ON COMBINATORIAL MODELS FOR AFFINE CRYSTALS

CARLY BRIGGS, CRISTIAN LENART, AND ADAM SCHULTZE

ABSTRACT. The tableau model for Kirillov-Reshetikhin (KR) crystals, which are finite dimensional crystals corresponding to certain affine Lie algebras, is commonly used for its ease of crystal operator calculations. However, its simplicity makes quite complex the calculation of statistics such as: keys (used to express Demazure characters), the crystal energy function (an affine grading on tensor products of KR crystals), and the combinatorial $R$-matrix (an affine crystal isomorphism permuting factors in a tensor product of KR crystals). It has been shown that these calculations are much simpler with the added structure in the quantum alcove model for KR crystals. In this paper, we give an explicit description of the crystal isomorphism between the mentioned realizations of KR crystals in all classical Lie types.

1. Introduction

Kashiwara’s crystals encode the structure of certain bases, called crystal bases [9], for highest weight representations of quantum groups $U_q(g)$ as $q$ goes to zero [7]. In this paper, we will focus on crystals corresponding to representations of both simple and affine Lie algebras [3]. As combinatorial objects, they can be visualized as directed graphs, whose edges are given by Kashiwara operators which are derived from the Chevalley generators of the corresponding quantum group [2]. The resulting combinatorics provides much insight into the original representations. For instance, for irreducible representations of simple Lie algebras, by developing a tensor product rule for the Kashiwara operators, one can visualize the decomposition of the tensor product of such representations into irreducible components [7].

Given a dominant weight $\lambda$ for a simple Lie algebra, there is a corresponding connected crystal graph, $B(\lambda)$, whose vertices can be realized as semistandard Young tableaux (certain fillings of Young diagrams [5]). We can also consider crystals for affine Lie algebras. Here, we will focus on Kirillov-Reshetikhin (KR) crystals, which are finite affine connected crystals, but can be decomposed into disjoint unions of classical crystals upon the removal of the affine edges [12]. These crystals are indexed by $r \times s$ rectangles and are denoted $B_r^s$. There are type specific models for KR crystals, like that given by Kashiwara-Nakashima (KN) tableaux [4], as well as type independent models such as the quantum alcove model, which works uniformly in all untwisted affine types [19, 22, 23]. We provide an explicit crystal isomorphism between the two models of the above mentioned KR crystals: the tableau model and the quantum alcove model.

Lenart and Postnikov defined the so-called alcove model for highest weight crystals associated to a semisimple Lie algebra $g$. In fact, the model was defined more generally, for symmetrizable Kac-Moody algebras $g$ [24, 25]. The alcove model is a discrete counterpart of the Littelmann path model.

Lenart and Lubovsky then generalized the alcove model to one for Kirillov-Reshetikhin (KR) crystals of affine Lie algebras; this is known as the quantum alcove model [19], as it is based on the quantum Bruhat graph. This graph first appeared in connection with the quantum cohomology of flag varieties of the corresponding finite Weyl group [6]. The path enumeration is determined by the choice of a certain sequence of alcoves, called an alcove path, like in the classical alcove model. If we restrict to paths in the usual Bruhat graph, we recover the classical alcove model. The
mentioned paths in the quantum Bruhat graph first appeared in [18], where they index the terms in the specialization to $t = 0$ of the Ram-Yip formula [29] for Macdonald polynomials $P_{\lambda}(X; q, t)$. Further, [19] defined crystal operators for the quantum alcove model, both the classical ones $f_i$, $i > 0$, and the affine operator $f_0$. It was shown in [22] that the quantum alcove model uniformly describes tensor products of column shape KR crystals for all untwisted affine types. An explicit crystal isomorphism was given in [19] for types $A$ and $C$ between the objects of the quantum alcove model and tensor products of Kashiwara-Nakashima (KN) columns [11], using the bijections constructed in [18].

In recent years, many applications have resulted from the quantum alcove model. In [22] it was shown that the so-called height statistic in the Ram-Yip formula mentioned above expresses the energy function on a tensor product of KR crystals, which endows it with an affine grading. The translation of this energy function to the tableau model in type $C$ was then given in [26]. On the other hand, extending Schützenberger’s *jeu de taquin* on Young tableaux to the quantum alcove model (which is based on so-called quantum Yang-Baxter moves) results in a realization of the combinatorial $R$-matrix [20]. Further, the quantum alcove model was used to determine *keys*, also known as initial direction in the (quantum) LS path model. These detect Demazure crystals inside highest weight ones (in the alcove model), and Demazure-type crystals inside tensor products of KR crystals (in the quantum alcove model), see [23]. The computation of these keys is easy in the (quantum) alcove model and the bijection given in this paper then transfers the key computation from the (quantum) alcove model to the tableau model, where algorithms are only known for types $A$ and $C$ [30, 31, 33].

While the tableau model is simpler, it has less easily accessible information, so it is generally hard to use in specific computations (like those of the energy function, the combinatorial $R$-matrix and keys, as described above). As these computations are much simpler in the quantum alcove model, an alternative is to relate them to the tableau model, via an explicit affine crystal isomorphism. As was mentioned above, this has been done in types $A$ and $C$. Here we extend this work to types $B$ and $D$, where the corresponding bijection is much more involved, and has important additional features.

**Acknowledgements.** C.B. was partially supported by the NSF grant DMS-1101264. C.L. was partially supported by the NSF grants DMS-1362627 and DMS-1855592. A.S. was partially supported by the NSF grant DMS-1362627 and the Chateaubriand Fellowship from the Embassy of France in the United States.

2. **Background**

2.1. **Root Systems.** [8] Let $\mathfrak{g}$ be a complex semisimple Lie algebra and $\mathfrak{h}$ a Cartan subalgebra, whose rank is $n$. Let $\Phi \subset \mathfrak{h}^*$ be the corresponding irreducible root system, $\mathfrak{h}_\mathbb{R}^*$ the real span of the roots, and $\Phi^+ \subset \Phi$ the set of positive roots.

Let $\rho := \frac{1}{2}(\sum_{\alpha \in \Phi^+} \alpha)$. Let $\alpha_1, \ldots, \alpha_n \in \Phi^+$ be the corresponding simple roots. We denote $\langle \cdot, \cdot \rangle$ the nondegenerate scalar product on $\mathfrak{h}_\mathbb{R}^*$ induced by the Killing form. Given a root $\alpha$, we consider the corresponding coroot $\alpha^\vee := 2\alpha/\langle \alpha, \alpha \rangle$ and reflection $s_\alpha$.

Let $W$ be the corresponding Weyl group, whose Coxeter generators are denoted, as usual, by $s_i := s_{\alpha_i}$. The length function on $W$ is denoted by $l(\cdot)$, the Bruhat order on $W$ is defined by its covers $w \lessdot ws_\alpha$, for $\alpha \in \Phi^+$, if $l(ws_\alpha) = l(w) + 1$. The mentioned covers correspond to the labeled directed edges of the Bruhat graph on $W$:

$$w \overset{\alpha}{\rightarrow} ws_\alpha \text{ for } w \lessdot ws_\alpha.$$ 

The weight lattice $\Lambda$ is given by

$$\Lambda = \{ \lambda \in \mathfrak{h}_\mathbb{R}^* : \langle \lambda, \alpha \rangle \in \mathbb{Z} \text{ for any } \alpha \in \Phi^+ \}.$$
Figure 1. The quantum Bruhat graph for the Weyl group of type $A_3$

The weight lattice $\Lambda$ is generated by the fundamental weights $\omega_1, \ldots, \omega_n$, which form the dual basis to the basis of simple coroots, i.e., $\langle \omega_i, \alpha^\vee \rangle = \delta_{i,j}$. The set $\Lambda^+$ of dominant weights is given by

$$\Lambda^+ := \{ \lambda \in \Lambda : \langle \lambda, \alpha^\vee \rangle \geq 0 \text{ for any } \alpha \in \Phi^+ \}.$$ 

Given $\alpha \in \Phi$ and $k \in \mathbb{Z}$, we denote by $s_{\alpha,k}$ the reflection in the affine hyperplane

$$H_{\alpha,k} := \{ \lambda \in \mathfrak{h}_R^* : \langle \lambda, \alpha^\vee \rangle = k \}.$$ 

These reflections generate the affine Weyl Group $W_{aff}$ for the dual root system $\Phi^\vee := \{ \alpha^\vee | \alpha \in \Phi \}$. The hyperplanes $H_{\alpha,k}$ divide the real vector space $\mathfrak{h}_R^*$ into open regions, called alcoves. The fundamental alcove $A_\circ$ is given by

$$A_\circ := \{ \lambda \in \mathfrak{h}_R^* | 0 < \langle \lambda, \alpha^\vee \rangle < 1 \text{ for all } \alpha \in \Phi^+ \}.$$ 

Define $w \triangleleft ws_\alpha$, for $\alpha \in \Phi^+$, if $l(ws_\alpha) = l(w) - 2\langle \rho, \alpha^\vee \rangle + 1$. These are the upward facing arrows in Figure 1. The quantum Bruhat graph [6] is defined by adding to the Bruhat graph the following edges (the downward facing arrows in Figure 1) labeled by positive roots $\alpha$:

$$w \xrightarrow{\alpha} ws_\alpha \text{ for } w \triangleleft ws_\alpha.$$ 

2.2. Kirillov-Reshetikhin (KR) crystals. Given a simple or affine Lie algebra $\mathfrak{g}$, we define the corresponding Kashiwara Crystal to be the basis resulting from taking the limit of the quantum group $U_q(\mathfrak{g})$ as $q$ goes to zero. Such bases can be given the structure of a colored oriented graph, which we call a Crystal Graph, whose arrows are defined by the Kashiwara operators which are related to the Chevalley generators [7]. We will now detail the resulting combinatorial structure of such crystal graphs. Our definitions closely follow those given in [2].

Definition 2.1.

Let $\Phi$ be the root system and $\Lambda$ the weight lattice, with simple roots $\alpha_i$ for $i$ in an indexing set $I$, associated to a Lie algebra $\mathfrak{g}$. The crystal of type $\Phi$ is then a nonempty set $B$ along with the following maps using $i \in I$ and an auxiliary element $0 \notin B$:
(1) the Kashiwara Operators $e_i, f_i : B \to B \sqcup 0$, with the conditions that if $x, y \in B$ then $e_i(x) = y$ if and only if $f_i(y) = x$.

(2) the $e_i$-string and $f_i$-string operators $e_i, \varphi_i : B \to \mathbb{Z} \sqcup \{-\infty\}$, with the condition that if $x, y \in B$ are such that $e_i(x) = y$, then $e_i(y) = \varepsilon(x) - 1$ and $\varphi_i(y) = \varphi_i(x) + 1$. In the case that $\varphi_i(x) = -\infty$ or $\varepsilon_i(x) = -\infty$, we require that $e_i(x) = f_i(x) = 0$.

(3) the weight map $\text{wt} : B \to \Lambda$, where if $x, y \in B$ are such that $e_i(x) = y$, then $\text{wt}(y) = \text{wt}(x) + \alpha_i$. Further, for each $x \in B$ and $i \in I$, we have that $\langle \text{wt}(x), \alpha_i \rangle = \varphi_i(x) - \varepsilon_i(x)$.

The associated crystal graph is then built with elements of $B$ as vertices and, for $x, y \in B$, we draw an edge $y \to x$ exactly when $f_i(y) = x$.

Given two $\mathfrak{g}$-crystals $B_1$ and $B_2$, we define their tensor product $B_1 \otimes B_2$ as follows. As a set, $B_1 \otimes B_2$ is the Cartesian product of the two sets. For $b = b_1 \otimes b_2 \in B_1 \otimes B_2$, the weight function is simply $\text{wt}(b) = \text{wt}(b_1) + \text{wt}(b_2)$. The crystal operator $f_i$ is given by

$$f_i(b_1 \otimes b_2) = \begin{cases} f_i(b_1) \otimes b_2 & \text{if } \varepsilon_i(b_1) \geq \varphi_i(b_2), \\ b_1 \otimes f_i(b_2) & \text{otherwise}, \end{cases}$$

while $e_i(b)$ is defined similarly.

The highest weight crystal $B(\lambda)$ of highest weight $\lambda \in \Lambda^+$ is a certain crystal with a unique element $\mu_\lambda$ such that $e_i(\mu_\lambda) = 0$ for all $i \in I$ and $\text{wt}(\mu_\lambda) = \lambda$. It encodes the structure of the crystal basis of the $U_q(\mathfrak{g})$-irreducible representation with highest weight $\lambda$ as $q$ goes to $0$.

A Kirillov-Reshetikhin (KR) crystal [12] is a finite crystal $B^{r,s}$ for an affine algebra, associated to a rectangle of height $r$ and length $s$. We now describe the KR crystals $B^{r,1}$ for type $A_{n-1}^{(1)}$ (where $r \in \{1, 2, \ldots, n-1\}$) as well as for types $B_n^{(1)}, C_n^{(1)}$, and $D_n^{(1)}$ (where $r \in \{1, 2, \ldots, n\}$). As a classical type crystal (i.e. with the removal of the $f_0$ arrows) in types $A_{n-1}^{(1)}$ and $C_n^{(1)}$, we have that the KR crystal $B^{r,1}$ is isomorphic to the corresponding highest weight crystal $B(\omega_r)$. Similarly, in types $C_n^{(1)}$ and $D_n^{(1)}$, we have that the KR crystal $B^{r,1}$, as a classical type crystal, is isomorphic to the crystal $B(\omega_r) \sqcup B(\omega_{r-2}) \sqcup B(\omega_{r-4}) \sqcup \ldots$ where each $B(\omega_k)$ is given by Kashiwara-Nakashima (KN) columns of height $k$ of the corresponding type.

**Definition 2.2.** Kashiwara-Nakashima (KN) columns of height $k$ are strictly increasing fillings of length $k$ columns with entries $\{1 < 2 < \ldots < n\}$ in type $A_{n-1}$ and with entries $\{1 < \ldots < n < \bar{n} < \ldots < \bar{1}\}$ in type $C_n, B_n$, and $D_n$ along with the additional conditions:

1. The entries are strictly increasing from the top to bottom with the exception that:
   (a) the letter $0$ (ordered between $n$ and $\bar{n}$) can appear in type $B_n$ and can be repeated, and
   (b) the letters $n$ and $\bar{n}$ in type $D_n$ can alternate.

2. If both letters $j$ and $\bar{j}$ appear in the column, and $j$ is in the $a$-th box from the top and $\bar{j}$ lies in the $b$-th box from the bottom, then $a + b \leq j$.

**Example 2.1.** A column of height 5 of type $B_6$

$$\begin{array}{cccc}
2 & 3 & 0 & 0 \\
2 & & & \\
\end{array}$$

For our purposes here, we would like to consider fillings of partition shapes, rather than just rectangular. We define the associated crystal as follows.
Definition 2.3. For a partition $p = (p_1 \geq p_2 \geq \ldots \geq p_r)$, we define
$$B^p := B^{p_1,1} \otimes B^{p_2,1} \otimes \ldots \otimes B^{p_r,1}.$$  

2.3. The quantum alcove model. We say that two alcoves are adjacent if they are distinct and have a common wall. Given a pair of adjacent alcoves $A$ and $B$, we write $A \xrightarrow{\beta} B$ if the common wall is of the form $H_{\beta,k}$ and the root $\beta \in \Phi$ points in the direction from $A$ to $B$.

Definition 2.4. [24]

An alcove path is a sequence of alcoves $(A_0, A_1, \ldots, A_m)$ such that $A_{j-1}$ and $A_j$ are adjacent, for $j = 1, \ldots, m$. We say that an alcove path is reduced if it has minimal length among all alcove paths from $A_0$ to $A_m$.

Definition 2.5. [24]

The sequence of roots $(\beta_1, \beta_2, \ldots, \beta_m)$ is called a $\lambda$-chain if
$$A_0 = A_0 \xrightarrow{-\beta_1} A_1 \xrightarrow{-\beta_2} \ldots \xrightarrow{-\beta_m} A_m = A_\lambda$$
is a reduced alcove path.

We now fix the dominant weight $\lambda = \omega_{p_1} + \ldots + \omega_{p_r}$ and an alcove path $\Pi = (A_0, \ldots, A_m)$ from $A_0 = A_0$ to $A_m = A_\lambda$. Note that $\Pi$ is determined by the corresponding $\lambda$-chain of positive roots $\Gamma := (\beta_1, \ldots, \beta_m)$. We let $r_i := s_{\beta_i}$ and $J = \{j_1, j_2, \ldots, j_s\} \subseteq [m]$. The elements of $J$ are called folding positions. We fold $\Pi$ in the hyperplanes corresponding to these positions and obtain a folded path. Like $\Pi$, the folded path can be recorded by a sequence of roots, namely $\Delta = \Gamma(J) = (\gamma_1, \gamma_2, \ldots, \gamma_m)$; here $\gamma_k = r_{j_1}r_{j_2}\ldots r_{j_p}(\beta_k)$, with $j_p$ the largest folding position less than $k$. Given $i \in J$, we say that $i$ is a positive folding position if $\gamma_i > 0$, and a negative folding position if $\gamma_i < 0$. We denote the positive folding positions by $J^+$ and the negative ones by $J^-$.  

Definition 2.6. [19]

A subset $J = \{j_1 < j_2 < \ldots < j_s\} \subseteq [m]$ (possibly empty) is an admissible subset if we have the following path in the quantum Bruhat graph on $W$:
$$1 \xrightarrow{\beta_1} r_{j_1} \xrightarrow{\beta_2} r_{j_1}r_{j_2} \xrightarrow{\beta_3} \ldots \xrightarrow{\beta_s} r_{j_1}r_{j_2}\ldots r_{j_s}.$$  

We call $\Delta = \Gamma(J)$ an admissible folding. Let $A = A(\mu)$ be the collection of all admissible subsets.

See Example 3.1 for an example of a $\Gamma$-chain and an admissible subset.

Theorem 2.7. [22] The collection of all admissible subsets $A(\lambda)$ is a combinatorial model for $B^p$.  

3. The bijection in types $A_{n-1}$ and $C_n$

3.1. The quantum alcove model and filling map in type $A_{n-1}$. We start with the basic facts about the root system for type $A_{n-1}$. We can identify the space $\mathfrak{h}_R^*$ with the quotient $V := \mathbb{R}^n / \mathbb{R}(1, \ldots, 1)$, where $\mathbb{R}(1, \ldots, 1)$ denotes the subspace in $\mathbb{R}^n$ spanned by the vector $(1, \ldots, 1)$. Let $\varepsilon_1, \ldots, \varepsilon_n \in V$ be the images of the coordinate vectors in $\mathbb{R}^n$. The root system is $\Phi = \{\alpha_{ij} := \varepsilon_i - \varepsilon_j : i \neq j, 1 \leq i, j \leq n\}$. The simple roots are $\alpha_i = \alpha_{i,i+1}$, for $i = 1, \ldots, n - 1$. The weight lattice is $\Lambda = \mathbb{Z}^n / \mathbb{Z}(1, \ldots, 1)$. The fundamental weights are $\omega_i = \varepsilon_1 + \varepsilon_2 + \ldots + \varepsilon_i$, for $i = 1, 2, \ldots, n - 1$. A dominant weight $\lambda = \lambda_1\varepsilon_1 + \ldots + \lambda_{n-1}\varepsilon_{n-1}$ is identified with the partition $(\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n-1} \geq \lambda_n = 0)$ having at most $n - 1$ parts. Note that $\rho = (n-1, n-2, \ldots, 0)$. Considering the Young diagram of the dominant weight $\lambda$ as a concatenation of columns, whose heights are $\lambda'_1, \lambda'_2, \ldots$, corresponds to expressing $\lambda$ as $\omega_{\lambda'_1} + \omega_{\lambda'_2} + \ldots$ (as usual, $\lambda'$ is the conjugate partition to $\lambda$).

The Weyl group $W$ is the symmetric group $S_n$, which acts on $V$ by permuting the coordinates $\varepsilon_1, \ldots, \varepsilon_n$. Permutations $w \in S_n$ are written in one-line notation $w = w(1) \ldots w(n)$. For simplicity,
we use the same notation \((i, j)\), with \(1 \leq i < j \leq n\), for the positive root \(\alpha_{ij}\) and the reflection \(s_{\alpha_{ij}}\), which is the transposition \(t_{ij}\) of \(i\) and \(j\).

We now consider the specialization of the quantum alcove model to type \(A_{n-1}\). For any \(k = 1, \ldots, n - 1\), we have the following \(\omega_k\)-chain, denoted by \(\Gamma(k)\): 

\[
( (k, k + 1), (k, k + 2), \ldots, (k, n), \ldots, (2, k + 1), (2, k + 2), \ldots, (2, n), (1, k + 1), (1, k + 2), \ldots, (1, n) ).
\]

We construct a \(\lambda\)-chain \(\Gamma = (\beta_1, \beta_2, \ldots, \beta_m)\) as the concatenation \(\Gamma := \Gamma_1 \ldots \Gamma_{\lambda_1}\), where \(\Gamma_i := \Gamma(\lambda_i')\). Let \(J = \{j_1 < \ldots < j_s\}\) be a set of folding positions in \(\Gamma\), not necessarily admissible, and let \(T\) be the corresponding list of roots of \(\Gamma\). The factorization of \(\Gamma\) induces a factorization on \(T\) as \(T = T_1 T_2 \ldots T_{\lambda_1}\). We denote by \(T_1 \ldots T_i\) the permutation obtained by multiplying the transpositions in \(T_1, \ldots, T_i\) considered from left to right. For \(w \in W\), written \(w = w_1 w_2 \ldots w_n\), let \(w[i, j] = w_i \ldots w_j\). To each \(J\) we can associate a filling of a Young diagram \(\lambda\), as follows.

**Definition 3.1.** Let \(\pi_i = \pi_i(T) := T_1 \ldots T_i\). We define the filling map, which produces a filling of the Young diagram \(\lambda\), by \(fill_A(J) = fill_A(T) := C_1 \ldots C_{\lambda_1}\), where \(C_i := \pi_i[1, \lambda_i']\). We define the sorted filling map \(sfill_A(J)\) to be the composition \(sort \circ fill_A(J)\), where \(sort\) reorders increasingly each column of \(fill_A(J)\).

**Definition 3.2.** Define a circular order \(\prec_i\) on \([n]\) starting at \(i\), by 

\[i \prec_i i + 1 \prec_i \ldots \prec_i n \prec_i 1 \prec_i \ldots \prec_i i - 1.\]

It is convenient to think of this order in terms of the numbers \(1, \ldots, n\) arranged on a circle clockwise. We make that convention that, whenever we write \(a < b < c < \ldots\), we refer to the circular order \(\prec\). Below is a criterion for the quantum Bruhat graph on the Weyl group \(W\), \(\mathrm{QBG}(W)\), in type \(A_{n-1}\) using these orders.

**Proposition 3.3.** [18] For \(1 \leq i < j \leq n\), we have an edge \(w \xrightarrow{(i,j)} w(i, j)\) in \(\mathrm{QBG}(W)\) if and only if there is no \(k\) such that \(i < k < j\) and \(w(i) \prec w(k) \prec w(j)\).

**Example 3.1.** Consider the dominant weight \(\lambda = 3\varepsilon_1 + 2\varepsilon_2 = \omega_1 + 2\omega_2\) in the root system \(A_2\), which corresponds to the Young diagram \(\begin{array}{c} \Box \end{array}\). The corresponding \(\lambda\)-chain is 

\[\Gamma = \Gamma_1 \Gamma_2 \Gamma_3 = \Gamma(2) \Gamma(2) \Gamma(1) = \{(2, 3), (1, 3)(2, 3), (3, 1)(2, 3), (3, 1)(2, 3)\} \}

Consider \(J = \{1, 2, 3, 5\}\), cf. the underlined roots, with \(T = \{(2, 3), (1, 3)(2, 3)(1, 2)\}\).

We write the permutations in Definition 2.6 as broken columns. Note that \(J\) is admissible since, based on Proposition 3.3, we have

\[
\begin{array}{cccccc}
1 & 2 & \prec & 3 & \prec & 2 \prec 1 \prec 2 \prec 1 \\
3 & 2 & \prec & 1 & \prec & 3 \prec 1 \prec 3 \prec 2
\end{array}
\]

where the symbols \(\prec\) and \(\prec\) signify Bruhat coverings as given in Section 2.1. By considering the top part of the last column in each segment, and by concatenating these columns left to right, we obtain \(fill_A(J) = \begin{array}{cccc}
2 & 2 & 1 \\
3 & 1
\end{array}\) and \(sfill_A(J) = \begin{array}{cccc}
2 & 1 & 1 \\
3 & 2
\end{array}\).

**Theorem 3.4.** [18] [19] The map \("sfill_A"\) is an affine crystal isomorphism between \(A(\lambda)\) and \(B^{\lambda} := B^{\lambda_1} \otimes B^{\lambda_2} \otimes \ldots\).
The proof of bijectivity is given in [18] by constructing an inverse map. We will now present the algorithm for constructing this map, as the corresponding construction in the other classical types is based on this algorithm.

3.2. The inverse map in type $A_{n-1}$. Consider $B^\lambda := B^\lambda_1 \otimes B^\lambda_2 \otimes \cdots = B(\omega_{\lambda_1}) \otimes B(\omega_{\lambda_2}) \otimes \cdots$. This is simply the set of column-strict fillings of the Young diagram $\lambda$ with integers in $[n]$. Fix a filling $b$ in $B^\lambda$ written as a concatenation of columns $b_1 \ldots b_{\lambda_1}$.

The algorithm for mapping $b$ to a sequence of roots $S \subset \Gamma$ consists of two sub-algorithms, which we call the Reorder algorithm (this reverses the ordering of columns $b_i$ from sort back to that of the corresponding column in the fill_A map) and the Path algorithm (this provides the corresponding path in the quantum Bruhat graph).

The Reorder algorithm (Algorithm 3.1) takes $b$ as input and outputs a filling $\text{ord}_A(b) = C$, a reordering of the column entries, based on the circle order given in Definition 3.2.

Algorithm 3.1. ("ord_A")

Let $C_1 := b_1$;

for $i$ from $2$ to $\lambda_1$ do

for $j$ from $1$ to $\lambda'_i$ do

let $C_i(j) := \min_{A_i \neq C_i(l)}(b_i \setminus \{C_i(1), \ldots, C_i(j-1)\})$

end do;

end do;

return $C := C_1 \ldots C_{\lambda_1}$.

Example 3.2. Algorithm 3.1 gives the filling $C$ from below.

$$b = \begin{array}{ccc}
3 & 2 & 1 \\
5 & 3 & 2 \\
6 & 4 & 4
\end{array} \xrightarrow{\text{ord}_A} \begin{array}{ccc}
3 & 3 & 4 \\
5 & 2 & 2 \\
6 & 4 & 1
\end{array} = C$$

The path algorithm (Algorithm 3.2) takes the reordered filling $C$ and outputs a sequence of roots $\text{Path}_A(C) = S \subset \Gamma$. Let $C_0$ be the increasing column filled with $1, 2, \ldots, n$.

Algorithm 3.2. ("Path_A")

for $i$ from $1$ to $\lambda_1$ do

let $S_i := \emptyset$, $A := C_{i-1}$;

for $(l, m)$ in $\Gamma_i$ do

if $A(l) \neq C_i(l)$ and $A(l) < A(m) < C_i(l)$ then let $S_i := S_i \cup \{(l, m)\}$ and $A := A(l, m)$;

end if;

end do;

end do;

return $S := S_1 \ldots S_{\lambda_1}$.

Example 3.3. Consider $b = \begin{array}{ccc}
1 & 1 & 2 \\
3 & 2 & 4
\end{array} \in B^{(3,2,1)}$, where $\lambda = \lambda' = (3, 2, 1)$ and $n = 4$. We have

$$\Gamma = \Gamma(3)\Gamma(2)\Gamma(1) = \{(3, 4), (2, 4), (1, 4)\}(2, 3), (2, 4), (1, 3), (1, 4)\}(1, 2), (1, 3), (1, 4)\}.$$ 

Notice that $\text{ord}_A(b) = b$, and that $\text{Path}_A \circ \text{ord}_A(b)$ outputs

$S = S_1S_2S_3 = \{(3, 4), (2, 4)\}(2, 3), (2, 4)\}(1, 2)\}$ via the following path in QBG(W):

$$\begin{array}{ccccccccc}
1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\
\xrightarrow{(3,4)} & \xrightarrow{(2,4)} & \xrightarrow{(2,3)} & \xrightarrow{(2,4)} & \xrightarrow{(1,2)} & \\
\end{array}$$
Theorem 3.5. [18] If fill\_A(T) = C, then the output of the Greedy algorithm C \rightarrow S is such that S = T. Moreover, the map “Path\_A \circ ord\_A” is the inverse of “sfill\_A”.

3.3. The quantum alcove model and filling map in type \( C_n \). We start with the basic facts about the root system for type \( C_n \). We can identify the space \( h^*_R \) with \( V := \mathbb{R}^n \), with coordinate vectors \( \varepsilon_1, \ldots, \varepsilon_n \in V \). The root system is \( \Phi = \{ \pm \varepsilon_i \pm \varepsilon_j : i \neq j, 1 \leq i < j \leq n \} \cup \{ \pm 2 \varepsilon_i : 1 \leq i \leq n \} \).

The Weyl group \( W \) is the group of signed permutations \( B_n \), which acts on \( V \) by permuting the coordinates and changing their signs. A signed permutation is a bijection \( w \) from \( \mathbb{Z}^n \) to \( \mathbb{Z}^n \) which satisfies \( w(\bar{i}) = \bar{w(i)} \). Here, \( \bar{i} \) is viewed as \( -i \), so that \( \bar{i} = i \), and we can define \( |i| \) and \( \text{sign}(i) \in \{ \pm 1 \} \), for \( i \in \mathbb{Z}^n \). We will use the so-called window notation \( w = w(1)w(2) \ldots w(n) \). For simplicity, given \( 1 \leq i < j \leq n \), we denote by \( (i, j) \) the roots \( \varepsilon_i - \varepsilon_j \) and \( \varepsilon_i + \varepsilon_j \), respectively; the corresponding reflections, denoted in the same way, are identified with the composition of transpositions \( t_{ij}t_{\bar{i}j} \) and \( t_{ij}t_{\bar{i}j} \), respectively. Finally, we denote by \( (i, \bar{i}) \) the root \( 2\varepsilon_i \) and the corresponding reflection, identified with the transposition \( t_{ii} \).

We now consider the specialization of the quantum alcove model to type \( C_n \). For any \( k = 1, \ldots, n \), we have the following (split) \( \omega_k \)-chain, denoted by \( (\Gamma^i(k)) \Gamma^r(k) \) [18], where:

\[
\begin{align*}
\Gamma^i(k) := & \Gamma^{k} \ldots \Gamma^{k}, \quad \Gamma^r(k) := \Gamma^{k} \ldots \Gamma^{2}, \\
\Gamma^{ki} := & \left( (i, k + 1), (i, k + 2), \ldots, (i, n), \\
& (i, \bar{i}), \\
& (i, \bar{n}), \ldots, (i, k + 1), \\
& (i, \bar{i} - 2), \ldots, (i, \bar{1}) \right), \\
\Gamma^i := & \left( (i, \bar{i} - 1), (i, \bar{i} - 2), \ldots, (i, \bar{1}) \right).
\end{align*}
\]

We refer to the four rows above in \( \Gamma^{ki} \) as stages I, II, III, and IV respectively. We can construct a \( \lambda \)-chain as a concatenation \( \Gamma := \Gamma^i(\lambda_1) \Gamma^r(\lambda_1) \Gamma^i(\lambda_2) \Gamma^r(\lambda_2) \), where \( \Gamma^i := \Gamma^i(\lambda_i) \) and \( \Gamma^r := \Gamma^r(\lambda_i) \). We will use interchangeably the set of positions \( J \) in the \( \omega \)-chain \( \Gamma \) and the sequence of roots \( T \) in \( \Gamma \) in those positions, which we call a folding sequence. The factorization of \( \Gamma \) with factors \( \Gamma^i, \Gamma^r \) induces a factorization of \( T \) with factors \( T^i, T^r \). We define the circle order \( \prec_\lambda \) in a similar way to Definition 3.2 but on the set \( \mathbb{Z}^n \). Below is a criterion for QBG(\( W \)) in type \( C_n \), analogous to Proposition 3.3.

Proposition 3.6. [18] The quantum Bruhat graph of type \( C_n \) has the following edges:

1. given \( 1 \leq i < j \leq n \), we have an edge \( \overset{(i,j)}{w} \rightarrow (i,j) \) if and only if there is no \( k \) such that \( i < k < j \) and \( w(i) < w(k) < w(j) \);

2. given \( 1 \leq i < j \leq n \), we have an edge \( \overset{(ij)}{w} \rightarrow (i,j) \) if and only if \( w(i) < w(j) \), \( \text{sign}(w(j)) = \text{sign}(w(\bar{j})) \), and there is no \( k \) such that \( i < k < j \) and \( w(i) < w(k) < w(\bar{j}) \);

3. given \( 1 \leq i \leq n \), we have an edge \( \overset{(i\bar{i})}{w} \rightarrow (i, \bar{i}) \) if and only if there is no \( k \) such that \( i < k < \tau \) (or equivalently, \( i < k \leq n \)) and \( w(i) < w(k) < \tau \).

Definition 3.7. Given a folding sequence \( T \), we consider the signed permutations \( \pi^i := T^i_1T^i_1 \ldots T^i_{i-1}T^i_{i-1}T^i_1 \), \( \pi^r := \pi^i \pi^r \). Then the filling map is the map “fill\_C” from folding sequences \( T \) in \( A(\lambda) \) to fillings fill\_C(T) = \( C^i_1C^i_1 \ldots C^i_{\lambda_1}C^r_{\lambda_1} \) of the shape \( 2\lambda \), which are viewed as concatenations of columns; here \( C^i_i := \pi^i_1[1, \lambda_i] \) and \( C^r_i := \pi^r_1[1, \lambda_i] \), for \( i = 1, \ldots, \lambda_1 \). We then define sfill\_C : \( A(\lambda) \rightarrow B^{\lambda} \) to be the composition “sort \circ fill\_C”, where “sort” reorders the entries of each column increasingly; here we represent a KR crystal \( B^{\lambda} \) as a split (also known as doubled) \( K \)N column of height \( r \), see Section 3.4.

Theorem 3.8. [18] [19] The map “sfill\_C” is an affine crystal isomorphism between \( A(\lambda) \) and \( B^{\lambda} \).
3.4. The inverse map in type $C_n$. Recall from the construction of the filling map in type $A_{n-1}$ that we treated the columns of a filling as initial segments of permutations. However, the KN columns of type $C_n$ allow for both $i$ and $\overline{i}$ to appear as entries in such a column. In order to pursue the analogy with type $A_{n-1}$, cf. Definition 3.7, we need to replace a KN column with its split version, i.e., two columns of the same height as the initial column. The splitting procedure, described below, gives an equivalent definition of KN columns, see Section 2.2.

**Definition 3.9.** Let $C$ be a column and $I = \{z_1 > \ldots > z_r\}$ be the set of unbarred letters $z$ such that the pair $(z, \overline{z})$ occurs in $C$. The column $C$ can be split when there exists a set of $r$ unbarred letters $J = \{t_1 > \ldots > t_r\} \subset \{1, \ldots, n\}$ such that $t_1$ is the greatest letter in $\{1, \ldots, n\}$ satisfying: $t_1 < z_1, t_1 \notin C$, and $\overline{t_1} \notin C$, and for $i = 2, \ldots, r$, the letter $t_i$ is the greatest value in $\{1, \ldots, n\}$ satisfying $t_i < \min(t_{i-1}, z_i), t_i \notin C$, and $\overline{t_i} \notin C$. In this case we write:

1. $rC$ for the column obtained by changing $\overline{z}_i$ into $\overline{t}_i$ in $C$ for each letter $z_i \in I$, and by reordering if necessary,
2. $lC$ for the column obtained by changing $z_i$ into $t_i$ in $C$ for each letter $z_i \in I$, and by reordering if necessary.

The pair $(lC, rC)$ is then called a split (or doubled) column.

Given our fixed dominant weight $\lambda$, an element $b$ of $B^\lambda$ can be viewed as a concatenation of KN columns $b_1 \ldots b_{\lambda_1}$, with $b_i$ of height $\lambda_i$. Let $b' := b'_1 b'_2 \ldots b'_{\lambda_1} b'_{\lambda_1}$ be the associated filling of the shape $2\lambda$, where $(b'_i, b'_{i+1}) := (lb_i, rb_i)$ is the splitting of the KN column $b_i$.

The algorithm for mapping $b'$ to a sequence of roots $S \subset \Gamma$ is similar to the type $A_{n-1}$ one. The Reorder algorithm “ord.C” for type $C_n$ is the obvious extension from type $A_{n-1}$. The path algorithm “Path.C" is also similar to its type $A_{n-1}$ counterpart, but merits discussion. Recall that an $\omega_k$-chain in type $C_n$ factors as $\Gamma'(k)\Gamma'(k)$. While the path algorithm parses through $\Gamma'(k)$, it outputs a chain from the previous right column to the current left column reordered. While the path algorithm parses through $\Gamma'(k)$, it outputs a chain between the current left and right columns, both reordered.

**Theorem 3.10.** The map “Path.C \circ ord.C \circ split.C" is the inverse of the type $C_n$ “sfill.C" map.

4. The bijection in type $B_n$

We now move to the main content of this paper: extending the work done in types $A_{n-1}$ and $C_n$ to both types $B_n$ and $D_n$. The filling map naturally extends to all classical types, however the corresponding inverse maps become more interesting as we progress to type $B_n$, and further still with type $D_n$. The changes in the inverse maps are direct consequences of differences between the corresponding structure of the KN columns, as well as differences in the quantum Bruhat graphs.

4.1. The type $B_n$ Kirillov-Reshetikhin crystals. We begin by recalling the basic facts of the type $B_n$ root system. Similar to type $C_n$, we can identify the space $\mathfrak{h}_R^*$ with $V := \mathbb{R}^n$, with coordinate vectors $\varepsilon_1, \ldots, \varepsilon_n \in V$. The root system is $\Phi = \{\pm \varepsilon_i \pm \varepsilon_j : i \neq j, 1 \leq i < j \leq n\} \cup \{\pm \varepsilon_i : 1 \leq i \leq n\}$.

The Weyl group $W$ is the group of signed permutations $B_n$, which acts on $V$ by permuting the coordinates and changing their signs. We again note that a signed permutation is a bijection $w$ from $[\overline{n}] := \{1 < 2 < \ldots < n < \overline{n} < \overline{n-1} < \ldots < 1\}$ to $[\overline{n}]$ which satisfies $w(\overline{i}) = \overline{w(i)}$. Here, $\overline{i}$ is viewed as $-i$, so that $\overline{i} = i$, and we can define $|i|$ and $\text{sign}(i) \in \{\pm 1\}$, for $i \in [\overline{n}]$ in the obvious way. We will use the so-called window notation $w = w(1)w(2)\ldots w(n)$. For simplicity, given $1 \leq i < j \leq n$, we denote by $(i, j)$ and $(i, \overline{j})$ the roots $\varepsilon_i - \varepsilon_j$ and $\varepsilon_i + \varepsilon_j$, respectively; the corresponding reflections, denoted in the same way, are identified with the composition of transpositions $t_{ij}t_{\overline{ij}}$ and $t_{ij}t_{\overline{ij}}$, respectively.
where each $B^{r,1}$ is a column shape type $B_n$ Kirillov-Reshitikhin crystal. When viewed as a classical type crystal, we have

$$B^{r,1} = B(\omega_r) \sqcup B(\omega_{r-2}) \sqcup B(\omega_{r-4}) \sqcup \ldots$$

where, as before, the elements of the set $B(\omega_k)$ are given by KN columns of height $k$. As in type $C_n$, the KN columns in type $B_n$ are allowed to contain both $i$ and $\tau$ values; they may also contain the value 0. This is addressed in the type $B_n$ splitting algorithm “split $B$” by adding the 0 values in the column to the set $I$ (see Definition 3.9), and then by proceeding as in type $C_n$ [15]. As it was in Type $C_n$, it will be useful to realize the tensor factors $B^{k,1}$ in terms of split columns of height $k$.

**4.2. The quantum alcove model and filling map in type $B_n$.** We now consider the specialization of the quantum alcove model to type $B_n$. For any $k = 1, \ldots, n$, we define the following (split) $\omega_k$-chain, denoted by $\Gamma^\ell(k)\Gamma^r(k)$ [18], similarly to type $C_n$ where:

\[
\Gamma^\ell(k) := \Gamma^k \ldots \Gamma^{k_1}, \quad \Gamma^r(k) := \Gamma^k \ldots \Gamma^2,
\]

\[
\Gamma^{k_i} := ((i,k+1), (i,k+2), \ldots, (i,n), (i,\tau), (i,\bar{n}), (i,\bar{n}-1), \ldots, (i,k+1), (i,i-1), (i,i-2), \ldots, (i,\bar{1})),
\]

\[
\Gamma^i := ((i,i-1), (i,i-2), \ldots, (i,\bar{1})).
\]

We will continue to refer to the four rows above in $\Gamma^{k_i}$ as stages I, II, III, and IV respectively (c.f. Figure 2). We can construct a $\lambda$-chain as a concatenation $\Gamma := \Gamma^1_{\ell} \Gamma^2_{\ell} \ldots \Gamma^r_{\ell} \Gamma^r_{i}$, where $\Gamma^i_{\ell} := \Gamma^\ell(\lambda^i)$ and $\Gamma^i := \Gamma^r(\lambda^i)$. We will use interchangeably the set of positions $J$ in the $\lambda$-chain $\Gamma$ and the sequence of roots $T$ in $\Gamma$ in those positions, which we call a folding sequence. The factorization of $\Gamma$ with factors $\Gamma^i_{\ell}, \Gamma^i$ induces a factorization of $T$ with factors $T_i^\ell, T_i^r$. We use the same circle order $\prec_a$ on the set $[\bar{n}]$ as the one in type $C_n$.

The following are conditions on the quantum Bruhat graph of type $B_n$.

**Proposition 4.1.** [1] The quantum Bruhat graph of type $B_n$ has the following edges.

1. Given $1 \leq i < j \leq n$, we have an edge $w \xrightarrow{(i,j)} w(i,j)$ if and only if there is no $k$ such that $i < k < j$ and $w(i) \prec w(k) \prec w(j)$.

2. Given $1 \leq i < j \leq n$, we have an edge $w \xrightarrow{(i,\bar{j})} w(i,\bar{j})$ if and only if one of the following conditions holds:
   (a) $w(i) < w(\bar{j})$, $\text{sign}(w(i)) = \text{sign}(w(\bar{j}))$, and there is no $k$ such that $i < k < \bar{j}$ and $w(i) < w(k) < w(\bar{j})$;
   (b) $\text{sign}(w(i)) = -1$, $\text{sign}(w(\bar{j})) = 1$, and there is no $k$ such that $i < k \neq j < \bar{j}$ and $w(i) < w(k) < w(\bar{j})$.

3. Given $1 \leq i \leq n$, we have an edge $w \xrightarrow{(i,\bar{i})} w(i,\bar{i})$ if and only if:
   (a) $w(i) < w(\bar{i})$ and there is no $k$ such that $i < k \neq \bar{i}$ and $w(i) < w(k) < w(\bar{i})$;
   (b) or $w(\bar{i}) < w(i)$ and $i = n$. 

Note that there are two major differences from the type $C_n$ quantum Bruhat graph criterion.

1. Since the root $(n, \pi)$ is not in $\Gamma(k)$ for any $k < n$, we will never have the case 3b. This means that we lose the ability to apply the transposition $t_{i\overline{i}}$ when $w(\overline{i}) < w(i)$.

2. In return, we gain some extra transpositions through 2b, as there are now cases where the quantum Bruhat graph criterion allows for an arrow $(i, j)$ when $w(i) > w(j)$.

We now provide a description of the filling map in type $B_n$. Note that it is the natural extension from type $C_n$.

**Definition 4.2.** Given a folding sequence $T$, we consider the signed permutations $\pi_1^i := T_1^i T_{i-1}^i \ldots T_{i-k}^i T_i^i$, $\pi^i_i := \pi_i^i T_i^i$. Then the filling map is the map “fill $B$” from folding sequences $T$ in $A(\lambda)$ to fillings $\text{fill}_B(T) = C_1^i C_1^{i'} \ldots C_1^{\lambda_1}$ of the shape $2\lambda$, which are viewed as concatenations of columns; here $C_i^i := \pi_i^i[1, \lambda_i]$ and $C_i^i := \pi_i^i[1, \lambda_i^i]$, for $i = 1, \ldots, \lambda_1$. We then define $\text{fill}_B : A(\lambda) \rightarrow B^{\lambda^\prime}$ to be the composition “sort $\circ$ fill $B$”, where “sort” reorders the entries of each column increasingly; here we represent a KR crystal $B^{k,1}$ as a split (also known as doubled) $KN$ column of height $k$, see Section 4.1.

4.3. The type $B_n$ inverse map. The main process remains similar to that of the type $C_n$ inverse map. However, the differences in the type $B_n$ quantum Bruhat graph, $B^{r,1}$ Kirillov-Reshetikhin crystals, and $KN$ columns add key new features. The fact that type $B_n$ columns can contain the value 0 was acknowledged in Section 4.1 with the revised algorithm split $B$. We now address the difficulties associated to the QBG criterion and KR crystals.

Recall that the KR crystals of column shape can be written as $B^{k,1} \cong B(\omega_k) \sqcup B(\omega_{k-2}) \sqcup B(\omega_{k-4}) \sqcup \ldots$ where the elements of the set $B(\omega_r)$ are given by $KN$ columns of height $r$. This means that each $B^{k,1}$ contains columns of height less than $k$. We need to extend them to full height $k$ so that the transpositions of the corresponding $\Gamma^l(k)\Gamma^r(k)$ may be correctly applied. The respective algorithm “extend” is given below.

**Algorithm 4.1.** [1] Given a split column $(lC, rC)$ of length $1 \leq r < n$ and $r \leq k < n$, append $\{\overline{t_1} < \ldots < \overline{t}_{r-k}\}$ to $lC$ and $\{i_1 < \ldots < i_{r-k}\}$ to $rC$, where $i_1$ is the minimal value in $[\pi]$ such that $i_1, \overline{t_1} \notin lC, rC$, and $i_t$ for $2 \leq t \leq r - k$ is minimum value in $[\pi]$ such that $i_t, \overline{t_t} \notin lC, rC$ and $i_t > i_{t-1}$. Sort the extended columns increasingly. Let $(\hat{l}C, \hat{r}C)$ be the extended split column.

**Example 4.1.** The following is a $KN$ column $C$ of type $B_8$ with its split columns and then extended columns to a height of 6.
Recall the type $B_n$ quantum Bruhat graph criterion (cf. Proposition $4.1$). The main differences from the type $C_n$ QBG are the loss of the ability to change negative to positive entries with the stage II roots, $(i,7)$, but gaining the ability to change negative to positive entries with the stage IV roots, $(i,7)$. At first glance, this does not seem to hinder the path algorithm $Path.C$ from Section $3.4$ if such a sign change is necessary, it is merely postponed. However, while the $(i,7)$ root only changes the sign in position $i$, the $(i,3)$ root changes the sign in position $j$ as well. The subtle difference in the quantum Bruhat criterion makes both the reorder and path algorithms from type $C_n$ fail in type $B_n$. We discuss two modifications to these algorithms, which depend on a certain pattern avoidance in two adjacent columns.

**Remark 4.3.** In types $A_{n-1}$ and $C_n$, the algorithm for forming the correct sequence of roots followed the rule that for the current word, $w$, and the root $(i,j)$, if $w(i) < wt_{ij}(i) \leq C(i)$, then add the root to the sequence, and otherwise do not and proceed. We will call a transposition following the above inequality a $Path.C$ transposition. Further, if $w(i) < C'(i) < wt_{ij}(i)$, we will say that the transposition passes the target (in row $i$). The original path algorithm has two underlying rules: one, we never apply a root which forces us to pass the target, and two, we always use a $Path.C$ root. In type $B_n$ (and, as we will later see, in type $D_n$ as well) there are exceptions to these two rules. Both exceptions come directly from the need to avoid the following pattern in two adjacent columns.

**Definition 4.4.** We say that columns $C = (l_1, l_2, \ldots, l_k)$ and $C' = (r_1, r_2, \ldots, r_k)$ are blocked off at $i$ by $b := r_i$ if and only if the following hold:

1. $|l_i| \leq b < n$, where $|l_i| = b$ if and only if $l_i = \bar{b}$;
2. $\{1, 2, \ldots, b\} \subset \{l_1, l_2, \ldots, l_i\}$ and $\{1, 2, \ldots, b\} \subset \{r_1, r_2, \ldots, r_i\}$;
3. $|\{j : 1 \leq j \leq i, l_j < 0, r_j > 0\}|$ is odd.

**Example 4.2.** The following columns $CC'$ of height $5$ with entries from $[8]$ are blocked-off at $4$ by $3$:

\[
\begin{array}{cccc}
1 & 1 & 4 & 5 \\
\bar{5} & \bar{2} & \bar{3} & \bar{8} \\
\end{array}
\]

We will find that if two columns (of length $k$) are blocked off at any $i \in [k-1]$, there will be no corresponding path in the quantum Bruhat graph between the columns. Therefore, given two columns $C$ and $C'$, we must not only avoid this pattern in the reordering of column $C'$, but also make sure that at any point in the application of roots in $T$, we never force the current column to be blocked off with $C'$ at any $i \in [k-1]$ either. The latter part will give way to the $Path.B$ algorithm.

We now define the type $B_n$ versions of the reorder and path algorithms. Let $b := b_1 b_1' \ldots b_k b_k' = b_1 \ldots b_{2\lambda_1}$ be extended split columns indexing a vertex of the crystal $B_{\lambda^T}$ of type $B_n$. Similarly, let $\Gamma := \Gamma_1 \Gamma_1' \ldots \Gamma_{\lambda_1} \Gamma_{\lambda_1}' = \Gamma_1 \ldots \Gamma_{2\lambda_1}$. Algorithm $4.2$ takes $b, \Gamma$ as input and returns a reordered filling $C$ of a Young diagram of shape $2\lambda$. 

\[
\begin{array}{ccc}
5 & 0 & 8 \\
\bar{5} & 8 & \bar{5} \\
\end{array}
\]

\[
\begin{array}{ccc}
4 & 5 & 7 \\
8 & 7 & \bar{5} \\
\end{array}
\]

\[
\begin{array}{ccc}
4 & 5 & 8 \\
\bar{7} & \bar{8} & \bar{1} \\
\end{array}
\]
Algorithm 4.2. (“ord\_B”)  
let $C_1 := b_1$;  
for $i$ from 2 to $2\lambda_1$ do  
for $j$ from 1 to $\lambda'_i - 1$ do  
let $C_i(j) := \min_{<_{C_i-1}(j)}(b_i \setminus \{C_i(1), \ldots, C_i(j - 1)\})$ so that $C_{i-1}, C_i$ not blocked off at $j$  
end do;  
let $C_i(\lambda'_i) := \min_{<_{C_i-1}(j)}(b_i \setminus \{C_i(1), \ldots, C_i(\lambda'_i - 1)\})$  
end do;  
return $C := C_1 \ldots C_{2\lambda_1} = C_1^l C_1^r \ldots C_{\lambda_1}^l C_{\lambda_1}^r$.

Example 4.3. Algorithm 4.2 gives the filling $C$ from $b$ below. Note that Algorithm 3.1 would have paired the 3 with the 3 in the 4th row. However, this would cause the two columns to be blocked off at 4 by 3, so the modified algorithm skips to the next value and pairs the 8 with the 3 instead.

$$b = \begin{array}{cccc} 1 & 1 & \ord_B & 1 \\ 4 & 3 & 7 & 5 \\ 3 & 8 & 5 & 2 \end{array} = C$$

The “Path\_B” algorithm (Algorithm 4.3) takes the reordered, extended, split filling $C = C_1 \ldots C_{2\lambda_1}$ given by Algorithm 4.2 and outputs a sequence of roots $\text{Path}\_B(C) = S \subseteq \Gamma$. We define $C_0$ to be the increasing column filled with 1, 2, \ldots, $n$. Note that the major difference between the Path\_B and Path\_C algorithms is the addition of blocked-off avoidance.

Algorithm 4.3. (“Path\_B”)  
for $i$ from 1 to $2\lambda_1$ do  
let $S_i := \emptyset$, $A := C_{i-1}$;  
for $(l, m)$ in $\Gamma_i$ do  
if $(l, m) = (i, i + 1)$ and $A, C_i$ are blocked off at $i$ by $C_i(i)$, then let $S_i := S_i, (i, i + 1)$, $A := A(i, i + 1)$;  
elsif $A(l) \neq C_i(l)$ and $A(l) < A(m) < C_i(l)$ and $A(l, m)$, $C_i$ not blocked off at $l$ by $C_i(l)$, then let $S_i := S_i, (l, m)$, $A := A(l, m)$;  
end if;  
end do;  
return $S := S_1 \ldots S_{2\lambda_1} = S_1^l S_1^r \ldots S_{\lambda_1}^l S_{\lambda_1}^r$.

Example 4.4. Consider the crystal $B^{(2,2)}$ of type $B_3$. Then $\lambda = (2, 2)$ and $\Gamma = \Gamma(2)\Gamma(2)$. Suppose that we have $\tilde{C}_1 = \begin{array}{cc} 3 \\ 2 \\ 1 \end{array}$ and $\tilde{C}_2 = \begin{array}{cc} 1 \\ 3 \\ 2 \end{array}$. Algorithm 4.3 produces the following subset of $\Gamma'(2) = \{(2, 3), (2, \overline{3}), (2, 3), (2, \overline{1}), (1, 3), (1, \overline{1}), (1, \overline{3})\}$:

$$\begin{array}{c|c|c|c|c|c} 3 & (2, 3) & 3 & (2, 3) & 2 & 1 \\ \hline 2 & 1 & 3 & 2 & 3 & 2 \\ \hline 1 & \overline{7} & \overline{1} & \overline{1} & \overline{2} \end{array}$$

Notice that Algorithm 3.2 would have called for the use of (1, 3) instead of (1, 3). This would have caused the resulting word to be blocked off with $\tilde{C}_2$ at 1 by 1 and thus the natural extension of algorithm Path\_C to type $B_n$ would not have terminated correctly.

Theorem 4.5. The map “path\_B \circ ord\_B \circ extend \circ split\_B” is the inverse of the type $B_n$ “sfill\_B” map.
The proof of Theorem 4.5 is based off of the following proposition. First we consider the following conditions on a pair of adjacent columns $CC'$ in a filling of a Young diagram.

**Conditions 4.6.** For any pair of indices $1 \leq i < l \leq k$,

1. $C(i) \neq C'(l)$
2. $C(i) \prec C'(l) \prec C'(i)$ only if the columns $C$ and $C't_{il}$ are blocked off at $i$ by $C'(l)$
3. $CC'$ are not blocked off at $i$ by $C'[i]$ for any $1 \leq i < k$

**Remark 4.7.** Let $b$ be a filling in $B_\lambda$, represented with split columns (as a filling of the shape $2\lambda$). Then Algorithm 4.12 constructs the unique filling $\sigma$ satisfying the following conditions: (i) the first column of $\sigma$ (of height $\lambda'_1$) is increasing, (ii) the adjacent columns of $\sigma$ satisfy Conditions 4.6, and (iii) $\text{sort}(\sigma) = b$.

**Proposition 4.8.** The restriction of the filling map "fill $B''$ to $A(\lambda)$ is injective. The image of this restriction consists of all fillings $C^1_1C^1_2 \cdots C^\lambda_iC^\lambda_j$ of the shape $2\lambda$ with integers in $[\pi]$ satisfying the following conditions: (i) $C^1_1$ is increasing; (ii) $(\text{ord}_{B}(C^1_1), \text{ord}_{B}(C^\lambda_i)) = (lD^1, rD^j)$ for some KN column $D^j$, for all $j$; (iii) any two adjacent columns satisfy Conditions 4.6.

The proof of Proposition 4.8 is based on the following two results, which will be proved in Sections 5 and 6 respectively.

**Proposition 4.9.** Consider a signed permutation $u$ in $B_n$, the column $C := u[1,k]$, and another column $CC'$ of height $k$. The pair of columns $CC'$ satisfies Conditions 4.6 if and only if there is a path $u = u_0, u_1, \ldots, u_p = v$ in the corresponding quantum Bruhat graph such that $v[1,k] = C'$ and the edge labels form a subsequence of $\Gamma_1(k)$. Moreover, the mentioned path is unique, and we have

$$C(i) = u_0(i) \preceq u_1(i) \preceq \ldots \preceq u_p(i) = C'(i), \text{ for } i = 1, \ldots, k.$$ 

To prove Proposition 4.9, we will first determine the necessary conditions for a path in the quantum Bruhat graph to exist and then construct a path using a more detailed version of Algorithm 4.3.

**Proposition 4.10.** Consider a signed permutation $u$ in $B_n$, the column $C := u[1,k]$, and another column $CC'$ of height $k$. The pair of columns $CC'$ satisfy Conditions 4.6 and $(\text{mod}_{\text{ord}}(C), \text{mod}_{\text{ord}}(C')) = (rD, lD)$ for some KN column $D$ if and only if there is a path $u = u_0, u_1, \ldots, u_p = v$ in the corresponding quantum Bruhat graph such that $v[1,k] = C'$ and the edge labels form a subsequence of $\Gamma_1(k)$. Moreover, the mentioned path is unique and, for each $i = 1, \ldots, k$, we have the following weakly increasing sequence:

$$C(i) = u_0(i) \preceq u_1(i) \preceq \ldots \preceq u_p(i) = C'(i).$$

To prove Proposition 4.10, we will first classify the split, extended and reordered KN columns, and then show that the resulting columns are exactly the necessary conditions for constructing a path in the quantum Bruhat graph.

**Proof.** (of Proposition 4.8) Consider a filling $\sigma = C^1_1C^1_2 \cdots C^\lambda_iC^\lambda_j$ satisfying conditions (i)-(iii), and let $u$ be the identity permutation. Apply Proposition 4.9 to $u$ and the column $C^1_1$; then set $u$ to the output signed permutation $v$, and apply Proposition 4.10 to $u$ and the column $C^1_1$. Continue in the way, by considering the columns $C^2_1, C^2_2, \ldots, C^\lambda_i, C^\lambda_j$, while each time setting $u$ to be the previously obtained $v$. This procedure constructs the unique folding pair $(w, T)$ mapped to $\sigma$ by the filling map $\text{fill } B$. Vice versa, a similar application of the two propositions shows that any filling in the image of $\text{fill } B$ satisfies conditions (i)-(iii).

**Proof.** (of Theorem 4.5) This is now immediate, based on Remark 4.7 and Proposition 4.8.
5. Proof of Proposition 4.9

5.1. Necessary conditions for reordered columns. Let $C$ and $C'$ be two columns of height $k$ with entries from $[\pi]$ where each column cannot contain both $i$ and $\pi$ for $1 \leq i \leq n$. Recall Algorithm 4.2 (the type $B_n$ reorder algorithm) and note that for each such pair of columns, it reorders the second based on the first in a way that is consistent with Conditions 4.6. In this section, we show that the first two parts of Conditions 4.6 are necessary for the existence of a path between the two columns in the quantum Bruhat graph. The necessity for the third condition will be realized in Section 5.3.

We begin by recalling Conditions 4.6 on adjacent columns $CC'$ where we now write the second condition in an equivalent way which will be more beneficial in this section.

For any pair of indices $1 \leq i < l \leq k$,

1. $C(i) \neq C'(l)$
2. the statement $C(i) \prec C'(l) < C'(i)$ is false unless the following are true.
   
   (a) $\{1, 2, \ldots, |C'(l)|\} \subseteq \{C(j)\}_{1 \leq j \leq i}$ and $\{1, 2, \ldots, |C'(l)|-1\} \subseteq \{C'(j)\}_{1 \leq j \leq i-1}$
   
   (b) $C'(l) > 0$ and $|C(i)| \leq C'(l)$ with equality iff $C(i) = C'(l)$,
   
   (c) and $p'$ is odd, where $p = \#\{j : 1 \leq j < i-1, C(j) < 0, C'(j) > 0\}$ and $p' = p + 1$ if $C(i) < 0$ and $p' = p$ otherwise.
3. $C, C'$ are not blocked off at $i$ by $C'[i]$ for any $1 \leq i < k$

The following lemma will show that for a given path in the quantum Bruhat graph, the only possible transposition in $T_i$ that can pass the target value $C'(i)$ is $(k, k+1)$. It also gives us the monotonicity of the values in position $i$ during the application of transpositions of $T_i$.

Lemma 5.1. For $T_i = (i, j_1)(i, j_2) \ldots (i, j_q)$, some $1 \leq i \leq k$, let $u_0 := \pi_{i+1}$ and $u_r := \pi_{i+1}t_{ij_1}t_{ij_2} \ldots t_{ij_r}$ for $1 \leq r \leq q$. Then $u_r(i) \prec u_{r+1}(i) \preceq C'(i)$ for $r = 1, \ldots, q-1$, while for $r = 0$ this fails only if $(i, j_1) = (k, k+1)$.

Proof. Suppose that there is some $0 \leq r < q - 1$ such that $u_r(i) \prec C'(i) \prec u_{r+1}(i)$. We first show that this cannot happen during the application of transpositions in $T_i$ for $i < k$ and then we show that it can only happen for $j_1 = k + 1$ in $T_k$.

Let $i < k$. Then if we have $u_r(i) \prec C'(i) \prec u_{r+1}(i)$, the QBG criterion then gives that

$$u_r(i) \prec C'(i) \prec u_{r+1}(i) \preceq u_r(k) = C'(k) < u_r(i).$$

Therefore some transposition $(i, j_r)$, for $r + 1 < r' \leq q$ will transpose over the value $C'(k)$, breaking the QBG criterion.

Now let $i = k$. Consider the case where $j_{r+1} \neq k + 1$. We will show that $u_r(i) \prec C'(i) \preceq u_{r+1}(i)$ cannot occur and the result follows. We break into two cases: when $j_{r+1} \neq k + 1$ and when $j_{r+1} = k + 1$.

Case 1: consider when $j_{r+1} \neq k + 1$. Then by the QBG criterion, we have

$$u_r(k) \prec C'(k) \prec u_{r+1}(k) \preceq u_r(k+1) < u_r(k).$$

However, the only way to transpose over the value in position $k + 1$ is with the root $(k, k + 1)$, after which there is no longer a root which can transpose over the root in position $j_r$.

Case 2: consider when $j_{r+1} = k + 1$. Then there is still the conflict of transposing over the value in position $j_r$. Indeed, this could only be done with a Stage III transposition, of which $(k, k + 1)$ is the last.

Therefore the only possible transposition where passing the target value $C'(i)$ can possibly occur is with the root $(k, k + 1)$. 

\[ \square \]
The next lemma shows us that if we have two columns $CC'$ satisfying Condition 2, we know exactly where in the application of roots in $T$ the current word’s value in position $i$ becomes greater than $C'(l)$ in the circle order starting at $C(i)$.

**Lemma 5.2.** Let $CC'$ be a pair of columns such that $a := C(i) < b := C'(l) < c := C'(i)$ for some $1 \leq i < l \leq k$. Then the reflection $(l, \pi)$ is in $T_i$. Furthermore, if $w$ is the word immediately prior to the application of $(l, \pi)$, then we have that $w(i) < C'(l) < C'(i)$ and $C'(l) < wt_{l, \pi}(i) < C'(i)$.

**Proof.** Let us assume that we have $a := C(i) < b := C'(l) < c := C'(i)$ for some $1 \leq i < l \leq k$. We first show that there is a root with the said properties and then show that this root is exactly the root $(l, \pi)$.

First, note that we cannot have $C_l(i) \leq b < c$, otherwise the entry in position $i$ of the signed permutation would then change from $C_l(i)$ to $C'(i)$ via reflections in $T_{l-1}T_{l-2} \ldots T_1$. One of these reflections would then transpose entries across $b$, violating the quantum Bruhat graph criterion. It must then be that $b < C_l(i) \leq c$. This means that at some point in the process of applying to $u$ the reflections in $T_k, T_{k-1}, \ldots, T_1$, we apply to the current signed permutation $w$ a reflection $(i, \overline{m})$ such that $a < a' := w(i) < b < c' := w(\overline{m}) < c$. Let $(i_0, \overline{m}), \ldots, (i_p, \overline{m})$ be the segment of $T_m$ starting with $(i, \overline{m})$ and consisting of roots $(-, \overline{m})$, where $i = i_0 > i_1 > \ldots > i_p \geq 1$. Let $a_r := w(i_r)$ for $0 \leq r \leq p$.

We now show that $(i, \overline{m})$ is our desired root, or equivalently, that $m = l$. Since $wt_{i, \overline{m}}t_{i_1, \overline{m}} \ldots t_{i_p, \overline{m}} = \pi_m$, the quantum Bruhat criterion gives that $\pi_m[i_p, \overline{m}]$ cannot have any entries between $a_p$ and $c'$ except maybe $\overline{a_p}$ in the $m$th position of $C_m$. It now follows that since $i_p \leq i < l \leq m < \overline{m}$, we cannot have $a_p < C_m(l) < c'$ unless $m = l$ where $C_m(m) = \overline{a_p}$. Suppose that $m \neq l$. Then $C_m(l) < a_p < c'$. Note that by Lemma 5.1 $\overline{a} < \overline{a'} = \overline{a_0} < \overline{a_1} < \ldots < \overline{a_p}$. This gives us that $a_p < a' < c'$ and so $C_m(l) < a_p < a' < b < c'$. Since $C'(l) = b$, there would then have to be some transposition in $T_{m-1}, \ldots, T_1$ which would transpose two values over $a_p$ in position $\overline{m}$. This contradicts the quantum Bruhat criterion, and so it must have been that $\overline{a_p} = b$ and $m = l$. \hfill $\Box$

**Example 5.1.** Consider the following quantum Bruhat graph path in $B_8$. Here we have $u[1,6]$ and $v[1,6]$ follow the proposed reorder criterion. Note that $u[4] < v[6] < v[4]$ and that $u[1,6], vt_{4,6}[1,6]$ is blocked off at 4 by 3. We see not only that $t_{6,7}$ is indeed in the path, but also that while $w[4] < v[6] < v[4]$ the next transposition gives $wt_{6,7}[4] < v[4] < v[6]$.

![Graph Path](image)

**Remark 5.3.** Let $CC'$ be a pair of columns such that $a := C(i) < b := C'(l) < c := C'(i)$ for some $1 \leq i < l \leq k$. We claim that $\{1,2,\ldots,|b| - 1\} \subseteq \{|C_m[j]|\}_{1 \leq j \leq i}$ and $\{1,2,\ldots,|b| - 1\} \subseteq \{|C_{m+1}[j]|\}_{1 \leq j \leq i}$. For the first, recall from the proof of Lemma 5.2 that there cannot be any values between $b$ and $c'$ in $\pi_m[i, \overline{m}]$. The same proof gave way to inequalities which tell us that $|b| < |c'|$. This means that there cannot be any values between $b$ and $c'$ in $\pi_m[i, \overline{m}]$. Since $i < m$, we have that there cannot be any values between $\overline{b}$ and $\overline{b}$ in $\pi_m[i, \overline{m}]$. The first claim then follows. For the second, notice that the only difference between the content of $C_{m+1}[1, i]$ and $C_m[i, i - 1]$ is the loss of $a_p = \overline{b}$.
The following Lemma will be used for showing the necessity of Condition 2c.

**Lemma 5.4.** Suppose that there is \( 1 \leq i < l \leq k \) such that \( C(i) < C'(l) < C'(i) \), and Conditions 2a,b are met. Then for all \( 1 \leq j \leq i - 1 \), if \( C(j), C'(j) > 0 \), then \( C(j) < C'(j) \).

**Proof.** Suppose that for such a \( j \), there is an \( r \in \{1, \ldots, q\} \) such that \( u_r(j) < 0 \). Recall that by the quantum Bruhat criterion, all upsteps in stages III and IV must maintain the same sign. This means that the transposition acting on the current word \( w \) changing position \( j \) from positive to negative must come from a stage I or II reflection which can only happen in \( T_j \) at \( (j, s_0) \) for some \( s_0 \in \{k + 1, \ldots, k + 1\} \cup \{j\} \). Since Condition 2a holds, \( wt_{j,s_0}(j) \notin \{\tilde{b}\} \). Let the remaining stage I reflections be \( (j, s_1), \ldots, (j, s_p) \) where \( k + 1 \leq s_0 < \cdots < s_p \leq n \). Notice that \( w[s_i] \notin \{\tilde{b}\} \) for any \( 0 \leq i \leq p \) by Condition 2a. This means that \( b < w'(j) = wt_{j,s_0} \cdots t_{j,s_p}[j] < \tilde{b} \). However, if \( w'[j] \in \{b, \ldots, n\} \), then one of the reflections \( (j, s_i) \) would have to transpose two values over \( b \) in position \( l \), contradicting the quantum Bruhat criterion. Thus \( w'(j) \in \{\pi, \ldots, b + 1\} \). Note that we cannot apply the \((j, \tilde{j})\) reflection at this time by quantum bruhat criterion, so we move on to stage III reflections. Again by Condition 2b, each stage III reflection can only change position \( j \) to a value in \( \{b + 1, \ldots, \tilde{b} + 1\} \). However, no stage III reflection can take position \( j \) to anything in \( \{b + 1, \ldots, w'[j]\} \) without contradicting either the quantum Bruhat criterion or Lemma 5.1. Thus if \( w'' \) is the word at the end of applying stage III transpositions, we have that \( w''(j) \in \{\pi, \ldots, b + 1\} \). This means that there must be a stage IV reflection taking \( w''(j) \) to a positive value. This cannot happen, as it would require a reflection to transpose over \( \tilde{b} \) in position \( l \), contradicting the quantum Bruhat graph criterion. Thus there is no such \( u_r(j) < 0 \) and the lemma holds.

**Remark 5.5.** By Lemma 5.1, it is clear that for each \( 1 \leq j < k \), there may only be at most one downstep during \( T_j \). Otherwise we would inevitably pass over the target.

The following Lemma was given by Lenart in [18] and is not type dependant so we will use it freely.

**Lemma 5.6.** Fix \( i \) and \( l \) with \( 1 \leq i < l \leq k \). Let \( a \) be the entry in position \( i \) of the signed permutation obtained at some point in the process of applying to \( u \) the reflections in \( T_k, T_{k-1}, \ldots, T_{l+1} \). Then either \( a \) appears in \( C_{l+1}[1, i] \) or \( a \) appears in \( C_{l+1}[l + 1, k] \).

**Proposition 5.7.** The pair of columns \( CC' \) satisfy Conditions 1 and 2.

**Proof.** The proof of condition 1 is the same as in type \( C_n \). Let us now assume that we have \( a := C(i) < b := C'(l) < c := C'(i) \) for some \( 1 \leq i < l \leq k \). Let \( a', c', m, \) and \( w \) be as in Lemma 5.2.

We first show that \( \{1, \ldots, b - 1\} \subseteq \{|C'(j)|\}_{1 \leq j \leq i} \). It is equivalent to show that values in \( \{\tilde{b} - 1\} \) are not in \( \pi_1[i + 1, n] \). This is true for positions \( m \) though \( k \), as \( \pi_1[m, k] = \pi_m[m, k] \) which was shown in Remark 5.3 to not contain elements of \( \{\tilde{b} - 1\} \). Suppose that \( |\pi_1(r)| \in \{1, \ldots, b - 1\} \) for some \( r \in \{i, \ldots, m - 1\} \). Then (from the proof of Lemma 5.2) \( \pi_m(r) \) is between \( c' \) and \( \tilde{b} \) and \( C'(r) = \pi_1(r) \) is in either \( \pi_m[1, i - 1] \) or \( \pi_m[i - 1, i] \) and by Lemma 5.6, necessarily the latter. But during the reflections in \( T_1, \ldots, T_m \) there would have to be a reflection that transposes over \( \tilde{b} \) in position \( \pi_{i'} \), contradicting the quantum Bruhat criterion. Now suppose that \( |\pi_1(r)| \in \{\tilde{b} - 1\} \) for some \( r \in \{k, \ldots, n, \pi, \ldots, k + 1\} \). But this would mean that some reflection in \( T_{m-1}, \ldots, T_1 \) would have to transpose an entry over \( b \) in position \( m \), contradicting the the quantum Bruhat criterion. Thus there are no elements of \( \{\tilde{b} - 1\} \) in \( \pi_1[i + 1, \tilde{i} + 1] \) and so \( \{1, \ldots, b - 1\} \subseteq \{|C'(j)|\}_{1 \leq j \leq i} \) as desired.

We now show that \( \{1, \ldots, b\} \subseteq \{|C[j]|\}_{1 \leq j \leq i} \). If not, then, given what we know about the structure of \( \pi_{m+1} \) from Remark 5.3, there must be \( t \in \{m + 1, \ldots, k\} \) and \( \tilde{t}' \in \{1, \ldots, i\} \) where
the application of the reflection $(i'_1, \overline{t})$ to the current word $w'$ transposes value $\gamma = w'[\overline{t}]$ and $w'[i'_1]$ where $\overline{b} \leq \gamma \leq b$. Let $(i'_1, \overline{t}), \ldots, (i'_r, \overline{t})$ be the remainder of $T_t$ starting with $(i'_1, \overline{t})$ and consisting of roots $(\cdot, \overline{t})$, where $i'_1 > i'_2 > \ldots > i'_r \geq 1$.

We claim that $\mu := w't_{i'_1, \overline{t}} \ldots t_{i'_r, \overline{t}}[\overline{t}] = w't_{i'_1, \overline{t}} \ldots t_{i'_r, \overline{t}}[t] = C_t[t] = C'(t) \in \{c', \ldots, \overline{c}\}$, $c' > 0$, and further that no values in $w'[\overline{t}]$ can lie between $\mu$ and $\overline{\gamma}$. This means that $w'[m]$ lies between $\overline{\gamma}$ and $\mu$. But, from the construction in the proof in Lemma 5.2, we had that $C_m[m] = \overline{c'}$, which would be impossible, as we would then have to traverse over $\overline{\mu}$ in position $t$ during one of the reflections in $T_{t-1}, \ldots, T_{m+1}$, breaking the quantum Bruhat criterion. Therefore there was no such $\gamma$ and it must be that $\{1, \ldots, b\} \subseteq \{C[j]\}_{1 \leq j \leq i}$. We finish this part by proving the two claims. First, that $\overline{\mu} := w't_{i'_1, \overline{t}} \ldots t_{i'_r, \overline{t}}[\overline{\gamma}] = w't_{i'_1, \overline{t}} \ldots t_{i'_r, \overline{t}}[t] = C_t[t] = C'[t] \in \{c', \ldots, c\}$. Recall from construction in the proof of Lemma 5.2 that the reflection $(m, \overline{t})$ will be used to transpose $a'$ and $c'$ in $w$ during the application of reflections in $T_m$ and so there are no values between $a'$ and $c'$ in $w[i, m]$ and nothing between $c'$ and $\overline{a}$ in $w[m, \overline{t}]$. This means that all values in $w[m, \overline{t}]$ lie in $\{\overline{a'}, \ldots, a'\} \cup \{c', \ldots, \overline{c}\}$. Notice that $w[m+1, m+1] = \pi_m[m+1, m+1] = \pi_1[1, m+1]$ and since \( \overline{b} \leq a' < b \), we have that $a' \in [\overline{b}]$, so by the condition set on the position of the values on $\pi_m$ in Remark 5.3 we get that values in $w[m+1, m+1]$ must come from $\{c', \ldots, \overline{c}\}$. Note that this forces $c' > 0$. The claim follows, as $t \in \{m+1, \ldots, k\}$.

For the second part of the claim, notice that by the quantum Bruhat criterion, no values in $w'[i'_1, \overline{t}]$ can lie between $w'[\overline{t}]$ and $w'[\overline{t}] = \overline{\gamma}$. Similarly, no values in $w't_{i'_1, \overline{t}}[i'_2, \overline{t}]$ can lie between $w'[i'_2]$ and $w't_{i'_1, \overline{t}}[\overline{\gamma}] = w'[i'_1]$, so no values in $w'[\overline{t}]$ can lie between $w'[i'_2]$ and $\overline{\gamma}$. Continue the same way with the rest of the $(t, \cdot)$ transpositions and we see that no values in $w't_{i'_1, \overline{t}} \ldots t_{i'_r, \overline{t}}[\overline{t}]$ can lie between $w'[i'_r] = \mu \in \{c', \ldots, c\}$ and $w'[\overline{t}]$, so no values in $w'[\overline{t}]$ can lie between $\mu$ and $\overline{\gamma}$.

We now wish to show that the existence of $a := C(i) < b := C'(l) < c := C'(i)$ implies Condition 2b. First, suppose that $b < 0$. Recall that $\overline{b} < a' < b < c$. We have already shown that $c' > 0$ and so $c' < a'$ and $\overline{c'} > \overline{a'}$. Then the reflection $(m, \overline{t})$ which transposes $\overline{c'}$ with $\overline{a'}$ is a downstep and, by the quantum Bruhat criterion, must be that $a'$ and $c'$ are of different sign. This means that $a' < 0$. But then via Lemma 5.2 the reflections $(m, \overline{t})$ and $(m, \overline{t})$ neither $a'$ nor $\overline{a'}$ can appear in the latter, and thus one of them must appear in the prior. Since $j < i$ the transposition $(m, \overline{t})$ occurs before the transposition $(m, \overline{t})$, the transposition $(m, \overline{t})$ cannot transpose the values $\overline{t}$ with $a'$ as it had originally been constructed, a contradiction. Thus $a = a'$ and Condition 2b holds.

We now wish to show that the existence of $a := C(i) < b := C'(l) < c := C'(i)$ implies Condition 2c. We first count the number of downsteps made with transpositions with values in positions $[1, i]$.

First we look at the number of downsteps in $T_k, \ldots, T_{t+1}$. Here, the downsteps including positions $1, \ldots, i - 1$ are only those of stage IV. By our construction thus far, we know that this transposes a negative value in $\{b+1, \ldots, b+1\}$ with a positive value. This then transposes over $\overline{t}$ in position $\overline{t}$, so by the quantum bruhat criterion, we know that no such downstep exists.
Second, we look at the number of downsteps in \( T_i \). By Condition 2b, \( C[i] \in \{b + 1, \ldots, b + 1\} \), and by hypothesis, \( C'[i] = b \). Thus there must be at least one downstep during \( T_i \). From Remark 5.3 we have that \( \{1, \ldots, b\} \) are in \( \pi_l[1, i] \) or \( \pi_l[\tilde{l}, \tilde{I}] \). By Lemma 5.2 the downstep must be a stage IV move with a value in some position in \( \{1, \ldots, i\} \). By Remark 5.5, there is at most one downstep in \( T_i \), so there is exactly one downstep in \( T_i \). We note here that the value in position \( i \) will never become negative again, as the only means of doing so would be with the root \((i, \tilde{i})\) in \( T_i \), but this would pass the target \( b > 0 \).

Third, we look at the downsteps in \( T_{i-1}, \ldots, T_i \). By similar argumentation to that of \( T_k \ldots T_{l+1} \), we see that any possible downstep with values in \( 1, \ldots, i \) would pass over \( b \) in position \( l \), breaking the quantum Bruhat criterion.

Finally, we look at the downsteps in \( T_{i-1}, \ldots, T_1 \). Notice that there are no downsteps with stages I,III without passing over \( b \) in position \( l \). There are no stage II downsteps at all in type \( B_n \). So all downsteps are of stage IV. Each of these change two positions in \( 1, \ldots, i \) from negative to positive. We note here that none of these positions can become negative again. Indeed the only way for this to happen would be with a stage II upstep, but this root has already been passed to use the stage IV downstep.

We have seen that there are an even number of downsteps in positions \( 1, i \) contributed by \( T_i, \ldots, T_1 \), one from \( T_i \), and none from anywhere else. We noted along the way that none of these values will become negative again during the remainder of the transpositions. This along with Lemma 5.4 and Remark 5.5 gives that there is an odd number of such negative to positive pairs in the said positions of \( C \) and \( C' \), as desired.

\[\square\]

5.2. Necessary conditions for the construction of a segment of the quantum Bruhat path. The following proposition shows a partial result for Proposition 4.9.

**Proposition 5.8.** Let \( u, i, C, C' \) be as previously defined and assume that the pair of columns \( CC' \) satisfies conditions 1 and 2. Assume also that \( C[i + 1, k] = C'[i + 1, k] \) for some \( i \) with \( 1 \leq i \leq k \). Then there is a unique path \( u = u_0, u_1, \ldots, u_q = v \) in the corresponding quantum Bruhat graph such that \( v(i) = C'(i) \) and the edge labels form a subsequence of \( \Gamma_{ki} \). Moreover, we have that if \( u(l) \neq v(l) \), then \( C(l) = u(l) < v(l) \leq C'(l) \) for \( l = 1, \ldots, i - 1 \).

In this section, we will provide necessary conditions for the results of Proposition 5.8. Assume for the moment that a path with the property stated in Proposition 5.8 exists. Let \( T \) be the sequence of edge labels for this path. Recall that the sequence of roots \( \Gamma_{ki} \) were split into four stages. We will factorize accordingly, giving \( T = T_1T_2T_3T_4 \) and define \( u_i := uT_i, u_{ii} := uT_1T_ii, u_{iii} := uT_1T_2T_3 \), and \( u_{iiv} := uT = v \).

The following lemmas give necessary conditions for the construction in the Proposition. They show that we will pass the target (cf. Remark 4.3) only with the root \((k, k + 1)\) exactly when the reordered columns \( CC' \) are blocked off at \( k \) and that we will skip a \( Path_C \) transposition exactly when the \( Path_C \) transposition would lead the resulting column to be blocked off with \( C' \). We note here that these necessary conditions give the uniqueness of the path proposed in Proposition 5.8.

Due to the third part of the block off condition, we will often need to discuss the signs of values in the same row of two columns. For two columns \( L \) and \( R \), we will denote the signs of position \( i \) of these columns by \( sgn(l_i)sgn(r_i) \).

**Example 5.2.** Consider the columns \( C = (1, 4, 2, 3, 8, 5) \) and \( C' = (1, 5, 2, 6, 8, 3) \). We then say that the values of \( CC' \) in position 4 are \( -+ \). We will commonly discuss how these sign pairs change...
We note that we can restate part 3 from Definition 4.4 as **there are an odd number of **++** pairs in positions 1 through i in the columnns CC′.**

**Lemma 5.9.** Suppose that \( u[i+1,k] = C'[i+1,k] \) and \( u,C' \) are blocked off at \( i \) by \( b := C'(i) \). Then, if we never pass the target, there is no \( T \subset \Gamma_{ki} \) such that \( uT[1,k] = C' \).

**Proof.** Let \( u \) and \( C' \) be as hypothesized. By the block off condition, either \( u(i) = \overline{b} \) or \( |u(i)| < b \). We first consider the case where \( u(i) = \overline{b} \). Since our target value is in position \( \overline{t} \), the only root available to us are those in stage I and stage II. However, by the block off condition, there is no \( k < l \leq n \) such that \( \overline{b} < u(l) < b \) and the root \((i,\overline{t})\) does not follow the quantum Bruhat graph criterion, as \( \overline{b} < 0 \). Since we do not pass the target, there is no path.

We now consider the case where \( |u(i)| < b \). First note that there are an odd number of **−−** pairs in \( u[1,i], C'[1,i] \) due to the block off condition. We claim that if there is some subchain \( T_i \subset \Gamma_{ki} \) such that \( uT_i(i) = \overline{b} \), then there are still an odd number of **−−** pairs between \( uT_i[1,i−1] \) and \( C'[1,i−1] \). Indeed, if \( 1 \leq u(i) < b \), then by Lemma 5.1 only upsteps will be used in \( T_i \) and so \( sgn(u(l)) = sgn(uT_i(l)) \) for all \( 1 \leq l < i \). We also consider the case \( \overline{b} < u(i) \leq T \). Here, there must be a downstep at some point in \( T_i \). It can not be in stage I or III, otherwise we would pass the target due to the block off condition. There are no stage II downsteps by quantum Bruhat criterion. Thus the downstep must be in stage IV. This single downstep will not only change the values in position \( i \) from **−−** to **++**, but also a **−−** to a **++** in some position \( 1 \leq l < i \). Therefore, regardless of the sign of \( u(i) \), the number of **−−** values in \( uT_i[1,i−1] \) and \( C'[1,i−1] \) will be odd.

The conclusion of the proof is done by showing that if there is a path \( T \subset rev(\Gamma) \) such that \( uT_iT = C' \), there must have been an even number of **−−** pairs between \( uT_i[1,i−1] \) and \( C'[1,i−1] \). We will do so by first showing that no **++** pairs will give way to **−−** at any time during \( T \) and further that all **−−** pairs are only changed to **++** pairs via stage IV downsteps, each of which turn two **−−** pairs to **++**. For the remainder of the proof, let \( u' = uT_i \) and note that \( u'[i,k] = C'[i,k] \) and \( u'(i) = b = C'(i) \).

We first consider some \( 1 \leq l < i \) where \( u'(l) > 0 \) and \( C'(l) > 0 \) and will show that position \( l \) will never be negative throughout the application of the remainder of the roots in \( T \). First note that by the quantum Bruhat criterion, we have that position \( l \) remains positive throughout \( T_{i−1}T_{i−2} \ldots T_{i+l}(l) \). Let \( s := u'T_{i−1}T_{i−2} \ldots T_{i+l}(l) \). Then we have the following two cases: either \( 1 \leq s \leq C'(l) \) or \( C'(l) < s \leq n \). For the first case, we are done via Lemma 5.1. Now suppose that \( C'(l) < s \leq n \). During stages I, II, and III of \( T_i \), we can only transpose with values between \( b \) and \( \overline{b} \), due to the nature of the block off condition. This means that at some point during stage IV of \( T_i \) two values would have to transpose over \( \overline{b} \) in position \( \overline{t} \), breaking the quantum Bruhat criterion. Therefore only the first case holds and no **++** pair between \( u'[1,i−1] \) and \( C'[1,i−1] \) will ever give way to a **−−** pair during the remainder of \( T \).

Finally, we show that all **−−** pairs in \( u'[1,i−1] \) and \( C'[1,i−1] \) can only become **++** pairs via stage IV downsteps, effectively changing two such **−−** pairs at a time. We now consider some \( 1 \leq l < i \) such that \( u'(l) < 0 \) and \( C'(l) > 0 \) and again let \( s = u'T_{i−1}T_{i−2} \ldots T_{i+l}(l) \). Suppose that during \( T_{i−1}T_{i−2} \ldots T_{i+l}(l) \) there are no stage IV down steps using position \( l \). We then have that \( s \) is negative. Further, we note that \( 1 \leq C'(l) < b \), otherwise there would come a point where

| \( C \) | \( C' \) | \( Ct_{67} \) | \( C' \) |
|---|---|---|---|
| 1 1 | 1 1 | 1 1 |
| 4 5 | 4 5 | 4 5 |
| 2 2 | 2 2 | 2 2 |
| 3 6 | 5 6 | 5 6 |
| 8 8 | 8 8 | 8 8 |
| 5 3 | 3 3 | 3 3 |
Lemma 5.9 tells us that there may be times when the two columns can never be blocked off with each other except for possibly at $C'$. Further, Lemma 5.9 tells us that there may be times when the Path$_C$ algorithm would call for a certain transposition, but it can not be included in the path, as its use would cause the current column to be blocked off with $C'$. The following lemmas will show that block-off avoidance is the only time a path is allowed to skip a root called by the Path$_C$ algorithm or pass the target with the root $(k, k + 1)$.

Lemma 5.11. Let the sequence $m_1, m_2, \ldots, m_r$ be such that $u_{i,m_1, t_{i,m_2}} \ldots t_{i,m_r}(i) = C'(i)$ and the roots $(i, m_i)$ form a path in the corresponding quantum Bruhat graph. Suppose that there is some $m_d \in \{k, \ldots, \hat{i}, \ldots, T\}$, such that at the current word $w$, we have $w(i) < ut_{i,m_d}(i) \leq C'(i)$, but $m_d$ is not in $\{m_1, m_2, \ldots, m_r\}$. Then $ut_{i,m_d}$ was blocked off at $i$ by $C'(i)$.

Proof. In the hypothesised construction, at some point in the process of applying the transpositions $(i, m_j)$ to $u$, the entry in position $i$ has to change from a value $a$ to a value $b$ across the value $u(m_d)$ where $a < u(m_d) < b$. This violates the quantum Bruhat graph criterion in all but the following two special cases:

Case 1: $a < 0$ and $b > 0$, the current transposition $(i, m_j)$ is in $T_{iii}$, and $u(m_d) = \overline{b}$. 

Case 2: $a < 0$ and $b > 0$, the current transposition $(i, m_j)$ is in $T_{iv}$, and $u(m_d) = \overline{a}$.

We show that the lemma holds for case one and case two follows similarly. Let $w' := ut_{i,m_1} \ldots t_{i,m_{j-1}}$. Then $w't_{i,m_j}(i) = b$. We need $b = C'(i) > 0$. Note that after stage II, the quantum Bruhat criterion no longer allows any positive to negative sign changes. Since $b > 0$, it must be that the target value $t = C'(i)$ is positive as well. Lemma 5.9 then gives that $1 \leq b \leq t \leq n$, providing us with the desired $|u(m_d)| \leq t$ as $u(m_d) = \overline{b}$ by hypothesis.

We now show that $\{1, \ldots, t-1\} \subset \{|w(j)|\}_{1 \leq j \leq i}$ and $\{1, \ldots, t\} \subset \{|C'(j)|\}_{1 \leq j \leq i}$. First note that we cannot have $b < \overline{a} < t$. Otherwise there would have to be root later in the sequence transposing two values $c$ and $d$ over $\overline{a}$ in position $\overline{m_j}$. Again, this is only possible if $d = \overline{a} = a$, but that would mean that the transposition would pass the target. Lemma 5.9 gave that this is only possible for the root $(k, k + 1)$, but $m_j$ is in $T_{iii}$ and is therefore not $k + 1$. So we now have $\overline{a} \leq a \leq \overline{b} \leq \overline{t} < 1 \leq b \leq t \leq \overline{a} \leq n$.

By the quantum Bruhat criterion, there are no values between $a$ and $b$ in $w'[i, m_j]$ except $\overline{b}$ in position $\overline{m_j} = m_d$. Suppose that some value between $b$ and $t$ lies in $w'[i, m_j]$. Then for $w'(i)$ to have the value $a$, either we have $i = k$ and we passed the target during $(k, k + 1)$, or we skipped a different Path$_C$ step as well. The latter cannot be, as it would then require another downstep, of which there can only be one for each $T_i$. The prior cannot be true either, as the passing of the target would lead to the need for an additional downstep as well.

Thus no values between $a$ and $t$ can lie in $w'[i, m_j]$ other than $b$ and $\overline{b}$. This means that no values in $[\overline{t} - \{b, \overline{b}\}]$ can lie in $w'[i, m_j]$. Since the root $(i, m_j)$ is in $T_{iii}$, we know that $w'[1, i-1] = w'[1, i-1]$. Thus $\{1, \ldots, t-1\} \setminus \{b\} \subset \{|w(j)|\}_{1 \leq j \leq i}$. We further note that during $T_i, T_{i-1}, \ldots, T_1$ none of the values in $[\overline{b}]$ can be transposed out of positions 1 through $i$ without passing over $t$ in position $i$. Thus $\{1, \ldots, t\} \subset \{|C'(j)|\}_{1 \leq j \leq i}$ as desired.

We now show that $wt_{i,m_j}[1, i]$ and $C'[1, i]$ have an odd number of $-+$ pairs. Since we have already shown that conditions $2a$ and $2b$ hold and $1 \leq b \leq t$, there must be an even number of minus plus pairs between $w't_{i,m_j}$ and $C'$ in the first $i$ positions. Otherwise, the application of the
root \((i, m_j)\) would cause \(w't_i,m_j[1,k]\) to be blocked off with \(C'\) at \(i\) by \(t\). This then means that there would have been an odd number of minus plus pairs between \(w't_i,m_j\) and \(C'\) in the first \(i\) positions. Thus the use of the root \((i, m_d)\) would have caused the columns \(w't_i,m_d[1,k]\) and \(C'\) to be blocked off at \(i\) by \(t\).

\[\square\]

**Remark 5.12.** The proof of Lemma 5.11 also gives us that a skipped \(Path.C\) transposition may only occur during stage I.

**Lemma 5.13.** Suppose that we have a path such that the root \((k, k+1)\) is used and its application to the word \(u\) passes the target in position \(k\). Then the columns \(u[1,k]\) and \(C'\) are blocked off at \(k\) by \(C'(k)\).

**Proof.** Note that this set up is equivalent to having some \(u' = u(k, k+1)\) and skipping the root \((k, k+1)\) which would now be called by the \(Path.C\) algorithm. Then by Lemma 5.11 we know that \(u'(k, k+1)[1,k]\) is blocked off with \(C'\) at \(k\) by \(C'(k)\). The proof concludes with the realization that \(u'(k, k+1) = u\). \[\square\]

**Remark 5.14.** The Lemmas 5.1, 5.9, 5.11 and 5.13 give conditions that dictate a unique path.

### 5.3. Constructing a segment of the quantum Bruhat path.

In this section we will provide an explicit algorithm for the unique path following the coditions set by Lemmas 5.1, 5.9, 5.11 and 5.13. Further, we will show that this path is the desired path for Proposition 5.8.

First we distinguish the following two cases:

1. \(C(i) \preceq C'(i) \prec C'(i)\)
2. \(C(i) \preceq C'(i) \prec C'(i)\).

We will also need the following notation:

1. \(M_I(u, i, C') := \max\{\{u(i)\} \cup \{u(l) : k + 1 \leq l \leq n, u(i) \prec u(l) \preceq C'(i)\}\}
2. \(M_{III}(u, i, C') := \max\{\{\pm u(i)\} \cup \{u(l) : k + 1 \leq l \leq k + 1, u(i) \prec u(l) \preceq C'(i)\}\}

The following lemma gives a little insight into the nature of \(M_I\). Its proof mirrors a similar lemma given by Lenart in [18] but cators to type \(B_n\).

**Lemma 5.15.** Under the hypothesis of Proposition 5.8 in Case 2 we have

\[C'(i) \preceq M_I(u, i, C') \preceq C'(i)\]

**Proof.** Without loss of generality, assume \(C'(i) > 0\) and that \(u(i) \neq C'(i)\). Let \(a = C'(i) \in [n]\) and \(A = \{\pi, a - 1, \ldots, 1, 1, \ldots, a - 1, a\}\). We need to show that \(u[k + 1, n]\) contains no elements in \(A\), so assume the contrary. Suppose that \(u[i + 1, k] = C'[i + 1, k]\) contains an element of \(A\). Then by Conditions 1 and 2, \(u(i) \in A\) and thus \(C'(i) \preceq u(i) \preceq M_I(u, i, C') \preceq C'(i)\). Now let \(u[i + 1, k] = C'[i + 1, k]\) not contain any element of \(A\). We conclude that \(u[1, i - 1]\) contains an element from each pair \(\{x, \pi\}\) of elements in \(A\). If \(u(i') \in A\) for \(i' < i\), we say that \(u(i')\) is matched with \(C'(i')\). Since \(C'(i) = a\), the only possible matches for \(u[1, i - 1] \cap A\) are elements in \(A \setminus \{a, \pi\}\), by the first part of the reorder condition. But these are too few to match \(a\) elements, which is a contradiction. \[\square\]

**Remark 5.16.** Since \(u(i) \preceq M_I \preceq M_{III} \preceq C'(i)\), we then have that \(C'(i) \preceq M_{III}(u, i, C') \preceq C'(i)\).

We now describe the algorithm that constructs the path in Proposition 5.8. The algorithm inputs the signed permutation \(u\), the target column \(C'\), and the position \(i\); it outputs the list of reflections
T determining the path and the permutation \( v \). It calls on the algorithm \textit{is-blocked-off} which inputs the first \( i \) entries of the permutation \( u \) and column \( C' \) as columns and the value \( i \); it then returns whether or not the given columns are blocked off at \( i \) by \( C'(i) \). Note that the following procedure is one iteration of Algorithm 4.3 where we explicitly go through the four stages of \( \Gamma^{ki} \).

\begin{verbatim}
procedure path-B(u,i,C');
    Let c := C'(i);
    if u(i) = c then return \( \emptyset, u \);
else
        Let S := \( \emptyset \), L := (k + 1, \ldots, n) v : u;
        if i = k and is-blocked-off(v[1,i],C'[1,i],i) then let S := S, (k,k + 1), v := v(k,k + 1), and
            L := L - (k + 1);
            end if;
            for m in L;
                if v(i) \prec v(m) \preceq c and not is-blocked-off(v(i,m)[1,i],C'[1,i],i) then let S := S, (i,m),
                    v := v(i,m);
            end if;
        end for;
    let T_{ii} := S, u_{ii} := v, L := (i-1, \ldots, 1);
    for m in L;
        if v(i) \prec v(m) \preceq c then let S := S, (i,m), v := v(i,m);
            end if;
        end for;
    return(S,v);
end.
\end{verbatim}

is-blocked-off(u[1,i],C'[1,i],i);
    Let a := u(i), b := C'(i);
    if a \neq b and |a| \leq b and b > 0
        and \{1, \ldots, b\} \subseteq |u[1,i]|, |C'[1,i]|
        and \#\{l : u(l) < 0, C'(l) > 0\} \mod 2 = 1
            then return TRUE;
        else return FALSE;
    end if;
end.

\textbf{Remark 5.17.} Condition 1 gives that \( C(l) \neq C'(i) \) for any \( l < i \). This implies that \( C'(i) \) appears in some position \( q > k \) in \( u \). Further, since two columns cannot be blocked off at \( i \) with \( C(i) = C'(i) \), there is no fear of the algorithm skipping the target value. This ensures that the algorithm
terminates correctly. The algorithm also respects Lemmas 5.1, 5.9, 5.11 and 5.13 which together give the uniqueness of the path.

There are two parts of quantum Bruhat criterion in type $B_n$: the sign criterion and the transposing over values criterion. The following will show that the constructed path follows both.

**Lemma 5.18.** If the skip Path$_C$ step is not called in the algorithm, then $u_i(i) = M_I$. Otherwise, $u_{iii}(i) = M_I$.

**Proof.** Suppose that $u_i \neq M_I$, then we have that $M_I \in u_i[k + 1, n]$ and $u_i(i) < M_I \leq C'(i)$. Note that $M_I \neq C'(i)$, as we know that the algorithm terminates correctly. The remaining strict inequality means that we skipped $M_I$ in the Path$_C$ algorithm. By Lemma 5.11, this means that the act of transposing the value $M_I$ into position $i$ would have caused the current word to have been blocked off with $C'$ at $i$ by $C'(i)$. Recall from the construction in Lemma 5.11, we must have that if $M_I < 0$, then $1 \leq M_I < u_{iii}(i) < t \leq n$ and if $M_I > 0$, then $\pi \leq M_I < u_{iii}(i) < M_I < t < n$. Further, by the block off condition imposed on the entries in positions $k+1$ through $k+1$, we have in both cases that the inequality is really an equality and so $u_{iii} = M_I$. \[\Box\]

**Lemma 5.19.** Let $u, i, C'$ be inputs for the algorithm and $u_{ii}$ be as given in the algorithm. Then $\pi \leq u_{iii} \leq C'(i)$.

**Proof.** If the skip Path$_C$ step was not called in the algorithm, then by Lemma 5.18 and Lemma 5.2 we know that $u_i(i) = M_I \in [C'(i), C'(i)]$. If $C'(i) > 0$, the result follows. Otherwise, we may have that $1 \leq C'(i) < u_i(i) < \pi$, in which case stage II of the algorithm would apply $(i, i)$ to $u_i$, giving $\pi \leq u_{iii} \leq C'(i)$.

If the skip Path$_C$ step is called at some point in stage I of the algorithm, then we know that $C'(i) > 0$ and $C'(i) < u_i(i) < C'(i)$. Further, the only value from $[C'(i)]$ in $u[k+1, n]$ is $M_I$. However, the algorithm skips $M_I$ and so we continue to have $C'(i) < u_i(i) < C'(i)$. After stage II of the algorithm, we can refine this inequality to be $\pi \leq u_{iii} < C'(i) < C'(i)$. \[\Box\]

**Remark 5.20.** There is no sign criterion for stage I transpositions and the criterion for stage II is built in to the algorithm. The fact that the procedure uses the Path$_C$ algorithm in stages III and IV along with Lemma 5.19 give that stage III and IV upsteps maintain the same sign and downsteps have a sign change of negative to positive.

**Remarks 5.21.** We now consider the criterion respecting the transposing of the value in position $i$ across the positions $i + 1, \ldots, n, \pi, \ldots, k + 1$ and $\gamma, \ldots, \zeta$.

1. For the positions $i + 1, \ldots, k$, recall from the proof of Proposition 5.7 that even if there is some $i < l \leq k$ such that $C(i) < C'(i) < C'(i)$, we have that $C'(i) < u(i) = CT_kT_{k-1} \ldots T_{i+1}(i) \leq C'(i)$. Therefore every value in $x \in u[i+1, k]$ is such that $x < u(i) < C'(i)$. The criterion is then met by Lemma 5.1 giving the monotonicity of path in position $i$ during the reflections in $T_i$.

2. If we do not skip Path$_C$ transposition in stage I, then the Path$_C$ algorithm will be used throughout, and there is no fear of transposing a value in position $i$ over the positions $k + 1, \ldots, n, \pi, \ldots, k + 1$ and $\gamma, \ldots, \zeta$.

3. If we do skip Path$_C$ transposition in stage I, then we know from Lemma 5.18 that the skipped value was $M_I$. Further, the procedure will only skip a Path$_C$ step if the corresponding transposition would result in the current word being blocked off with $C'$ at $i$ by $C'(i)$. This means that there are no values from $[C'(i)]$, other than $M_I$ and $M_I$, in positions $[k + 1, k + 1]$ in $u, u_i, u_{ii}$ and $u_{iii}$. Thus the only value in positions $[k + 1, k + 1]$ which would lead to possibly transposing over $M_I$ would be $M_I$, but this is permitted by the quantum bruhat criterion. Recall that $u_{iii} = M_I$. If $M_I = M_{III}$, then the Path$_C$ algorithm guarantees that the value in position $i$ will not transpose across positions $[\gamma, \zeta]$ during
stage IV transpositions. If \( \overline{M_I} = \overline{M_{III}} \), then we note that transposing \( \overline{M_I} \) across \( M_I \) is permissible by quantum Bruhat criterion, and thereafter the Path,\( C \) algorithm procedure will guarantee that the value in position \( i \) will not transpose across positions \([\overline{i}, \overline{2}]\) during the remainder of the stage IV transpositions.

The following lemma will further the results from Lemma 5.1 to all of \( T \) rather than just in \( T_i \). The proof follows similarly to work in \([18]\) with some additions concerning type \( B_n \).

**Lemma 5.22.** If such a path hypothesised in Proposition 5.8 exists and \( u(l) \neq v(l) \), then \( C(l) = u(l) < v(l) \leq C'(l) \) for \( l = 1, \ldots, i - 1 \).

**Proof.** Suppose that this fails for some \( l \) and let \( l_1 < i \) be the largest such \( l \). Let \( w \) be the signed permutation to which the reflection \( (i, \overline{i}) \) is applied in stage IV. Then we have that

\[
a := w(i) \prec b := \overline{C'(l_1)} \prec c_1 := w(l_1).
\]

Now let \( \overline{C} := w[1, k] \) and \((i, \overline{i}), (i, \overline{i_2}), \ldots, (i, \overline{l_p})\) be the remainder of the stage IV roots in \( T_1 \) where \( l_1 > l_2 > \ldots > l_p \). If \( c_i := w(l_i) \), we then have that \( a < b < c_1 < c_2 < \ldots < c_p \) by Lemma 5.1.

By the quantum Bruhat criterion, there can be no values between \( a \) and \( c_p \) in \( w[i, \overline{l}] \). Therefore, for any \( x \in \{b, b + 1, \ldots, c_p\} \), either \( x \) or \( \overline{x} \) is in \( \overline{C}[1, i] \), say in position \( j \). We claim that the possible values for \( C'(j) \) are \( \{\pm b, \pm(b + 1), \ldots, \pm(c_p - 1)\} \). Note then that overall we have too few choices for these positions, which gives a contradiction. We prove the claim in the following two cases.

**Case 1:** \( \overline{C}(j) \in \{b, b + 1, \ldots, c_p\} \). Since \( w(l_p) = \overline{c_p} \), we know that \( x \) is not \( c_p \). Also, since \( C'(i) = c_p \), we have that \( C'(j) \in \{x, x + 1, \ldots, c_p - 1\} \) by Reorder condition 2 unless \( \overline{CC'}t_{i_1} \) is blocked off at \( j \) by \( c_p \). However, since \( w \) is in stage IV of \( T_1 \), we know from Lemma 5.18 that \( \overline{c_p} \prec a \prec c_p \), but \( a = \overline{C}(i) \), so the blocked off condition cannot hold.

**Case 2:** \( \overline{C}(j) \in \{\overline{c_p}, \ldots, \overline{b}\} \). Then \( j \leq l_1 \), otherwise one of the roots \((i, \overline{l})\) in \( T \) will transpose values over \( x \) in position \( j \), breaking the quantum Bruhat criterion. Since \( \overline{C'}(l_1) = \overline{b} \), we can assume \( j < l_1 \). Reorder criterion 2 then gives that \( C'(j) \in \{\overline{x}, \ldots, \overline{b + 1}\} \) unless \( \overline{CC'}t_{j1} \) is blocked off at \( j \) by \( \overline{b} \). But this cannot be, as \( |\overline{C}(j)| > \overline{b} \) by assumption.

Thus the claim is proven and the lemma holds.

**Proof.** (Proof of Proposition 5.8)

Assume that \( u(i) \neq C'(i) \). Then, given Remarks 5.17, 5.20, and 5.21, the only fact left to prove is that the reflections in stage IV satisfy the condition in the corresponding quantum Bruhat graph criterion which refers to the transposition of the value in position \( i \) across the positions \( \overline{k}, \ldots, \overline{i + 1} \). Suppose that at some point in stage IV we apply to the current permutation \( w \) a reflection \((i, \overline{i})\), with \( l < i \), such that for some \( j \in \{i + 1, \ldots, k\} \) we have \( w(i) \prec w(\overline{j}) = C'(j) \prec w(\overline{l}) \) or equivalently,

\[
w(l) < w(j) = C'(j) < w(\overline{l}).
\]

Then by Lemma 5.22 we have that

\[
C(l) \leq w(l) < w(\overline{l}) \leq C'(l).
\]

However Proposition 5.7 showed that we have \( C(l) < C'(j) < C'(l) \) only if \( CC't_{i_1} \) is blocked off at \( j \) by \( C'(j) \). But from Lemma 5.2 we know that \( C_j(j) \leq C_j(l) \leq C'(l) \) where we note that \( C_j(j) = w(j) = C'(j) \). Lemma 5.22 also gives that \( C_j(l) \leq w(l) \leq C'(l) \). However we now have that \( C_j(j) = w(j) \leq C_j(l) \leq w(l) \leq C'(l) \) and \( w(l) < w(j) < C'(l) \), a contradiction. Thus no such \( j \) exists. This completes the proof that the algorithm constructs a path in the quantum Bruhat graph. \( \square \)
Now that we have shown that there is a unique path to determine each \( T \) and that our algorithm follows that path, we can now show that the third condition given on two columns \( CC' \) from Conditions 4.6 is indeed necessary to build the entire path given in Proposition 4.9.

**Lemma 5.23.** The two columns \( CC' \) are not blocked off by \( C'[i] \) for any \( 1 \leq i < k \).

**Proof.** It suffices to show that if \( CC' \) is blocked off at some \( 1 \leq i < k \) by \( b = C'[i] \), then \( CT_kT_{k-1} \ldots T_{i+1}C' \) is blocked off at \( i \) as well in which case the result follows from Lemma 5.9.

Suppose there is such an occurrence. Then for some transposition in \( t_{im} \in T \) for \( 1 \leq m \leq i < l \leq k \) and the current word \( w \) we have that \( wC' \) is blocked off at \( i \) but \( wt_{im}C' \) is not blocked off at \( i \). Due to Lemma 5.6 we must only check that \( t_{im} \) does not make it so that \( wt_{it}[i] \in [b, b+1] \) and that \( wt_{im}[1, i] \) and \( C'[1, i] \) do not have an even number of \(+\) pairs.

For the first, we suppose that \( m = i \). Since \( wC' \) is blocked off at \( i \), it must be that \( w[i], w[\overline{i}] \in [b, b] \) and \( w[l], C'[l] \in [b+1, b+1] \). By monotonicity of path given in Lemma 5.1 we know that \( w[l] < w[\overline{1}] < C'[l] \) but then \( \overline{b} \leq w[i] < b = C'[i] < \overline{w[l]} = wt_{it} \) which contradicts Lemma 5.22. Thus \( m \neq i \).

For the second, we suppose that \( 1 \leq m < i \) and \( t_{im} \) is a stage IV downstep. Then \( w[l] < 0 \) and \( wt_{im} > 0 \). Lemma 5.1 gives that \( w[l] < wt_{im} < C'[l] \) and since stage IV transpositions cannot change sign \( + \) to \( - \), it must be that \( C'[l] > 0 \) as well. Therefore \( w[l] < \overline{a} < C'[l] \) which means that at some point during \( T \) we would need to transpose over \( \overline{a} \) in position \( 7 \), breaking the quantum Bruhat criterion. Thus there is no such stage IV downstep.

\( \square \)

The procedure \textit{path-B} does not allow adjacent columns to be blocked off at \( i \) while applying transpositions in \( T \). The following lemma shows that these same transpositions never cause the columns to be blocked off at any other position either.

**Lemma 5.24.** Let \( u = u_0, u_1, \ldots, u_q = v \) be a path as hypothesised in Proposition 5.8. Then \( u_j[1, l], C'[1, l] \) is not blocked off at \( l \) by \( C'(l) \) for any \( 0 \leq j \leq q \) and \( 1 \leq l \leq i \) except possibly for \( u_0 \) if \( i = l = k \).

**Proof.** This is clearly true for \( l = i \) by the procedure \textit{path-B}. Further, this is true for \( j = 0 \) by the Conditions 1 and 2. Suppose that there is some \( j > 0 \) and \( l < i \) such that \( u_j[1, l], C'[1, l] \) is blocked off at \( l \) by \( C'(l) \). Without loss of generality, let \( j \) be the minimum and \( l \) be the maximum of all such occurrences. This means that \( u_{j-1}[1, l], C'[1, l] \) is not blocked off at \( l \) by \( C'(l) \) and \( u_j = u_{j-1}t_{im} \) some \( 1 \leq m \leq l \). We will show that this implies that \( u_j[1, i], C'[1, i] \) is blocked off at \( i \) by \( C'(i) \), contradicting procedure \textit{path-B}.

We split this into two cases: one where the transposition \( t_{im} \) places a value between \( b \) and \( b \) into position \( m \) and a second where the transposition \( t_{im} \) changes the number of \(+\) pairs in the first \( l \) positions of \( u_j, C' \) to be odd.

Case 1: Here we have that \( u_{j-1}[i] \) is in \( C'[l], C'[l] \) but \( u_{j-1}[m] \) is not and neither is \( C'[i] \) since \( u_j \) and \( C' \) are blocked off at \( l \) by \( C'[l] \) by hypothesis. Also, by Lemma 5.1 we have that \( u_{j-1}[i] \equiv u_{j-1}[m] \equiv C'[i] \).

We first show that \( C'[i] > 0 \) and that \( |u_{j-1}[i]| \leq C'[i] \). Note that if \( u_{j-1}[i] < 0 \), then \( u_{j-1}[m] > 0 \) since there are no stage IV sign preserving down steps. Further, this means that \( C'[i] > 0 \) as well, since there are no stage IV sign changing up steps. Similarly, if \( u_{j-1}[i] > 0 \), then \( u_{j-1}[m], C'[i] > 0 \) since all stage IV up steps preserve sign. So either way, \( C'[i] > 0 \) and it follows that \( |u_{j-1}[i]| \leq C'[i] \) since \( u_{j-1}[i] \) is in \( C'[l], C'[l] \) but \( C'[i] \) is not.
Since stage IV moves preserves the signs in both positions \( C \) and \( l \), we get that \( u_l \) are blocked off at \( t \) in position \( l \). Now, if there is 1 \( C \) row placed, \( u \) at \( t \) in position \( l \). Since there are no changes of positive to negative sign via a stage IV transposition, the change to \( u \) and \( l \) \( + \) pairs in the first \( l \) rows to be odd. As a starting point, since \( u \) and \( C \) are blocked off at \( l \), it must be that there are an odd number of \( + \) pairs in the first \( l \) positions of these two columns. We claim that there are no \( + \) pairs in \( u_{j-1}[l+1, i-1] \) and \( C'[l+1, i-1] \). Since stage IV moves preserves the signs in both positions \( i \) and \( m \) or changes both from \( - \) to \( + \), we get that \( u_{j-1} \) and \( C' \) also has an odd number of \( + \) pairs in the first \( i \) positions.

We conclude by proving the claim. Suppose that there is some \( l < x < i \) such that \( u_{j-1}[x] < 0 \) and \( C'[x] > 0 \). First note that because the transposition \( t_{x} \) followed the QBG criterion, it must be that \( u_{j-1}[x] < u_{j-1}[m] \). Also, \( C'[x] < u_{j-1}[x] < n \), otherwise we would transpose over \( u_{j-1}[x] \). But then \( u_{j-1}[x] \) will have to transpose over \( C'[i] \) in position \( τ \) at some point during the application of \( T \) \( T \) in position \( x \). This contradicts the QBG criterion, and so no such \( + \) pair can exist.

Case 2: Here we show that if the first two block off conditions hold for blocking off \( u_{j-1} \) and \( C' \) at \( l \) then the transposition \( t_{x} \) cannot change the number of \( + \) pairs in the first \( l \) rows to be odd. Since there are no changes of positive to negative sign via a stage IV transposition, the change to an odd number of \( + \) pairs must be through the removal of a \( - \) pair in position \( m \). This means that \( u_{j-1}[i] < 0 \) and \( u_{j}[i] > 0 \). By our starting assumption for this case, we know that \( C'[l] > 0 \) and \( u_{j-1}[l] \) is in \( C'[l], C'[l] \) but \( u_{j-1}[i] < 0 \) and \( u_{j}[i] > 0 \) are not. This means that when applying \( t_{x} \), \( u_{j-1}[i] \) transposes over \( u_{j-1}[l] \) in position \( l \), breaking the QBG criterion.

Proof. (of Proposition 4.8)
The uniqueness of path and monotonicity follows from Proposition 5.8. To prove that the 3 conditions implies the existence of the chain, we iterate the construction in Proposition 5.8 using Algorithm Path_B for \( i = k, k-1, \ldots, 1 \). It remains to be shown that if the 3 necessary conditions hold for \( C'_l \), then they hold for \( C_{l+1} \) as well. We see that Lemma 5.22 gives that the first condition holds as well as that if \( C'_l[l_1] < C'_l[l_2] < C'_l[l_2] \) then \( C_{l+1}[l_1] < C_{l+1}[l_2] \) \( < C'_l[l_2] \) for \( 1 \leq l_1 < l_2 \leq i \). Now, if there is \( 1 \leq l_1 < l_2 \leq i \) such that \( C'_l[l_1] < C'_l[l_2] \) \( < C'_l[l_1] \) then we have that \( C'_l \) and \( C'_l[l_1, l_2] \) are blocked off at \( l_1 \) by \( C'_l[l_2] \) by hypothesis. Then if \( C_{l+1}[l_1] \) \( < C'_l[l_2] < C'_l[l_1] \), it must be that \( C_{l+1} \) and \( C'_l[l_1, l_2] \) are blocked off at \( l_1 \) by \( C'_l[l_2] \) by proof of Lemma 5.23. To finish, we acknowledge that \( C_{l+1} \) and \( C' \) are not blocked off at any \( l \leq i-1 \) by Lemma 5.24.

6. Proof of Proposition 4.10

6.1. Classifying Split, Extended columns. We now work towards building a subpath of \( Γ_r \) between the two height \( k \) columns resulting from the split extension of a KN column of height some multiple of 2 less than \( k \). We do so by first classifying some properties of such pairs of columns, and then showing that these properties are sufficient to build a unique path in the QBG. For this part, we will follow the work of [1] where a subcase of such pairs of columns were classified.

Conditions 6.1. Consider the following conditions on a pair of columns \( CC' \).

1. We have \( \{C(i); i = 1, \ldots, k\} = \{C'(i); i = 1, \ldots, k\} \).
\( \text{int}(C, C') := \left( \bigcup_{i=1}^{k} \{ j \in [\pi] : C(i) < j < C'(i) \} \right) \setminus \{ \pm C(i) : i = 1, \ldots, k \}, \)

then we have \( \text{int}(C, C') = \emptyset. \)

(3) If \( C(i) \) and \( C'(i) \) are the same sign, then \( C(i) < C'(i). \) Additionally, there are an even number of entries where \( C(i) \) is negative and \( C'(i) \) is positive.

(4) \( CC' \) follow the Conditions 4.6.

We will first define an initial matching on the set of columns \( \hat{K}N_k \) and then show that this matching satisfies Conditions 1, 2, and 3. We will follow up with a reordering of this matching, and then show that these reordered matchings continue to follow the three conditions as well as Condition 4. Further, we will later find that these conditions are the necessary requirements to determine a segment of the QBG path.

**Definition 6.2.** \([1]\)

We define a pair of columns \( BB' \) which we will call the initial matching. Let \( B := \hat{A} \) and define \( B' \) by matching each value in \( B \) from top to bottom with a value in \( rA \) as follows:

1. If \( a \) was not involved with the splitting or extending process, match \( a \) to itself.
2. If \( a \in [n] \) is non-zero and required splitting, match \( b \) to \( a \) and \( a \) to \( b \) where \( b \in [n] \) is the value used to split \( a \).
3. If \( a \in [n] \) is the result of a zero splitting, match \( a \) with \( a \).
4. If \( a \in [n] \) is a result of extending, match \( a \) with \( a \).

Since the initial matching is a direct result of the splitting extending algorithms, we will refer to an initial matching and a split extended \( KN \) column interchangeably.

**Lemma 6.3.** \([1]\) Any initial matching satisfies Conditions 1, 2, and 3.

We now give an algorithm which takes an initial matching \( BB' \) and produces what we will call a corrected matching \( CC' \) where \( C \) is the increasing column of entries from \( B \) and \( C' \) is a reordering of \( B' \).

**Definition 6.4.** Given a pair of columns \( BB' \) defined by an initial matching, we produce a corrected matching \( CC' \) by the following algorithm:

Let \( C := B \) and \( C' := B' \);

for \( i \) from 1 to \( k - 1 \) do

\[
\text{let } j \geq i \text{ be such that } C'(j) = \min_{C(j)} \{ C'(l) : l = i, \ldots, k \};
\]

\[
\text{let } C' = C't_{ij};
\]

end;

We also recall the conditions set on two columns in types \( A_{n-1} \) and \( C_n \).

**Definition 6.5.** We will refer to the following two conditions as Conditions 4'. For any pair of indices \( 1 \leq i < l \leq k \),

1. \( C(i) \neq C'(l) \)
2. and the statement \( C(i) < C'(l) < C'(i) \) is false

**Lemma 6.6.** Any corrected matching \( CC' \) satisfies Conditions 1, 2, 3, and 4.

**Proof.** It was shown in \([1]\) that the corrected matching \( CC' \) satisfies Conditions 1, 2, 3, and 4'. It remains to be shown that 4 holds. Note that 4' implies that parts 1 and 2 of Condition 4. We finish by showing that the corrected matching \( CC' \) is not blocked off for any \( 1 \leq i \leq k \). Suppose that there is such an \( i \) where \( CC' \) is blocked off. Since there must be an odd number of \( -- \) pairs
in the first $i$ rows, there must be at least one negative value in $C$ in position less than or equal to $i$. Since $C$ is increasing, it must be that $i = k$. If not, then $|C[k]| < |C[i]| \leq C'[i]$, which contradicts the block off assumption. However, $CC'$ cannot be blocked off at $k$ because Condition 3 gives that there are an even number of $-+$ pairs.

We now have that there is a matching which follows the four conditions for when $C$ is ordered increasingly. Next, we show that there is a matching that follows the four conditions for any reordering of $C$.

**Definition 6.7.** Given a corrected matching $CC'$ and $\sigma \in S_k$, we produce a reordered matching $DD'$ by the following algorithm:

Let $D := C\sigma$ and $D' := C'\sigma$;

for $i$ from 1 to $k - 1$ do

let $j \geq i$ be such that $D'(j) = \min_{D(j)} \{D'(l) : l = i, \ldots, k$ such that $DD't_{il}$ is not blocked off at $i\}$;

let $D' = D't_{ij}$;

end;

We will refer to the result of each $i$th iteration of the algorithm by $DD'_i$. We let $CC' = DD'_0$.

**Remark 6.8.** For the following lemmas, we consider an equivalent version of condition 2. For any two columns $CC'$, let

$$int_i(C, C') := \{j \in [n] : C(i) < j < C'(i)\} \cap \{\pm C(i) : i = k + 1, \ldots, n\}.$$ 

Then $int(C, C') = \bigcup_{i=1}^{k} int_i(C, C')$.

The following lemma shows that transpositions, which do not result from block-off avoidance, in the reorder algorithm maintain conditions 1,2 and 3. the method for proving follows similarly to techniques used in [1], as it pertained to a subset of columns where the the block-off condition would not appear.

**Lemma 6.9.** Let $DD'_{t_{i-1}}$ be the result of applying $i - 1$ iterations of the algorithm in 6.7 to corrected columns $CC'$ and suppose that the $i$th iteration of the algorithm, which calls for some transposition $t_{ij}$, does not result from a block off avoidance. If $int_i(D, D'_{t_{i-1}}) = int_j(D, D'_{t_{i-1}}) = \emptyset$ and condition 3 holds for $DD'_{t_{i-1}}$, then $int_i(D, D'_i) = int_j(D, D'_i) = \emptyset$ and condition 3 holds for $DD'_i$.

**Proof.** We begin by noting that if $i = j$, then the result follows, as we then have $DD'_{t_{i-1}} = DD'_i$. Now assume that $j > i$. Since the algorithm did not choose $D'_{t_{i-1}}[j]$ as an act of block-off avoidance, we have that $D[i] < D'_{t_{i-1}}[j] < D'_{t_{i-1}}[i]$. There are then three places where $D[j]$ could lie within this ordering. Notice that we cannot have $D[i] < D'_{t_{i-1}}[j] < D[j] < D'_{t_{i-1}}[i]$, as it would break the hypothesis that $int_i(D, D'_{t_{i-1}}) = int_j(D, D'_{t_{i-1}}) = \emptyset$ because $k < n$. It must then be that we have either $D[j] < D[i] < D'_{t_{i-1}}[j] < D'_{t_{i-1}}[i]$ or $D[i] < D[j] < D'_{t_{i-1}}[j] < D'_{t_{i-1}}[i]$, which we will refer to as case I and case II respectively. Notice that in both cases, we have that $int_i(D, D'_i), int_j(D, D'_i) \subseteq int_i(D, D'_{t_{i-1}}) \cup int_j(D, D'_{t_{i-1}}) = \emptyset$. It remains to be shown that condition 3 holds for $DD'_i$. This clearly holds for pairs in positions other tha $i$ and $j$, as they are unchanged in this stage of the algorithm and it was assumed that condition 3 held for $DD'_{t_{i-1}}$.

We now show the monotonicity in matched pairs of the same sign and first check for position $i$. Since $k < n$ and $int_i(D, D'_i) = \emptyset$, there is some $a \in [n]$ such that $a, \bar{a} \notin [D[i], D'[i]]$. Suppose that $sgn(D[i]) = sgn(D'[i])$. If condition 3 does not hold at $i$, we have that $D'[i] < D[i]$. If $D[i] \in [n]$, then we must have that $D[i] < \bar{a} < D'[i]$, a contradiction to our choice of $a$. Similarly,
if \(D[i] \in [\pi] \setminus [n]\), then \(D[i] < a < D'_i[i]\), again contradicting our choice of \(a\). Thus \(D[i] < D'_i[i]\) as desired. The proof for position \(j\) follows similarly.

We now show that the number of \(-+\) pairs remains unchanged after the application of \(t_{ij}\) to \(D'\). As before, since \(k < n\) and \(\text{int}_i(D, D'_{i-1}) \cup \text{int}_j(D, D'_{i-1}) = \emptyset\), there is some \(a \in [n]\) such that \(a, \bar{a} \notin \{D[i], D'_i[i]\}\) and \(a, \bar{a} \notin \{D[j], D'_i[j]\}\). This means that the progression through the four values in the orderings of case I and case II above never pass over the values \(a\) or \(\bar{a}\) and so the sign can only change once (either positive to negative or negative to positive). If we consider all eight configurations of four \(+\) and \(-\) values ordered with only one sign change (for example, \(+ + + +\) or \(- - - -\)) and compare them to the inequalities of case I and case II, we see that in both cases the number of \(-+\) pairs is preserved.

**Corollary 6.10.** Let \(DD'_{i-1}\) be the result of applying \(i - 1\) iterations of the algorithm in 6.7 to corrected columns \(CC'\) and suppose that the \(i^{th}\) iteration of of the algorithm, which calls for some transposition \(t_{ij}\), does not result from a block off avoidance. If conditions 2 and 3 hold for \(D D'_{i-1}\), then conditions 2 and 3 hold for \(D D'_{i}\).

**Lemma 6.11.** Let \(DD'_{i-1}\) be the result of applying \(i - 1\) iterations of the algorithm in 6.7 to corrected columns \(CC'\) and suppose that the \(i^{th}\) iteration of of the algorithm, which calls for some transposition \(t_{ij}\), results from a block off avoidance. If conditions 2 and 3 hold for \(D D'_{i-1}\), then conditions 2 and 3 hold for \(D D'_{j}\) for some \(i < j \leq j\).

*Proof.* The case where \(i = j\) is trivial, so suppose that \(j > i\). Since the algorithm choose \(D'_{i-1}[j]\) from a block-off avoidance, we have that \(D[i] < D'_{i-1}[i] < D'_{i-1}[j]\). There are then three places where \(D[j]\) could lie in within this ordering. If we have \(D[j] < D[i] < D'_{i-1}[i] < D'_{i-1}[j]\) or \(D[i] < D[j] < D'_{i-1}[i] < D'_{i-1}[j]\), then we again have that \(\text{int}_i(D, D'_{i}), \text{int}_j(D, D'_{i}) \subseteq \text{int}_i(D, D'_{i-1}) \cup \text{int}_j(D, D'_{i-1}) = \emptyset\) and so condition 2 still holds. Condition 3 then holds via similar argumentation to that in the proof of Lemma 6.9.

Unfortunately, if we have \(D[i] < D'_{i}[i] < D[j] < D'_{i}[j]\), we are not guaranteed that \(\text{int}_j(D, D'_{i}) = \emptyset\) or the monotonicity of values in row \(j\) of \(D D'_{i}\). For this case, we will first show that the parity condition of \(-+\) pairs still holds for \(D D'_{i}\) and that the monotonicity of same sign row entries holds for all rows not \(j\). Then we use Lemma 6.9 and the fact that the transposition is avoiding a block off configuration to then show that by the end of the \(j^{th}\) iteration of the algorithm, conditions 2 and 3 are restored.

We first note that \(D D'_{i-1}\) must have been blocked off at \(i\) by \(D'_i[i]\). If not, the \(i^{th}\) iteration of the algorithm would have called for the \(t_{ii}\). Since \(D D'_{i-1}\) was blocked off at \(i\) (and therefore an odd number of \(-+\) pairs in the first \(i\) rows), and condition 3 held for \(D D'_{i-1}\) (so there is an even number of \(-+\) pairs in the columns), there must be some \(-+\) pair in some row \(j' > i\). Since all of the values (up to absolute value) between \(\overline{D'_{i-1}[i]}\) and \(\overline{D'_{i-1}[i]}\) must lie in the first \(i\) rows of \(D D'_{i-1}\), it must be that \(\overline{D[j']} < \overline{D'_{i-1}[i]} < D[i] < D'_{i-1}[i] < D'_{i-1}[j] \leq D'_{i-1}[j'] \leq n\). This containment of inequalities along with the fact that condition 3 held for \(D D'_{i-1}\) gives that \(\text{int}_j(D, D'_{i}) \cap \text{int}_j(D, D'_{i-1}) = \emptyset\). Furthermore, the algorithm gives that \(D'_{i-1}[i] < D'_{i-1}[j] < D'_{i-1}[j']\), so it must be that \(D'_{i-1}[i] > 0\). Thus \(D'_{i-1}[i] < D[j] < D'_{i-1}[j]\) are all positive values and so \(D D'_{i}\) fails the monotonicity of condition 3 in row \(j\). Since \(D D'_{i-1}\) was blocked off at \(i\), it must be that \(D'_{i-1}[i] > 0\) as well. Thus the number of \(-+\) pairs is unchanged by the transposition \(t_{ij}\) and so the number of \(-+\) pairs in \(D D'_{i}\) remains even. Finally, since we have from above that \(\overline{D'_{i-1}[i]} < D[i] < D'_{i-1}[i] < D'_{i-1}[j] \leq n\), we have the monotonicity of same sign values (if it applies) in row \(i\) of \(D D'_{i}\).
Let \( j = \min(j, j') \). Note that during the \( i+1 \) through \( j-1 \) iterations of the algorithm, there will never be a block off avoidance move. Indeed, it would have to be blocked off at some value greater than \( D'_{j-1}[i] \), but this value is in position \( j \). We further note that none of these stages of the algorithm will ever call for a transposition with position \( j \). This follows from the fact that none of the pairs in these iterations are \( + - \) pairs by our choice of \( j' \) and that \( D'_{j-1}[i] \) is the minimum positive value below position \( i \). Therefore, by Lemma \[6.9\] we have that \( DD'_{j-1} \) still follows conditions 2 and 3 everywhere but in position \( j \). We now break into two cases, depending on the value of \( j \).

Suppose that \( j = j' \). Then the algorithm will call for the transposition \( t_{j'j} \) and if we consider the fact that \( int_{j'}(DD'_{j'1}) = \emptyset \) and the ordering of these values given above, we see that both conditions 2 and 3 once again hold.

Suppose that \( j = j' \). Then the \( j^{th} \) iteration of the algorithm again calls for the transposition \( t_{j'j} \) and places some \( x \) in a position where \( D[j] < x < D'_{j-1}[j'] \) and conditions 2 and 3 are restored. \( \square \)

We will use the following proposition to prove Proposition \[4.10\].

**Proposition 6.12.** A corrected matching \( CC' \) satisfies conditions 1, 2, 3 and 4 if and only if the corresponding reordered matching \( DD' \) satisfies conditions 1, 2, 3 and 4.

**Proof.** We first show that if \( CC' \) follows the conditions, then so does the corresponding pair \( DD' \). Condition 1 clearly holds, as the values themselves are unaltered in the columns when converting from a corrected matching to a reordered matching. Iterations of Corollary \[6.10\] and Lemma \[6.11\] give that conditions 2 and 3 hold for the reordered columns \( DD' \). Condition 4 holds because it is equivalent to the reorder algorithm. We note here that \( DD' \) is not blocked off at \( k \) because condition 3 gave that there are an even number of \( + - \) pairs in the two columns. The opposite implication is clear by the same reasoning. \( \square \)

It remains to be shown that these same four conditions on two columns are exactly the conditions needed to describe split extended \( KN \) columns of type \( B_n \).

**Proposition 6.13.** The set of corrected matchings \( DD' \) following conditions 1, 2, 3 and 4 are in bijection with the split extended reordered \( KN \) columns of type \( B_n \).

**Proof.** Note that Proposition \[6.12\] reduces the columns to be considered to just those \( CC' \) sorted columns. This case was shown in Theorem 4.4 of [1]. \( \square \)

### 6.2. Building a segment of the QBG Path between Split, Extended columns.

**Lemma 6.14.** The pair \( DD' \) satisfies Conditions \[6.7\] if and only if there is a path \( u = u_0, u_1, \ldots, u_p = v \) in the corresponding quantum Bruhat graph such that \( v_{[1,k]} = C' \) and the edge labels form a sub-sequence of \( \Gamma_r(k) \). Moreover, the mentioned path is unique, and for each \( i = 1, \ldots, k \), we have

\[
C(i) = u_0(i) \preceq u_1(i) \preceq \ldots \preceq u_p(i) = C'(i).
\]

**Proof.** The necessity of Condition 4 is given by Propositions \[5.7\] and \[5.23\]. The necessity of Condition 1 is clear from the selection of roots available in \( \Gamma_r(k) \). The necessity of Conditions 2 and 3 follow from the quantum Bruhat criterion of type \( B_n \). Indeed Condition 2 prevents us from ever transposing over a value in positions \( k + 1, \ldots, n, n, \ldots, k + 1 \) while Condition 3 follows from the sign restrictions of stage IV moves.

Now, if the four conditions are satisfied, then we obtain the desired path through the iteration of the \( Path_C \) algorithm (cf. Proposition \[4.9\] and its proof). Indeed, Condition 1 assures us that the algorithm terminates correctly even with the limited selection of transpositions in \( \Gamma_r(k) \). The monotonicity of same sign pairs from Condition 3 along with Condition 2 assures us that the
algorithm never selections reflections of the form \((i, m)\) or \((i, \overline{m})\) for \(m > k\). The pairity condition of \(\rightarrow \) pairs from Condition 3 assures us that \(DD'\) are not blocked off at \(at\) so that there is no need for the root \((k, k + 1)\), which is unavailable in \(\Gamma_k(k)\). Condition 4 grants use of Lemma 5.24 so every iteration starts off with two columns which are not blocked off.

\[\square\]

We can now prove Proposition 4.10.

**Proof.** (of Proposition 4.10) This follows directly from Lemma 6.14 and Proposition 6.13 \[\square\]

7. **The bijection in type \(D_n\)**

We briefly outline the major differences in the type \(D_n\) constructions. First, since \(KN\) columns of type \(D_n\) have no relation in the ordering of \(n\) and \(\overline{n}\), the type \(D_n\) splitting algorithm “split\_D” begins by converting all \((n, \overline{n})\) pairs in a given column to 0 values, and then it continues as in type \(B_n\) \[15\]. There is still need for the extending algorithm, and we use the same one as in type \(B_n\) (“extend”). The quantum Bruhat graph criterion in type \(D_n\) differs from type \(B_n\) in that we no longer have any arrows of the form \((i, \overline{j})\), but in return we have less restriction concerning arrows of the form \((i, j)\). This change requires further modifications to the path and reordering algorithms, based on the following type \(D_n\) blocked off condition.

**Definition 7.1.** We say that columns \(C = (l_1, l_2, \ldots, l_k)\) and \(C' = (r_1, r_2, \ldots, r_k)\) are type \(D_n\) blocked off at \(i\) by \(b = r_i\) if and only if \(C\) and \(C'\) are blocked off at \(i\) by \(b = r_i\), or the following hold:

1. \([-|l_i| < b < 0, where \(-|l_i| = b\) if and only if \(l_i = b\);
2. \(\{b, b + 1, \ldots, n\} \subset \{|l_1|, |l_2|, \ldots, |l_i|\}\) and \(\{b, b + 1, \ldots, n\} \subset \{|r_1|, |r_2|, \ldots, |r_i|\}\);
3. and \(|\{j : 1 \geq j \geq i, l_j > 0, r_j < 0\}|\) is odd.

We then define “Path\_D” and “ord\_D” to be as in type \(B_n\), but by replacing “blocked off” with “type \(D_n\) blocked off”.

**Theorem 7.2.** The map “Path\_D \circ ord\_D \circ extend \circ split\_D” is the inverse of the type \(D_n\) “sfill\_D” map.

**References**

[1] C. Briggs. On Combinatorial Models for Kirillov-Reshetikhin Crystals of Type B. ProQuest Dissertations And Theses; Thesis (Ph.D.), 79-01(E):56, 2017.
[2] D. Bump and A. Schilling. Crystal Bases: Representations and Combinatorics. World Scientific, 2017.
[3] R. Carter. Lie Algebras of Finite and Affine Type Cambridge Studies in Advanced Mathematics, 96. Cambridge University Press, 2005.
[4] G. Fourier, M. Okado, A. Schilling. Kirillov-Reshetikhin crystals for nonexceptional types. Adv. Math., 222(3):1080-1116, 2009.
[5] W. Fulton. Young Tableaux. Cambridge University Press, 1996.
[6] W. Fulton and C. Woodward. On the quantum product of Schubert classes. J. Algebraic Geom., 13:641-661, 2004.
[7] J. Hong and S. Kang. Introduction to Quantum Groups and Crystal Bases. Graduate Studies in Mathematics, 42, 2002.
[8] J. Humphreys. Reflection Groups and Coxeter Groups. Cambridge University Press, 1990.
[9] M. Kashiwara. On crystal bases of the q-analogue of universal enveloping algebras. Duke Math. J., 63:465-516, 1991.
[10] M. Kashiwara. Crystal bases of modified quantized enveloping algebra. Duke Math. J., 73:383-413, 1994.
[11] M. Kashiwara and T. Nakashima. Crystal graphs for representations of the q-analogue of classical Lie algebras. J. Algebra, 165:295-345, 1994.
[12] N. Kirillov and N. Reshetikhin. Representations of Yangians and multiplicities of the inclusion of the irreducible components of the tensor product of representations of simple Lie algebras. J. Soviet Math., 52:3156-3164, 1990.
[13] A. Lascoux. Cyclic permutations on words, tableaux and harmonic polynomials. Proceedings of the Hyderabad Conf. on Alg. Groups., 323-247, 1991.
[14] C. Lecouvey. Schensted-type correspondence, plactic monoid, and jeu de taquin for type \(C_n\). J. Algebra, 247: 295-331, 2002.
[15] C. Lecouvey. Schensted-type correspondence and plactic monoids for types $B_n$ and $D_n$. *J. Algebraic Combin.*, 18:99–133, 2003.
[16] C. Lecouvey and C. Lenart. Atomic decomposition of characters and crystals. *Adv. Math.*, IF 1.494, 2020.
[17] C. Lecouvey, M. Okado, and M. Shimozono. Affine crystals, one-dimensional sums and parabolic Lusztig $q$-analogues. *Math. Zeit.*, 271 no.3–4:819–865, 2012.
[18] C. Lenart. From Macdonald polynomials to a charge statistic beyond type $A$. *J. Combin. Theory Ser. A*, 119:683–712, 2012.
[19] C. Lenart and A. Lubovsky. A generalization of the alcove model and its applications. *J. Algebraic Combin.*, 41:751–783, 2015.
[20] C. Lenart and A. Lubovsky. A uniform realization of the combinatorial $R$-matrix. *Adv. Math.*, 334:151-183, 2018.
[21] C. Lenart, S. Naito, D. Sagaki, A. Schilling, M. Shimozono. A uniform model for Kirillov-Reshetikhin crystals I: Lifting the parabolic quantum Bruhat graph. *Int. Math. Res. Not.*, 7:1848-1901, 2015.
[22] C. Lenart, S. Naito, D. Sagaki, A. Schilling, M. Shimozono. A uniform model for Kirillov-Reshetikhin crystals II: Path models and $P=X$. *Int. Math. Res. Not.*, 14:4259–4319, 2017.
[23] C. Lenart, S. Naito, D. Sagaki, A. Schilling, M. Shimozono. A uniform model for Kirillov-Reshetikhin crystals III: Nonsymmetric Macdonald polynomials at $t=0$ and Demazure characters. *Transform. Groups*, 22:1041–1079, 2017.
[24] C. Lenart and A. Postnikov. Affine Weyl groups in $K$-theory and representation theory. *Int. Math. Res. Not.*, Art. ID rnm038:1-65, 2007.
[25] C. Lenart and A. Postnikov. A combinatorial model for crystals of Kac-Moody algebras. *Trans. Amer. Math. Soc.*, 360:4349-4381, 2008.
[26] C. Lenart and A. Schilling. Crystal energy functions via the charge in types $A$ and $C$. *Math. Z.*, 273, 2011.
[27] G. Lusztig. Singularities, character formulas, and a $q$-analogue of weight multiplicities. *Analyse et topologie sure les espaces singuliers (II-III)*, Astérisque 101-102:208-227, 1983.
[28] A. Nakayashiki and Y. Yamada. Kostka polynomials and energy functions in solvable lattice models. *Selecta Math. (N.S.)*, 3:547-599, 1997.
[29] A. Ram and M. Yip. A combinatorial formula for Macdonald polynomials. *Adv. Math.*, 226:309-331, 2011.
[30] J. M. Santos. Symplectic keys and Demazure atoms in type $C$. *Sém. Lothar. Combin.*, 84B , Art. 49, 2020.
[31] J. M. Santos. Symplectic right keys – Type C Willis’ direct way (extended abstract), *arXiv:2104.15000* to appear in Proceedings of Formal Power Series and Algebraic Combinatorics, 2021.
[32] A. Schilling. A bijection between type $D_n^{(1)}$ crystals and rigged configurations. *J. Algebra*, 285:292–334, 2005.
[33] J. Sheats. A symplectic jeu de taquin bijection between the tableaux of King and of De Concini. *Trans. Amer. Math. Soc.*, 351:3569–3607, 1999.

(Carly Briggs) BENNINGTON COLLEGE, ONE COLLEGE DRIVE, BENNINGTON, VT 05201, U.S.A.
*Email address: carlybriggs@bennington.edu*

(Cristian Lenart) DEPARTMENT OF MATHEMATICS AND STATISTICS, STATE UNIVERSITY OF NEW YORK AT ALBANY, ALBANY, NY 12222, U.S.A.
*Email address: clenart@albany.edu*

(Adam Schultze) DEPARTMENT OF MATHEMATICS, STATISTICS, AND COMPUTER SCIENCE, ST. OLAF COLLEGE, 1520 ST. OLAF AVENUE, NORTHFIELD, MN 55057, U.S.A.
*Email address: schultze@stolaf.edu*