)$ by Lemma 2.7.39. By Lemma 2.7.7, we see that writing $\tau(w) = W_1 \otimes ... \otimes W_x$, that $\text{DCE}_1(w) = \text{DCE}_1(T_\infty^{-1}(w))$ is equal to the number of entries of W_1 greater than 1. Then, using Theorem 2.7.15, the fact that time evolution is invertible, Lemma 2.7.41, and Lemma 2.7.40, the result follows. ☐

We can obtain a more powerful Theorem by performing raise on the leftmost index containing the minimum DCE. This results in Theorem 2.7.43, which is the main Theorem of this thesis. It follows from the rigged configuration argument part of the plan that we discussed earlier this section.

Theorem 2.7.43. Let $w \in B^n$ be highest weight such that $\tau(w) = W_1 \otimes ... \otimes W_x$. Let X be the rigged configuration corresponding to the reversed word of w. Then, $X^\sharp = \Phi(\text{raise}(w))$.

Proof. Suppose $\text{raise}(w) = \text{raise}_{i_0}(w)$. Proceed by induction on i_0. The base case is handled by Lemmas 2.7.40, 2.7.41 and the observation made above. Suppose $i_0 > 1$. Consider the word $w' = \text{RR}(W_2 \otimes ... \otimes W_x)$ with length ℓ. Let X' be the corresponding rigged configuration. Write $W_1 = v_1...v_k$. We want to show that $\Phi(v_1...v_k w')^\sharp = \Phi(\text{raise}_{i_0}(v_1...v_k w'))$. Note that $\text{raise}_{i_0}(v_1...v_k w') = v_1...v_k \text{raise}_{i_0-1}(w')$. So we wish to show that $\Phi(v_1...v_k w')^\sharp = \Phi(v_1...v_k \text{raise}_{i_0-1}(w'))$. Define $X'_j := \Phi(v_{k-j}...v_k \text{raise}_{i_0-1}(w'))$ and $Y_j := \Phi(v_{k-j}...v_k w')$ for $j > 0$ and $X'_0 = \Phi(\text{raise}_{i_0-1}(w')), Y_0 = \Phi(w')$. Define $\rho = \min_{1 \leq i < x} \text{DCE}_i(w)$, and $\alpha = \max(a : (\rho, a) \in X^{(1)})$.

First, note that because $\text{DCE}_1(w) > \rho$ we have by Theorem 2.4.21 and Lemma 2.4.9 that $\Phi(w')^{(1)}$ and $\Phi(w)^{(1)}$ have the same rows of length ρ (and their riggings agree).

Therefore $Y_0 = \Phi(w')$ has $Y_0^{(1)}$ containing (ρ, α). Another thing we note before we proceed is that the first DCE value of $v_1...v_k \text{raise}_{i_0-1}$ is at least ρ because $\text{DCE}_1(w) > \rho$.

We will show that each $X_j'^{(t)} = Y_j^{(t)}$ for $t > 1$ and that $X_j'^{(1)} = (Y_j^{(1)} \setminus \{(\rho, \alpha)\}) \cup \{(\rho, \alpha+1)\}$ for all $0 \leq j < k$. We also show that each $Y_j^{(1)}$ contains a copy of (ρ, α) and that the chain tables satisfy $\mathcal{M}(v_{k-j}...v_k w')_{t,\ell+j} = \mathcal{M}(v_{k-j}...v_k \text{raise}_{i_0-1}(w'))_{t,\ell+j}$ for all t. Clearly these properties holds for $j = 0$ (note that $Y^{(1)}$ contains (ρ, α) by Theorem 2.7.35.

For $0 < j < k$, suppose the property holds for $j-1$. Note that X'_{j-1}, Y'_{j-1} have the same configurations. So, their vacancy numbers are the same. Moreover, for $t > 1$, we have $X_{j-1}'^{(t)} = Y_{j-1}^{(t)}$. Therefore, $\widetilde{X'_{j-1}}^{(t)} = \widetilde{Y_{j-1}}^{(t)}$. By the definition of the KKR bijection, $\mathcal{M}(v_{k-j}...v_k \text{raise}_{i_0-1}(w'))_{t,\ell+j} = \mathcal{M}(v_{k-j}...v_k w')_{t,\ell+j}$.

Moreover, if $j \leq \rho$ then by the connectedness theorem,

$$\begin{aligned} \mathcal{M}(v_{k-j}...v_k \text{raise}_{i_0-1}(w'))_{1,\ell+j} &= \mathcal{M}(v_{k-j}...v_k w')_{1,\ell+j} \\ &= j - 1 \end{aligned}$$

If $j > \rho$, we note that $\mathcal{M}(v_{k-j}...v_k \text{raise}_{i_0-1}(w'))_{1,\ell+j}$ is the length of the largest singular row in $X_{j-1}'^{(1)}$ at most $\mathcal{M}(v_{k-j}...v_k \text{raise}_{i_0-1}(w'))_{2,\ell+j}$. Also, $\mathcal{M}(v_{k-j}...v_k w')_{1,\ell+j}$ is the length of the largest singular row in $Y_{j-1}^{(1)}$ which is at most $\mathcal{M}(v_{k-j}...v_k w')_{2,\ell+j}$. Recall that $\mathcal{M}(v_{k-j}...v_k w')_{2,\ell+j} = \mathcal{M}(v_{k-j}...v_k \text{raise}_{i_0-1}(w'))_{2,\ell+j}$. Moreover, both $X_{j-1}'^{(1)}$ and $Y_{j-1}^{(1)}$ have a singular string of length $h := \mathcal{M}(v_{k-j+1}...v_k w')_{1,\ell+j-1} + 1 = \mathcal{M}(v_{k-j+1}...v_k \text{raise}_{i_0-1}(w')) + 1 \geq \rho$. By induction, the coriggings on rows of length greater than $h \geq \rho + 1$ are equal. It follows that $g := \mathcal{M}(v_{k-j}...v_k w')_{1,\ell+j} = \mathcal{M}(v_{k-j}...v_k \text{raise}_{i_0-1}(w'))_{1,\ell+j}$. Let β be the vacancy number $P_g^{(1)}(X'_{j-1}) = P_g^{(1)}(Y_{j-1})$.

Then,

$$X_j'^{(1)} = (X_{j-1}'^{(1)} \setminus \{(g, \beta)\}) \cup \{(g+1, P_{g+1}^{(1)}(X'_j))\} \tag{2.11}$$

$$Y_j^{(1)} = (Y_{j-1}^{(1)} \setminus \{(g, \beta)\}) \cup \{(g+1, P_{g+1}^{(1)}(Y_j))\}. \tag{2.12}$$

Note that if $(g, \beta) = (\rho, \alpha)$, then by the chain lemma, $j - 1 = \rho$ Therefore, the multiplicity of (ρ, α) in $Y_{j-1}^{(1)}$ is at least two. Indeed, the word $v_{k-\rho+1}...v_k$ is connected and hence creates a new singular row of length ρ. The case where $(g, \beta) = (\rho, \alpha)$ is to say that $\beta = \alpha$ hence, that new singular row of length ρ also has rigging α. So there are at least two. Hence, $Y_j^{(1)}$ contains a copy of (ρ, α). We also note that

$$\begin{aligned}
X_j'^{(1)} &= (X_{j-1}'^{(1)} \setminus \{(g,\beta)\}) \cup \{(g,\beta+1)\} \\
&= (((Y_{j-1}^{(1)} \setminus \{(\rho,\alpha)\}) \cup \{(\rho,\alpha+1)\}) \setminus \{(g,\beta)\}) \cup \{(g,\beta+1)\} \\
&= (((Y_{j-1}^{(1)} \setminus \{(g,\beta)\}) \cup \{(g,\beta+1)\}) \setminus \{(\rho,\alpha)\}) \cup \{(\rho,\alpha+1)\} \\
&= (Y_j^{(1)} \setminus \{(\rho,\alpha)\}) \cup \{(\rho,\alpha+1)\}
\end{aligned}$$

Where the third line follows from the fact that by induction, $Y_{j-1}^{(1)}$ contains a copy of (ρ,α). We obtain every other $X_j'^{(t)} = Y_j^{(t)}$ for $t > 1$ using the equivalence of the chain table entries $\mathcal{M}(v_{k-j}...v_k w')_{t,\ell+r} = \mathcal{M}(v_{k-j}...v_k \mathrm{raise}_{i_0-1}(w'))_{t,\ell+r}$ for $0 \le r \le j$ and Lemma 2.4.24. This completes the inner induction. Hence, also the outer induction. The result follows. $\square$

Example 2.7.44. Consider the highest weight element $w = 223312211233221111 \in B^{18}$. The corresponding rigged configuration is

$$\begin{array}{ccc}
1^{18} &
\begin{array}{r|c|c|c|c|l}
\cline{2-5} 0 & & & & & 0 \\
\cline{2-5} 2 & & & & \multicolumn{2}{l}{1} \\
\cline{2-4} 6 & & & \multicolumn{3}{l}{3} \\
\cline{2-3} 6 & & & \multicolumn{3}{l}{2} \\
\cline{2-3}
\end{array} &
\begin{array}{r|c|c|l}
\cline{2-3} 0 & & & 0 \\
\cline{2-3} 0 & & & 0 \\
\cline{2-3}
\end{array} . \\
\nu^{(0)} & \nu^{(1)} & \nu^{(2)}
\end{array}$$

Let us compute raise(w). Note that

$$\tau(w) = \begin{array}{|c|c|c|c|}\hline 2&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|}\hline 1&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&1&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline\end{array}.$$

Hence, $\mathrm{DCE}_1(w) = 3, \mathrm{DCE}_2(w) = 2, \mathrm{DCE}_3(w) = 2, \mathrm{DCE}_4(w) = 2$. The left-most minimizing index of DCE is then 2 so that raise $=$ raise$_2$. The left-most unbumped element when inserting the third tensor factor into the second tensor factor is a 1. Therefore, we get raise$_2(w)$ by moving the 1 in the second tensor factor of $\tau(w)$ to the third:

$$\tau(\mathrm{raise}(w)) = \begin{array}{|c|c|c|c|}\hline 2&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&1&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline\end{array}.$$

In other words, raise$(w) = 223322111233221111$. The corresponding rigged configuration is

$$\begin{array}{ccc}
1^{18} &
\begin{array}{r|c|c|c|c|l}
\cline{2-5} 0 & & & & & 0 \\
\cline{2-5} 2 & & & & \multicolumn{2}{l}{1} \\
\cline{2-4} 6 & & & \multicolumn{3}{l}{4} \\
\cline{2-3} 6 & & & \multicolumn{3}{l}{2} \\
\cline{2-3}
\end{array} &
\begin{array}{r|c|c|l}
\cline{2-3} 0 & & & 0 \\
\cline{2-3} 0 & & & 0 \\
\cline{2-3}
\end{array} \\
\nu^{(0)} & \nu^{(1)} & \nu^{(2)}
\end{array}$$

which is $\Phi(w)^\sharp$ as expected.

Example 2.7.45. As stated in the introduction, there is a method to apply raise on a standard tableau directly. However, we note that this procedure is less useful both theoretically and practically. First we must describe what τ does to a standard tableau. What it does is colour the entries reading them in order. The same colour is used until the reading order moves south. For example,

$$T = \begin{array}{|c|c|c|c|}\hline 1 & 2 & 5 & 6 \\ \hline 3 & 4 & 8 & 9 \\ \hline 7 \\ \cline{1-1}\end{array} \mapsto \begin{array}{|c|c|c|c|}\hline 1 & 1 & 2 & 2 \\ \hline 2 & 2 & 3 & 3 \\ \hline 3 \\ \cline{1-1}\end{array}.$$

This is known as *destandardization* in [1]. Let k be the largest entry in this new semistandard tableau. We note that DCE_i is equal to the number of matched pairs between $(k-i)$ and $(k-i+1)$ when computing e_{k-i} in the sense of the signature rule (Algorithm 2.5.4). So, we see that $\mathrm{DCE}_1(T)$ is the number of 2s matched to 3s when computing $e_2(T)$ which is 3 in the above example. Also, $\mathrm{DCE}_2(T)$ is the number of 1s matched to 2s when computing $e_1(T)$ which is 2 in the above example. Hence, the minimizing DCE value is at 2. So, raise $=$ raise_2. On τ, we wish to compute ref_2 which corresponds to the raising operator $e_{3-2} = e_1$ (see Remark 2.6.7). We have

$$e_1\left(\begin{array}{|c|c|c|c|}\hline 1 & 1 & 2 & 2 \\ \hline 2 & 2 & 3 & 3 \\ \hline 3 \\ \cline{1-1}\end{array}\right) = \begin{array}{|c|c|c|c|}\hline 1 & 1 & 1 & 2 \\ \hline 2 & 2 & 3 & 3 \\ \hline 3 \\ \cline{1-1}\end{array}.$$

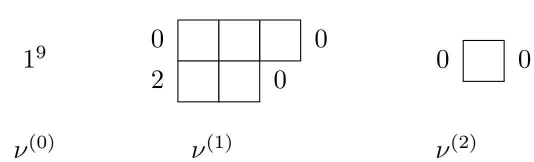
The semistandard tableau on the right corresponds to $\tau \circ \mathrm{raise}$. So we need to restandardize to obtain the image of raise,

$$\begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 6 \\ \hline 4 & 5 & 8 & 9 \\ \hline 7 \\ \cline{1-1}\end{array}.$$

The rigged configuration corresponding to

$$\begin{array}{|c|c|c|c|}\hline 1 & 2 & 5 & 6 \\ \hline 3 & 4 & 8 & 9 \\ \hline 7 \\ \cline{1-1}\end{array}$$

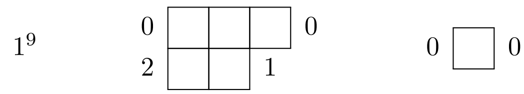
is



and the rigged configuration corresponding to

$$\begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 6 \\ \hline 4 & 5 & 8 & 9 \\ \hline 7 \\ \cline{1-1}\end{array}$$

is



$\nu^{(0)}$ $\qquad$ $\nu^{(1)}$ $\qquad$ $\nu^{(2)}$

which is what we expected.

2.8 Extensions of Raise

We now drop the assumption that words in B^n begin and end with a large number of ones. So far, we have only introduced methods to manipulate the riggings on the smallest row (and larger rows in the special case $\mathrm{DCE}_1(w) \leq \mathrm{DCE}_2(w)$). The perturbations $X^\sharp$ on rigged configurations seem hyper specific. Moreover, we only know how to compute $X^\sharp$ directly on a highest weight $w \in B^n$. In particular, we can only compute it directly on the tableau if that tableau is standard. The goal of this section is to explain how with a bit of work, the core results of the thesis can be extended to more powerful ones.

The main extension we wish to present, is a technique for manipulating the rigging on any row of the first rigged partition. This procedure (a specific instance of Algorithm 2.8.9) works by applying raise or raise^{-1} to carefully chosen subwords.

There are two other extensions we will briefly discuss. The first is a hide preserving standardization algorithm which is invertible modulo the action of the symmetric group. In particular, if the content of a semistandard tableau is recorded, then most of our results (including Algorithm 2.8.9) can be applied to semistandard tableaux by going through this standardization map.

The last extension we discuss in this section is a method to compute hide directly on the crystal element $w \in B^n$ (respectively, standard tableau). Note that this also yields a method to compute hide directly on semistandard tableau, by passing through the hide preserving standardization algorithm. There is already a technique to compute hide directly on highest weight elements of $B_{(\mu_1)} \otimes ... \otimes B_{(\mu_\ell)}$ (see [9]). The technique we present is arguably simpler when done on elements of B^n. Our technique also doesn't require assuming our element $w \in B^n$ lives in a large amount of vacuum as we did in many parts of Section 2.7.

We explain how to manipulate arbitrary riggings in 2.8.1. We then introduce both the hide algorithm, and the standardization algorithm in 2.8.2.

2.8.1 Manipulating Arbitrary Riggings of the First Rigged Partition

First, we must introduce a concept of a *spell*.

Definition 2.8.1. Let $w \in B^n$ with $\tau(w) = W_1 \otimes ... \otimes W_x$. We say a function f is a *spell* on w if $f(w) \in B^n$, and $\mathrm{hide}(\Phi(w)) = \mathrm{hide}(\Phi(f(w)))$.

Remark 2.8.2. The reason why we use an entire function f, instead of just another element w' with $\mathrm{hide}(w) = \mathrm{hide}(w')$ is because we want to convey that f is somehow changing the riggings of $\Phi(w)^{(1)}$ and nothing else.

Example 2.8.3. raise is always a spell on w if none of the smallest rows of $\Phi(w)^{(1)}$ are singular.

In fact, all of the spells we consider will be of the form of raise on some subword. As such, we need to identify when spells on a subword *induce* a spell on the entire word. This motivates the following definition.

Definition 2.8.4. Let $w \in B^n$ with $\tau(w) = W_1 \otimes ... \otimes W_x$. Let f be a spell on $\mathrm{RR}(W_k \otimes W_{k+1} \otimes ... \otimes W_x)$. If $\tilde{f} := 1^{\sum_{i=1}^{k-1}|W_k|} \otimes f$ is a spell on w, then $\tilde{f}$ is an induced spell. We say $\tilde{f}$ is induced from f.

The following two theorems give us a way to detect when a spell on a subword is indeed an induced spell.

Theorem 2.8.5. Let $w \in B^n$ be highest weight with $\tau(w) = W_1 \otimes ... \otimes W_x$. Suppose that $|W_1| = \mathrm{DCE}_1(w)$. Define $w' = \mathrm{RR}(W_2) \otimes ... \otimes \mathrm{RR}(W_x)$. Let f be a spell on w'. Then, $\mathrm{id}^{\otimes|W_1|} \otimes f$ is a spell on w if the length of the first ascent sequence of $\mathrm{RR}(W_1) \otimes f(w')$ is equal to $\mathrm{DCE}_1(\mathrm{RR}(W_1) \otimes f(w'))$. Moreover if $Y = \Phi(\mathrm{RR}(W_1) \otimes f(w'))$ then $Y^{(1)} = \Phi(f(w'))^{(1)} \cup \{(\rho, \alpha)\}$ where $\alpha = P_\rho^{(1)}(Y)$

Proof. Write $v = \mathrm{RR}(W_1) \otimes f(w')$ and $\tau(v) = V_1 \otimes ... \otimes V_x$. The hypothesis assumes that $\mathrm{DCE}_1(v) = |V_1| = |W_1|$. This implies that $f(w') = \mathrm{RR}(V_2) \otimes ... \otimes \mathrm{RR}(V_x)$. Write $v = v_1 ... v_n$. Note that Theorem 2.4.21 implies $v_\rho ... v_1$ is connected. Therefore, $\mathcal{M}(v_n...v_1)_{1,n-\rho+i} = i - 1$ for $1 \leq i \leq \rho$. We note $\mathcal{M}(v_n...v_1)_{j,n-\rho} = \mathcal{M}(w_n...w_1)_{j,n-\rho}$ for $j > 1$ because by the definition of a spell, the proper configuration and riggings of $\Phi(w_{\rho+1}...w_n), \Phi(v_{\rho+1}...v_n)$ are the same hence, the corigged partitions at index greater than one are equal. Then induction and Lemma 2.4.24 imply that $\Phi(v), \Phi(w)$ only differ by riggings on their first rigged partition. The fact that $Y^{(1)} = \Phi(f(w'))^{(1)} \cup \{(\rho, \alpha)\}$ follows from Theorem 2.4.21 and the definition of the KKR bijection (in particular, that rows that are lengthened are kept singular). □

Note that the vacancy numbers changing at various steps of the KKR bijection make it difficult to work on Yamanouchi words directly. In particular, vacancy numbers could be higher at at subword of a Yamanouchi word, allowing for spells that cannot extend to induced spells. Fortunately, there is another version of Theorem 2.8.5 which tells us when an spells on subwords are induced spells.

Theorem 2.8.6. Let $w \in B^n$ be highest weight with $\tau(w) = W_1 \otimes ... \otimes W_x$. Let $w' = \mathrm{RR}(W_2 \otimes ... \otimes W_x)$ and let $|W_1| = \mathrm{DCE}_1(w)$ with $\mathrm{DCE}_1(w) = \min_{1 \leq i < x} \mathrm{DCE}_i(w)$. If f is a spell on w', then $\tilde{f} = \mathrm{id}^{|W_1|} \otimes f$ is an induced spell on w if and only if $\mathrm{DCE}_1(\tilde{f}(w)) = \mathrm{DCE}_1(w)$, and if $\tilde{f}(w) = W_1' \otimes ... \otimes W_y'$ then $|W_1'| = |W_1| = \mathrm{DCE}_1(w)$.

Proof. By Theorem 2.8.5, it suffices to show that if either of the conditions $\mathrm{DCE}_1(\tilde{f}(w)) = \mathrm{DCE}_1(w)$ or $|W_1'| = |W_1|$ fail, then $\tilde{f}$ is not an induced spell. Note that we always have $|W_1'| \geq |W_1|$. If $|W_1'| > |W_1|$ then $\tilde{f}(w)$ has less ascent sequences and hence the size of the first column of the shape $\Phi(\tilde{f}(w))^{(1)}$ is smaller than that of $\Phi(w)^{(1)}$.

If $\mathrm{DCE}_1(\tilde{f}(w)) \neq \mathrm{DCE}_1(w)$ then the shapes of $\Phi(\tilde{f}(w))^{(1)}, \Phi(w)^{(1)}$ are again the same. Indeed, if $\mathrm{DCE}_1(\tilde{f}(w))$ is smaller, then because $\mathrm{DCE}_1(w) = \min_{1 \leq i < x} \mathrm{DCE}_i(w)$, we have by Theorem 2.7.35 that $\Phi(\tilde{f}(w))^{(1)}$ has a row of length $\mathrm{DCE}_1(\tilde{f}(w)) < \mathrm{DCE}_1(w)$. If $\mathrm{DCE}_1(\tilde{f}(w)) > \mathrm{DCE}_1(w)$ then it follows $|W_1'| > |W_1|$ which we showed implies that $\tilde{f}$ is not an induced spell. □

The above theorem tells us that if our spell increased any riggings by too much on a restricted word, then we will know by comparing DCE_1 with the length of the first ascent sequence..

Our goal is to be able to manipulate all the riggings of the first rigged partition. This will involve understanding raise^{-1} (for example, if we wish to decrease riggings). However, the importance of

raise^{-1} goes beyond that. The inverse of raise is also needed to gain access to the riggings attached to larger rows. The smallest rows are like the outer layers of an onion. To get to the larger rows, we have to peel them back first, change the inner layers and then put the outer layers back.

We know that because raise_{i_0} is invertible, it should in principle be possible to invert raise as long as we know which index i_0 had raise $= \text{raise}_{i_0}$. Additionally, it is clear that $\sharp$ can be inverted (it is injective). Suppose $\text{raise}(w) = u$, $w = w_1 ... w_n$ and $u = u_1 ... u_n$. We want to determine w. Suppose the left-most index minimizing DCE on w is at i_0 (say $\text{DCE}_{i_0}(w) = \rho$). We then know two things. The first is that $\text{DCE}_{i_0}(u) = \rho$ as well by definition of refactorization. The second is that the only DCE values that can possibly differ between u and w are $\text{DCE}_{i_0-1}(u) \leq \text{DCE}_{i_0-1}(w)$ and $\text{DCE}_{i_0+1}(u) \geq \text{DCE}_{i_0+1}(w)$. Therefore let j_0 be the left-most index minimizing DCE on u. We have that either $j_0 = i_0$ or $j_0 = i_0 + 1$. If we can distinguish these two cases, we can invert raise. The way we tell is simply that $\text{raise}_{i_0}^{-1}$ should not change the location of the left-most index minimizing DCE. This is because $\text{raise}(\text{raise}_{i_0}^{-1}) = \text{id}$.

In Section 2.6 we realized raise as some DSR_j and gave an interpretation of the j. We now do something similar for raise_i^{-1}. Suppose we want to compute $\text{raise}_i(w)^{-1}$. Read the first $\text{DCE}_i(w)$ letters of W_i from right to left. For each letter, find the largest letter smaller than it in W_{i+1}, and cross it out in W_{i+1}. The left-most uncrossed letter is moved over.

Example 2.8.7. Suppose we want to compute $\text{raise}_1^{-1}(223312223)$. Note that

$$\tau(223312223) = \boxed{2}\boxed{2}\boxed{3}\boxed{3} \otimes \boxed{1}\boxed{2}\boxed{2}\boxed{2}\boxed{3}$$

The DCE is 3. Performing the matching procedure described earlier we obtain

$$\boxed{2}\boxed{2}\boxed{3}\boxed{3} \otimes \boxed{\not 1}\boxed{2}\boxed{\not 2}\boxed{\not 2}\boxed{3}$$

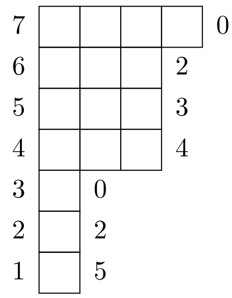
As the two 3s cross out the 2s, and the 2 cancels the 1. The left-most remaining number is 2. Therefore, $\text{raise}_1^{-1}(223312223)$ moves a 2 over from the second ascent sequence and we obtain

$$\text{raise}_1^{-1}(223312223) = 222331223$$

We are now able to give an algorithm to increase/decrease the rigging on any row, and demonstrate precisely when the rigging is invalid (either increased beyond the vacancy number, or decreased below zero). We describe the *Peel Spell Procedure* (PSP), and afterwards explain how it can be applied to the above problem.

First we must define a total ordering on the rows of a rigged partition. We will use this ordering to talk about which rigged row we wish to modify. We compare $(x, \alpha), (y, \beta) \in X^{(i)}$ pseudo-lexicographically. That is, if $x < y$ then $(x, \alpha) < (y, \beta)$. If $x = y$ then $(x, \alpha) < (y, \beta)$ if $\alpha > \beta$. In other words, if we draw the rows with smaller rigging higher, then the typical drawing of a rigged partition displays the order with the largest (rigged) row at the top, and the smallest (rigged) row at the bottom. Define the *rank* of a row in a rigged partition, to be where it occurs in this ordering (where the smallest is given rank 1).

Example 2.8.8. The rows of the rigged partition are drawn top to bottom in decreasing order where the rank is written in place of the vacancy numbers.



Here, $\mathrm{rank}(1,5) = 1$, $\mathrm{rank}(1,2) = 2$, $\mathrm{rank}(1,0) = 3$, $\mathrm{rank}(3,4) = 4$, $\mathrm{rank}(3,3) = 5$, $\mathrm{rank}(3,2) = 6$, and $\mathrm{rank}(4,0) = 7$.

The below algorithm provides us a way to perform a spell "only on rows above a certain rank", and stops at step (3) or (5) if and only if the change to the riggings on rows above a certain rank is incompatible with the vacancy numbers of the whole rigged partition.

Algorithm 2.8.9 (Peel Spell Procedure)**.** Let f be a spell on all highest weight words. Let $w \in B^n$ be highest weight and consider $X = \Phi(w)$. Let $k \le |X^{(1)}|$. Define $S := \{(x,\alpha) \in X^{(1)} : \mathrm{rank}(x,\alpha) \ge k\}$, and $S_T := X^{(1)} \setminus S$. Set $i = 1$, and $a_j = 0$ for all j.

(1) If $\min_{i \le j < x} \mathrm{DCE}_j(w) < |W_i|$, set $w \mapsto \mathrm{RR}(W_1) \otimes ... \otimes \mathrm{RR}(W_{i-1}) \otimes \mathrm{raise}(\mathrm{RR}(W_i) \otimes ... \otimes \mathrm{RR}(W_x))$ and increment $a_i \mapsto a_i + 1$. Otherwise, set $i = i + 1$.

(2) If $i < k$, go back to step (1). Otherwise, proceed to step (3).

(3) Set $w \mapsto \mathrm{RR}(W_1) \otimes ... \otimes \mathrm{RR}(W_{k-1}) \otimes f(\mathrm{RR}(W_k) \otimes ... \otimes \mathrm{RR}(W_x))$. If $\mathrm{DCE}_{k-1}(w) < |W_{k-1}|$ then stop. Otherwise, set $i = k - 1$ and proceed to step (4)

(4) Set $w \mapsto \mathrm{RR}(W_1) \otimes ... \otimes \mathrm{RR}(W_{i-1}) \otimes \mathrm{raise}^{-1^{(a_i)}}(\mathrm{RR}(W_i) \otimes ... \otimes \mathrm{RR}(W_x))$.

(5) If $\mathrm{DCE}_{i-1}(w) < |W_{i-1}|$ then stop. Otherwise, continue to step (6)

(6) If $i > 1$, set $i \mapsto i - 1$ and go back to step (4).

The resulting word w has the same hide as the original. Moreover, $X^{(1)} = f(X_S^{(1)}) \cup S_T$.

By Theorem 2.8.6, the condition to stop in step (4) checks whether or not $X_S^{(1)} \cup S_T$ has proper riggings. If we never stop at step (3) or (5), then $X_S^{(1)} \cup S_T$ has all riggings bounded below the corresponding vacancy numbers. We will later demonstrate this stopping condition (see Example 2.8.11).

If we take $f = \mathrm{raise}$ or $f = \mathrm{raise}^{-1}$ in the above algorithm, then we obtain a method to increase or decrease (respectively) the rigging on the row of rank k. Indeed, note that the row rank length k, is the smallest row with the largest rigging among all rows of rank at least k.

Example 2.8.10. Consider the highest weight element $w = 2222113311222111 \in B^{16}$. We wish to increase the rank 2 row in $\Phi(w)^{(1)}$ by one. First write

$$\tau(w) = \begin{array}{|c|c|c|c|}\hline 2 & 2 & 2 & 2 \\ \hline \end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1 & 1 & 3 & 3 \\ \hline \end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 2 & 2 & 2 \\ \hline \end{array} \otimes \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline \end{array}.$$

For simplicity of notation, we will work directly with $\tau(w)$, and identify raise_i with ref_i. We start at $i = 1$. First note that the DCE values reading left to right are $2, 2, 3$. We have the minimum DCE value $\text{DCE}_1(w) = \text{DCE}_2(w) = 2 < 4$. So we apply raise on w until they are equal. This takes 2 iterations of raise:

$$\tau(w) = \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&2&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&1&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|}\hline 1&1&1\\\hline\end{array}.$$

So we have $a_1 = 2$. Now we set $i = 2$ and can move on to step (3). The spell we want to apply is raise. The DCE values (ignoring index 1) are $4, 3$. So $\text{raise} = \text{raise}_3$ and we get

$$\tau(w) = \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&2&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline\end{array}.$$

Now we perform step (4). We have $i = 2 - 1 = 1$. So we perform raise^{-1} $a_1 = 2$ times. Because $i = 1$, we perform raise^{-1} on the entire word. First we see that the DCE values are $2, 3, 3$. So there is no ambiguity in this case, and $\text{raise}^{-1} = \text{raise}_1^{-1}$. One application yields

$$\tau(w) = \begin{array}{|c|c|c|}\hline 2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&1&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline\end{array}.$$

Now the DCE values are $2, 2, 3$. So either $\text{raise}^{-1} = \text{raise}_1^{-1}$ or raise_2^{-1}. We check that $\text{raise}^{-1} = \text{raise}_1^{-1}$ here and hence, applying $\text{raise}^{-1} = \text{raise}_1^{-1}$ we get

$$\tau(w) = \begin{array}{|c|c|c|c|}\hline 2&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&1&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline\end{array}.$$

We have performed raise^{-1} $a_1 = 2$ times, and also $i = 1$. So we are done here. Let us confirm our answer. The rigged configuration corresponding to the original word w is

$$\underset{\nu^{(0)}}{1^{15}} \qquad \underset{\nu^{(1)}}{\begin{array}{r|c|c|c|c|l}\cline{2-5} 0 & & & & & 0\\\cline{2-5} 2 & & & & \multicolumn{2}{|l}{0}\\\cline{2-4} 6 & & & & \multicolumn{2}{|l}{4}\\\cline{2-4}\end{array}} \qquad \underset{\nu^{(2)}}{\begin{array}{r|c|c|l}\cline{2-3} 2 & & & 0\\\cline{2-3}\end{array}} \quad .$$

The word we get after applying PSP is 2222113312221111 which has corresponding rigged configuration

$$\underset{\nu^{(0)}}{1^{15}} \qquad \underset{\nu^{(1)}}{\begin{array}{r|c|c|c|c|l}\cline{2-5} 0 & & & & & 0\\\cline{2-5} 2 & & & & \multicolumn{2}{|l}{1}\\\cline{2-4} 6 & & & & \multicolumn{2}{|l}{4}\\\cline{2-4}\end{array}} \qquad \underset{\nu^{(2)}}{\begin{array}{r|c|c|l}\cline{2-3} 2 & & & 0\\\cline{2-3}\end{array}}$$

as desired.

Example 2.8.11. Consider again the highest weight element $w = 2222113311222111 \in B^{16}$. In the above example we see that the vacancy number $P_1^{(4)}(\nu) = 0$. Hence, there should be an issue if we try to increase the rigging of the rank 3 row by 1. We can reuse some of our work, and pick up where $i = 2$,

$$\tau(w) = \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&2&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&1&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|}\hline 1&1&1\\\hline\end{array}.$$

So we have $a_1 = 2$. Now we set $i = 2$. The DCE values (ignoring index 1) are $4, 3$. So we perform raise on the subword corresponding to the $2, 3, 4$th tensor factor until the length of the second tensor factor is 3. After one iteration (note that here raise $=$ raise$_3$ because $\mathrm{DCE}_3(w) = 3 < \mathrm{DCE}_2(w) = 4$) we have

$$\tau(w) = \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&2&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline\end{array}.$$

We still have that the second tensor factor has length $6 > 3$. Now, the left-most minimizing index for DCE is at 2. So we apply raise $=$ raise$_2$,

$$\tau(w) = \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&1&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&2&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline\end{array}.$$

We continue to apply raise on the subword not containing the first tensor factor, until the second tensor factor has length 3. We obtain

$$\tau(w) = \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|}\hline 2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|c|}\hline 1&1&1&2&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline\end{array}.$$

We had to apply raise 4 times so that $a_2 = 4$. Now, we set $i = 3$, and apply our desired spell, raise, on the subword corresponding to the last two tensor factors. We get

$$\tau(w) = \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|}\hline 2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&2&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&1&1&1&1\\\hline\end{array}.$$

Now, we can proceed to step (4). We have $i = 3 - 1 = 2$. Perform raise^{-1} on the last three tensor factors $a_2 = 4$ times. After one application,

$$\tau(w) = \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&2&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&1&1&1&1\\\hline\end{array}.$$

After another application,

$$\tau(w) = \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&2&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&1&1&1&1\\\hline\end{array}.$$

And another...

$$\tau(w) = \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&2&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&1&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&1&1&1\\\hline\end{array}.$$

And the final one,

$$\tau(w) = \begin{array}{|c|c|}\hline 2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|}\hline 1&2&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&1&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|}\hline 1&1&1\\\hline\end{array}.$$

Proceeding to step (5), we see $\mathrm{DCE}_1 = 1$, but the first tensor factor has length 2. So we stop here, and note that our attempted change to the rank 3 row was illegal.

If one only wants to manipulate the smallest rows, there is a simpler procedure.

Definition 2.8.12. Let X be a rigged configuration such that the smallest row of $X^{(1)}$ is k. Suppose there is a non-singular row of length k in $X^{(1)}$. Let $\alpha = \max(y : (k, y) \in X^{(1)}, y < P_k^{(1)}(X))$. Define $X^\dagger$ such that ${X^\dagger}^{(i)} = X^{(i)}$ for $i > 1$, they have the same underlying configuration, and ${X^\dagger}^{(1)} = (X^{(1)} \setminus \{(k, \alpha)\}) \cup \{(k, \alpha + 1)\}$.

Definition 2.8.13. Let $w \in B^n$ be a highest weight word with $\tau(w) = W_1 \otimes ... \otimes W_x$. Let $k = \min_{1 \le i < x} \mathrm{DCE}_i(w)$. Consider the set of indices $S = \{j : \mathrm{DCE}_j(w) = k\}$. Suppose that $\mathrm{DCE}_1(w) = ... = \mathrm{DCE}_\ell(w) = k$. Consider the smallest index $i_0 \in S$ such that $i_0 > \ell$ and $\min_{1 \le i < x} \mathrm{DCE}_i(\mathrm{raise}_{i_0}(w)) = k$. Then $\overline{\mathrm{raise}}(w) := \mathrm{raise}_{i_0}(w)$.

The following Theorem tells us that $\overline{\mathrm{raise}}$ increases the rigging of the smallest row with largest *non-singular* rigging.

Theorem 2.8.14. Let $w \in B^n$ be highest weight with $\Phi(w) = X$. Then $\Phi(\overline{\mathrm{raise}}(w)) = X^\dagger$.

Proof. Let $\Phi(v) = X^\dagger$ with $v = v_1...v_n$. Let ℓ be the smallest index with $v_\ell \neq w_\ell$. Define $w' = w_\ell...w_n$ and $v' = v_\ell...v_n$. Consider $X' = \Phi(w')$ and $Y' = \Phi(v')$. Note that X', Y' are the same, up to a single row of length k in the first rigged partition ${X'}^{(1)}, {Y'}^{(1)}$ which differs by exactly one. That is, there is some α such that ${Y'}^{(1)} = ({X'}^{(1)} \setminus \{(k, \alpha)\}) \cup \{(k, \alpha+1)\}$. If $v_\ell \neq w_\ell$, then it must be because the first entry of the last column of the chain tables corresponding to Y' is $k-1$. In particular, the smallest singular row of ${Y'}^{(1)}$ is of length k, and there is no singular row of length k in ${X'}^{(1)}$. We moreover see by Proposition 2.4.23 that ${Y'}^{(1)}$ (and hence also ${X'}^{(1)}$) has no rows of length less than k. Among all rows of length k in ${X'}^{(1)}$, the one with the largest rigging has rigging α (because $(k, \alpha+1)$ is singular in ${Y'}^{(1)}$). It follows that $Y' = {X'}^{\sharp}$. Hence by Theorem 2.7.43, $v' = \mathrm{raise}(w')$. In fact, define $\beta \le \ell$ so that $w_\beta \le w_{\beta+1} \le ... \le w_\ell$ and $w_{\beta-1} > w_\beta$. Notice that if we define $X'' = \Phi(w_\beta...w_n)$ and $Y'' = \Phi(v_\beta...v_n)$ then (by the chain lemma) we see also that ${X''}^{\sharp} = {Y''}^{\sharp}$. This also tells us that $v = w_1...w_{\ell-1}\mathrm{raise}(w') = \mathrm{raise}_{i_0}(w)$ for some i_0. Let $S = \{j : \mathrm{DCE}_j(w) = k\}$. Because the minimum DCE is an invariant of the configuration by Theorem 2.7.35, we have $\min_{1 \le i < x} \mathrm{DCE}_i(\mathrm{raise}_{i_0}(w)) = \min_{1 \le i < x} \mathrm{DCE}_i(v) = k$. Let $\gamma = P_k^{(1)}(X) = P_k^{(1)}(Y)$. We note that by Proposition 2.4.23, the row of length k that the ascent sequences in $v_1...v_\beta, w_1...w_\beta$ construct all have rigging $\gamma > \alpha$. It follows i_0 is the smallest such index and hence $v = \mathrm{raise}_{i_0}(w) = \overline{\mathrm{raise}}(w)$. $\square$

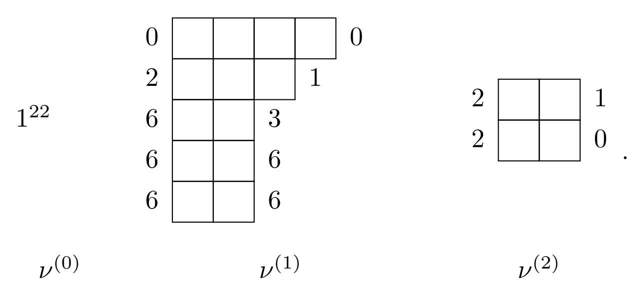
Example 2.8.15. Consider the element $w = 2312312222111332221111 \in B^{22}$. Writing $\tau(w)$,

$$\boxed{2\,|\,3} \otimes \boxed{1\,|\,2\,|\,3} \otimes \boxed{1\,|\,2\,|\,2\,|\,2\,|\,2} \otimes \boxed{1\,|\,1\,|\,1\,|\,3\,|\,3} \otimes \boxed{2\,|\,2\,|\,2} \otimes \boxed{1\,|\,1\,|\,1\,|\,1}.$$

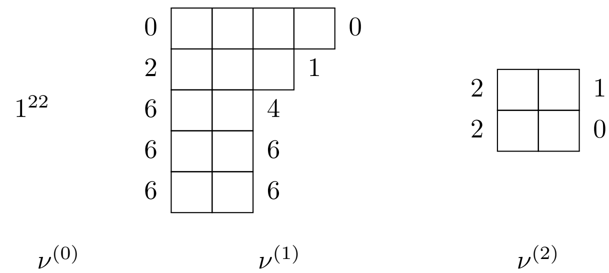
The DCE values are (reading left to right), $2, 2, 3, 2, 3$. Because $\mathrm{DCE}_1 = \mathrm{DCE}_2 = 2$, we are looking for the left-most minimizing DCE value i_0, where we insist that $i_0 > 2$, and that performing raise_{i_0} does not affect $\mathrm{DCE}_1, \mathrm{DCE}_2$. We see that this is acheived at $i_0 = 4$. So performing raise_4 (respectively, ref_4 on $\tau(w)$) we get

$$\boxed{2\,|\,3} \otimes \boxed{1\,|\,2\,|\,3} \otimes \boxed{1\,|\,2\,|\,2\,|\,2\,|\,2} \otimes \boxed{1\,|\,1\,|\,3\,|\,3} \otimes \boxed{1\,|\,2\,|\,2\,|\,2} \otimes \boxed{1\,|\,1\,|\,1\,|\,1}.$$

So, $\overline{\mathrm{raise}}(w) = 2312312222113312221111$. The rigged configuration corresponding to w is



The rigged configuration corresponding to $\overline{\text{raise}}(w)$ is



as expected.

2.8.2 Hide and Standardization

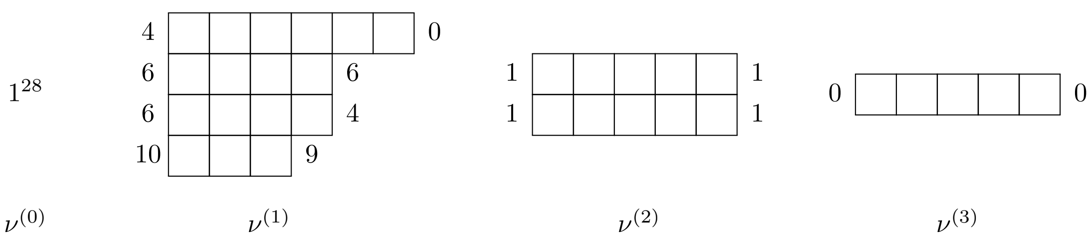
In [15], Sakamoto presented a crystal algorithm to compute $\text{hide}(\nu, J)$. We introduce an arguably simpler algorithm.

Let $w \in B^n$ with $\tau(w) = W_1 \otimes ... \otimes W_x$. Perform raise until

$$|W_1| = \text{DCE}_1(w) = \min_{1 \le i < x} \text{DCE}_i(w).$$

Let $V_1 = W_1$. Set $w = \text{RR}(W_2 \otimes ... \otimes W_x)$ and repeat until we have defined $V_1, ..., V_x$. Let V_i' be V_i with all letters reduced by one. The crystal element $V_1' \otimes ... \otimes V_x'$ corresponds to $\text{hide}(\Phi(w))$ under the KKR map.

Example 2.8.16. Consider $w = 4444133411123331222222111111 \in B^{28}$. The corresponding rigged configuration is



In this example, we will always write τ of the word for simplicity of computations. We have

$$\boxed{4\,|\,4\,|\,4\,|\,4} \otimes \boxed{1\,|\,3\,|\,3\,|\,4} \otimes \boxed{1\,|\,1\,|\,1\,|\,2\,|\,3\,|\,3\,|\,3}$$
$$\otimes \boxed{1\,|\,2\,|\,2\,|\,2\,|\,2\,|\,2\,|\,2} \otimes \boxed{1\,|\,1\,|\,1\,|\,1\,|\,1\,|\,1}$$

The DCE values left to right are $3, 3, 4, 6$. The minimum value is 3, and the left most occurence of it occurs at index 1. Therefore we apply raise_1 and obtain

$$\boxed{4\,|\,4\,|\,4} \otimes \boxed{1\,|\,3\,|\,3\,|\,4\,|\,4} \otimes \boxed{1\,|\,1\,|\,1\,|\,2\,|\,3\,|\,3\,|\,3}$$
$$\otimes \boxed{1\,|\,2\,|\,2\,|\,2\,|\,2\,|\,2\,|\,2} \otimes \boxed{1\,|\,1\,|\,1\,|\,1\,|\,1\,|\,1}$$

Now, the first ascent sequence's essential part is of length 3, which is also equal to its DCE value. Hence, we set $V_1 = \begin{array}{|c|c|c|}\hline 4&4&4\\\hline\end{array}$ and restrict ourselves to the other tensor factors,

$$\begin{array}{|c|c|c|c|c|}\hline 1&3&3&4&4\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|c|}\hline 1&1&1&2&3&3&3\\\hline\end{array}$$
$$\otimes \begin{array}{|c|c|c|c|c|c|c|}\hline 1&2&2&2&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&1&1&1&1\\\hline\end{array}$$

The DCE values left to right are $4, 4, 6$. The minimum value is 4, and the left most occurence is at index 1. So we again apply raise_1 to obtain

$$\begin{array}{|c|c|c|c|}\hline 3&3&4&4\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|c|c|}\hline 1&1&1&1&2&3&3&3\\\hline\end{array}$$
$$\otimes \begin{array}{|c|c|c|c|c|c|c|}\hline 1&2&2&2&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&1&1&1&1\\\hline\end{array}$$

Now, DCE_1 is equal to the length of its essential part, so we set $V_2 = \begin{array}{|c|c|c|c|}\hline 3&3&4&4\\\hline\end{array}$ and now only consider the later tensor factors:

$$\begin{array}{|c|c|c|c|c|c|c|c|}\hline 1&1&1&1&2&3&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|c|}\hline 1&2&2&2&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&1&1&1&1\\\hline\end{array}$$

The DCE values are $4, 6$. Applying raise_1 4 times we obtain

$$\begin{array}{|c|c|c|c|}\hline 2&3&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|}\hline 1&1&1&1&1&2&2&2&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&1&1&1&1\\\hline\end{array}$$

The length of the first tensor factor is equal to the first DCE value, so $V_3 = \begin{array}{|c|c|c|c|}\hline 2&3&3&3\\\hline\end{array}$. Concentrating on the last two factos,

$$\begin{array}{|c|c|c|c|c|c|c|c|c|c|c|}\hline 1&1&1&1&1&2&2&2&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&1&1&1&1\\\hline\end{array}.$$

Performing raise until the first factor becomes singular, we get

$$\begin{array}{|c|c|c|c|c|c|}\hline 2&2&2&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|}\hline 1&1&1&1&1&1&1&1&1&1&1\\\hline\end{array}.$$

So, $V_4 = \begin{array}{|c|c|c|c|c|c|}\hline 2&2&2&2&2&2\\\hline\end{array}$. Therefore,

$$\begin{aligned} V_1' &= \begin{array}{|c|c|c|}\hline 3&3&3\\\hline\end{array} \\ V_2' &= \begin{array}{|c|c|c|c|}\hline 2&2&3&3\\\hline\end{array} \\ V_3' &= \begin{array}{|c|c|c|c|}\hline 1&2&2&2\\\hline\end{array} \\ V_4' &= \begin{array}{|c|c|c|c|c|c|}\hline 1&1&1&1&1&1\\\hline\end{array}. \end{aligned}$$

So the crystal element which corresponds to

$$\text{hide}(\Phi(4444133411123331222222111111))$$

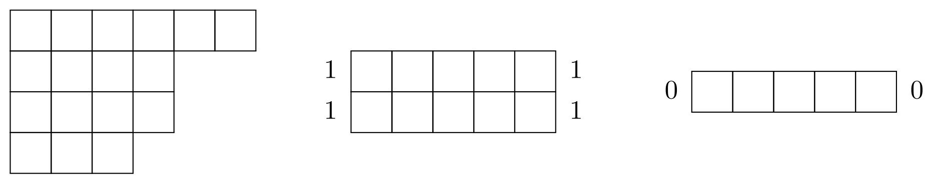
is

$$\begin{array}{|c|c|c|}\hline 3&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 2&2&3&3\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|}\hline 1&2&2&2\\\hline\end{array} \otimes \begin{array}{|c|c|c|c|c|c|}\hline 1&1&1&1&1&1\\\hline\end{array}$$

The corresponding semistandard tableau is

1	1	1	1	1	1	2
2	2	2	3	3		
3	3	4	4	4		

The corresponding rigged configuration is



$\nu^{(0)}$ $\nu^{(1)}$ $\nu^{(2)}$

Which is indeed the hide of the desired rigged configuration.

One of the main applications of hide is to determine the entire configuration. To do so, we need to know how to perform hide on a semistandard tableau. This is because it is very unlikely that performing hide on a standard tableau will again yield a standard tableau. Indeed see that in Example 2.8.16 we end up with a path which corresponds to a semistandard tableau, rather than a standard tableau.

To extend our algorithm for computing hide directly on highest weight elements $w \in B^n$, we introduce a hide preserving standardization map. In particular, the map also preserves the riggings on the first rigged partition.

Theorem 2.8.17. Let $w = W_1 \otimes ... \otimes W_k \in B_{(r_1)} \otimes ... \otimes B_{(r_k)}$ be highest weight with $X = \Phi(w)$. Define a procedure as follows: if $r_1 > 1$, u be the smallest element of W_1 and let W_1' be W_1 with u removed. Map $w \mapsto \boxed{u} \otimes W_1' \otimes ... \otimes W_k$. Via combinatorial R, we have

$$\boxed{u} \otimes W_1' \otimes ... \otimes W_k \simeq V_1 \otimes ... \otimes V_k \otimes \boxed{u'} =: w' \in B_{(r_1-1)} \otimes B_{(r_2)} \otimes ... \otimes B_{(r_k)} \otimes B$$

If $r_1 = 1$, then we immediately apply combinatorial R,

$$W_1 \otimes ... \otimes W_k \simeq V_1 \otimes ... \otimes V_k =: w' \in B_{(r_2)} \otimes ... \otimes B_{(r_k)} \otimes B_{(r_1)}$$

Set $w = w'$ and repeat until the word lives in $B^{\sum_{i=1}^k r_i}$. Let $Y = \Phi(w)$ be the rigged configuration corresponding to the path $w \in B^{\sum_{i=1}^k r_i}$ obtained after applying the algorithm. Then, $\text{hide}(X) = \text{hide}(Y)$ and $X^{(1)} = Y^{(1)}$.

Proof. Follows from [16, Definition 4.5] (and the fact that the definition agrees with the standard one) as well as [8, Lemma 8.5]. □

The tableau interpetation is the following. Take a semistandard tableau T with largest entry n. If there is more than one entry with value n, take the right most and change it to $n+1$. Then apply the crystal reflection operators $s_1 \circ ... \circ s_n$. Repeat until the tableau is standard.

So to extend the above algorithm for hide, we simply apply this procedure first.

We also note that the procedure described in Theorem 2.8.17 is invertible modulo combinatorial R on paths (respectively the action of S_n on tableaux). Hence, we can also apply PSP to semistandard tableaux by conjugating by this standardization procedure.

Bibliography

[1] Sami Assaf and Dominic Searles. Kohnert polynomials. *Experimental Mathematics*, 31(1):93–119, 2022.

[2] Daniel Bump and Anne Schilling. *Crystal bases: Representations and Combinatorics*. World Scientific Publishing Company, 2017.

[3] Jacques Désarménien, Bernard Leclerc, and J-Y Thibon. Hall-littlewood functions and kostka-foulkes polynomials in representation theory. *Séminaire Lotharingien de Combinatoire [electronic only]*, 32:38–p, 1994.

[4] Goro Hatayama, Kazuhiro Hikami, Rei Inoue, Atsuo Kuniba, Taichiro Takagi, and Tetsuji Tokihiro. The $a_m^{(1)}$ automata related to crystals of symmetric tensors. *Journal of Mathematical Physics*, 42(1):274–308, 2001.

[5] Masaki Kashiwara. On crystal bases of the q-analogue of universal enveloping algebras. 1991.

[6] S. V. Kerov, A. N. Kirillov, and N. Yu. Reshetikhin. Combinatorics, bethe ansatz, and representations of the symmetric group. *Journal of Soviet Mathematics*, 41:916–924, 1988.

[7] A. N. Kirillov and N. Yu. Reshetikhin. The bethe ansatz and the combinatorics of young tableaux. *Journal of Soviet Mathematics*, 41:925–955, 1988.

[8] Anatol N. Kirillov, Anne Schilling, and Mark Shimozono. A bijection between littlewood-richardson tableaux and rigged configurations. *Selecta Mathematica*, 8(1):67, 2002.

[9] Atsuo Kuniba, Masato Okado, Reiho Sakamoto, Taichiro Takagi, and Yasuhiko Yamada. Crystal interpretation of kerov–kirillov–reshetikhin bijection. *Nuclear Physics B*, 740(3):299–327, April 2006. URL: `http://dx.doi.org/10.1016/j.nuclphysb.2006.02.005`, `doi:10.1016/j.nuclphysb.2006.02.005`.

[10] Thomas Lam, Pavlo Pylyavskyy, and Reiho Sakamoto. Rigged configurations and cylindric loop schur functions. *Annales de l'Institut Henri Poincaré*, 5(4):513–555, 2018.

[11] Holly Mandel and Oliver Pechenik. Orbits of plane partitions of exceptional lie type. *European Journal of Combinatorics*, 74:90–109, 2018.

[12] V. Reiner, D. Stanton, and D. White. The cyclic sieving phenomenon. *Journal of Combinatorial Theory, Series A*, 108(1):17–50, 2004. `doi:10.1016/j.jcta.2004.04.009`.

[13] Brendon Rhoades. Cyclic sieving, promotion, and representation theory. *Journal of Combinatorial Theory, Series A*, 117(1):38–76, 2010.

[14] Brendon Rhoades. A skein action of the symmetric group on noncrossing partitions. *Journal of Algebraic Combinatorics*, 45:81–127, 2017.

[15] Reiho Sakamoto. Crystal interpretation of kerov–kirillov–reshetikhin bijection ii. proof for case. *Journal of Algebraic Combinatorics*, 27(1):55–98, 2008.

[16] Anne Schilling. X=m theorem: Fermionic formulas and rigged configurations under review, 2005. URL: `https://arxiv.org/abs/math/0512161`, `arXiv:math/0512161`.

[17] Travis Scrimshaw. Rigged configurations as tropicalizations of loop schur functions. *Journal of Integrable Systems*, 2(1):xyw015, 2017. `doi:10.1093/integr/xyw015`.

[18] Mark Shimozono. Crystals for dummies. 2005. URL: `https://www.aimath.org/WWN/kostka/crysdumb.pdf`.

[19] Richard P. Stanley. *Enumerative Combinatorics Vol. 1*. Cambridge University Press, 2011.

[20] Richard P. Stanley. *Enumerative Combinatorics Vol. 2*. Cambridge University Press, 2023.

[21] Daisuke Takahashi. On some soliton systems defined by using boxes and balls. In *Proceedings of the international symposium on nonlinear theory and its applications (NOLTA'93)*, pages 555–558, 1993.

[22] Daisuke Takahashi and Junkichi Satsuma. A soliton cellular automaton. *Journal of the Physical Society of Japan*, 59(10):3514–3519, 1990.