Perforated Tableaux: A Combinatorial Model for Crystal Graphs in Type $A_{n-1}$

Glenn D. Appleby · Tamsen Whitehead

Received: 26 August 2021 / Accepted: 19 April 2022 / Published online: 23 May 2022
© The Author(s), under exclusive licence to Springer Nature B.V. 2022

Abstract
We present a combinatorial model, called perforated tableaux, to study $A_{n-1}$ crystals, unifying several previously studied combinatorial models. We identify nodes in the $k$-fold tensor product of the standard crystal with length $k$ words in $[n] = \{1, \ldots, n\}$. We model this crystal with perforated tableaux (ptableaux), extending this identification isomorphically to biwords, RSK $(P, Q)$ tableaux pairs, and matrix models. In the ptableaux setting, crystal operators are more simply defined and we can identify highest weights visually without computation. We generalize the tensor products in the Littlewood-Richardson rule to all of $[n]^\otimes k$, and not just the irreducible crystals whose reading words come from semistandard Young tableaux. We relate evacuation (Lusztig involution) to products of ptableaux crystal operators, and find a combinatorial algorithm to compute commutators of highest weight ptableaux.

Keywords Crystals · Representation theory · Tableaux · Combinatorial models

1 Introduction
We introduce a new combinatorial model for the $GL_n$ crystal structure (of type $A_{n-1}$), called perforated tableaux (or simply, ptableaux), and study its relation to crystal structures on $[n]^\otimes k$, the $k$-fold tensor product of the standard $GL_n$ crystal, denoted by $[n]$, along with its relation to other combintorial models. Perforated tableaux are rectangular arrays containing concatenated horizontal strips of positive integers, and blank boxes (“perforations”), such that the content in non-blank entries weakly increases from left to right and strictly
increases from top to bottom. Entire rows of blanks are allowed; entire columns of blanks are not. For example the diagram below is a perforated tableau:

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

While the $GL_n$ crystal structure in type $A_{n-1}$ is certainly well-studied, the perforated tableau model allows us to unify several previously separate models. The term “perforated tableau” appears in the work of Benkart, Sottile, and Stroomer (see [3]) in their work on tableau-switching algorithms. Our definition is similar to theirs, but not identical, and our use of perforated tableaux as models for crystal graphs is new.

The authors would like to thank Florence Maas-Gariepy for her very careful reading of an early version of this paper and her many useful suggestions, along with the contributions of the referee for their valuable suggestions and for bringing to our attention several results related to ours.

1.1 Combinatorial Models for Crystals

A combinatorial model for a crystal graph is a set of objects used as nodes of the graph (here, of type $A_{n-1}$) so that we can define crystal operators $e_i, f_j$ on such objects by combinatorial means. Often, one can then compute various multiplicities by enumerating some set of combinatorial objects; the overall goal of such models is to gain some insight into the operations involved, and potentially to simplify computations.

Below, we show how to obtain easily-defined bijections between perforated tableau and other well-studied combinatorial models, and describe why the ptableau model may have advantages over them.

Bump and Schilling [5] define $B_n$ as the standard $GL_n$ crystal of type $A_{n-1}$ whose graph is composed of tableaux of one box:

```
  1 \rightarrow f_1 \rightarrow 2 \rightarrow f_2 \rightarrow \cdots \rightarrow f_{n-1} \rightarrow n
```

Instead of $B_n$, we’ll let $[n] = \{1, 2, \ldots, n\}$ denote both the set of nodes of the crystal graph, and the graph itself, but will always make clear by context when we put the standard crystal structure on this set. We let $[n]^{\otimes k}$ denote the tensor product of $k$ copies of the standard $GL_n$ crystal with itself. Thus, nodes in the crystal $[n]^{\otimes k}$ are identified with words of length $k$ from the set $[n]$. The notation $[n]^{\otimes k}$ is used for both this set of words, and the crystal graphs whose nodes are labeled with the words. That is, we identify the node in the crystal $[n]^{\otimes k}$ given by the tensor product $\omega_1 \otimes \omega_2 \otimes \cdots \otimes \omega_{k-1} \otimes \omega_k \in [n]^{\otimes k}$ with the word $\omega_1 \omega_2 \cdots \omega_{k-1} \omega_k$.

There are two choices for defining crystal operators $e_i, f_i$ on $[n]^{\otimes k}$. We will adopt the Kashiwara convention for crystal operators [12] (precise definitions in Section 4 below). This is (essentially) the reverse to what is found in many sources, including Bump-Schilling [5], but it has been used for some recent constructions (See [8]). We do this because it allows us, when reading a word from left to right, to construct an associated perforated tableau, also obtained left to right.

While $[n]^{\otimes k}$ decomposes as a sum of irreducible crystal graphs, and all isomorphism types of $GL_n$ crystal structures are found among them, other combinatorial models for
crystal graphs such as biwords, matrix models, and RSK \((P, Q)\) pairs, have been studied as well \([5–8, 10, 12, 13, 16–18]\).

For example, start with some \(\omega \in [3]^{\otimes 11}\):

\[
\omega = 2122331331,
\]
on which crystal operators \(e_i, f_i\) act within a \(GL_3\) crystal structure. We parse \(\omega\) into weakly decreasing factors, and denote the parsing by \(P\) and the parsed word as \(\omega_P\):

\[
\omega_P = 21|22|331|331.
\]

In Section 3 we define a bijection from parsed words to ptableaux. For now, we associate to \(\omega_P\) a biword by putting constant factors composed of 1’s, and then 2’s, etc., over each successive factor of \(\omega_P\):

\[
\begin{pmatrix}
\tau \\
\omega
\end{pmatrix} = 
\begin{pmatrix}
1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 & 4 \\
2 & 1 & 2 & 2 & 3 & 3 & 1 & 3 & 1
\end{pmatrix}.
\]

The \(GL_3\) crystal structure on this biword is obtained simply by letting the crystal operators act on \(\omega\). It can be shown that such operators preserve the parsing \([5, 7]\).

In addition to biwords, matrix models for crystal graphs have been studied \([5, 7, 16–18]\).

Continuing our example, we compute a \(4 \times 3\) matrix \(M\) associated to our biword, by setting the \((i, j)\) entry of \(M\) equal to the number of times \(i\) appears over \(j\) in the biword:

\[
M = \begin{bmatrix}
1 & 1 & 0 \\
0 & 2 & 0 \\
1 & 0 & 2 \\
1 & 0 & 2
\end{bmatrix}.
\]

This map is a bijection from biwords \((\tau, \omega)^T\), with \(\tau \in [\ell]^{\otimes k}\) and \(\omega \in [n]^{\otimes k}\) to \(\text{Mat}_{\ell \times n}(\mathbb{N})\), the \(\ell \times n\) matrices with non-negative integer entries. In this case, a \(GL_3\) crystal structure can be defined for matrices \(M\) by computing along the columns of \(M\) (See \([5, 18]\)), and one shows this structure commutes with the map from biwords to matrices, yielding isomorphic crystal structures.

Lastly, we can also associate a pair \((P, Q)\) of semistandard Young tableaux (SSYT) to this biword by means of insertion algorithms \([5, 7]\). Again, there is more than one such choice of algorithm. Here we use column insertion on the word \(\omega\), going left to right, using \(\tau\) to determine the recording tableau. In our example, we obtain:

\[
P = \begin{bmatrix}
1 & 1 & 1 & 2 & 2 & 3 & 3 \\
2 & 3 \\
\end{bmatrix}, \quad Q = \begin{bmatrix}
1 & 1 & 2 & 3 & 4 & 4 & 4 \\
2 & 3 \\
\end{bmatrix}.
\]

The \(GL_3\) crystal structure on SSYT \((P, Q)\) of the same shape is that induced by the structure on the word \(\omega\) (one shows that the shape of the associated tableaux, and indeed the tableau \(Q\), are fixed by this action).

It can be shown (using our choices for crystal operators on words, matrix conventions, and insertion schemes) that the maps between these three models (biwords, matrices, RSK \((P, Q)\) pairs) are actually isomorphisms of their crystal graph structures.

Where do perforated tableau fit among these models? There are simply defined bijections from biwords to ptableaux, too. Starting from the biword given above, we work from left to right. The entries in \(\tau\) in the top row will become the content (entries) in an associated ptableau \(T\). If \(i\) in \(\tau\) is above some \(j\) in \(\omega\) of the biword, we put an \(i\) in row \(j\) of \(T\), requiring...
such entries form a horizontal strip (and potentially inserting blanks, or “perforations”) so that the $T$ is column strict:

$$
\begin{pmatrix}
\tau \\
\omega
\end{pmatrix}
= 
\begin{pmatrix}
1 & 1 & 2 & 2 & 3 & 3 & 3 & 4 & 4 & 4 \\
2 & 1 & 2 & 2 & 3 & 3 & 1 & 3 & 3 & 1
\end{pmatrix}
\Rightarrow
T = 
\begin{pmatrix}
1 & 3 & 4 \\
1 & 2 & 2 \\
3 & 3 & 4 & 4
\end{pmatrix}
$$

It is not hard to show this map is a bijection. In Section 4 we define crystal operators $e_i, f_i$ on perforated tableaux (“ptableaux”), and prove that these operators commute with those given by the crystal graph structure on $[n]^\otimes k$, and hence determining, (in general), an $GL_n$ crystal structure on ptableaux isomorphic to the structures of the other examples.

We hope to demonstrate (1) perforated tableaux are often among the simpler models to use for various computations, including computations of crystal operators and determining highest weights, and (2) many combinatorial algorithms that have appeared in several distinct models all naturally appear within the perforated tableaux model.

Among the key advantages realized in perforated tableaux, we have:

1. Computing crystal operators on ptableaux is done with an easy, single, jeu de taquin move, in which content moves down under the lowering operators $f_i$, and upwards with the raising operators $e_i$.
2. A ptableau is highest weight when its content is partition-shaped (unperforated). Thus, we can determine highest weights visually, without computation, unlike the $[n]^\otimes k$, biwords, or matrix models.
3. The perforated tableaux model has a natural and simple tensor product operation.
4. We can give a simple combinatorial enumeration of the Littlewood-Richardson rule for tensor product multiplicities by enumerating certain ptableaux. This rule directly generalizes classical “Littlewood-Richardson fillings.” The “word condition” of the classical case is not necessary generally, but it does locate one particular crystal graph within a fixed isomorphism class of irreducible crystal graphs.
5. A version of Schützenberger evacuation is used to compute the Lusztig involution on the crystal structure of RSK $(P, Q)$ pairs. We prove this evacuation is actually a product of ptableaux crystal operators, so that evacuation on SSYT is viewed, in this context, as the operator taking a highest weight ptableau to its corresponding lowest weight.
6. Let $B_\mu$ and $B_\nu$ be irreducible crystals of highest weights $\mu$ and $\nu$, respectively. Lenart defined a non-combinatorial crystal isomorphism $\phi : B_\mu \otimes B_\nu \rightarrow B_\nu \otimes B_\mu$, called a commutator [14, 15]. We prove a generalized form of tableau switching for ptableaux that computes $\phi$ combinatorially for arbitrary highest weight elements in tensor products of irreducible ptableaux graphs. Our algorithm generalizes related work of James and Kerber [11], and also Benkart, Sottile, and Stroomer [3].

We should note that one key feature of the RSK $(P, Q)$ tableaux model is its use in solving the isomorphism and plactic equivalence problem [5]: if two nodes in a crystal graph are represented by pairs $(P, Q)$ and $(P', Q')$, respectively, then the nodes lie in isomorphic crystals precisely when $Q$ and $Q'$ are the same shape, and they lie in the same irreducible crystal graph when $Q = Q'$. They are plactically equivalent (in same location in isomorphic crystal graphs) when $P = P'$. In [2], we present a beautiful solution to this problem, using ptableaux.

### 1.2 Duality and Bicrystals

Another important feature of the biword, matrix, and RSK pair models is the existence of a duality map, allowing one to put a bicrystal structure on these combinatorial objects (as
GL\(_\ell \times GL_n\) crystals for appropriate \(\ell\), along with several fundamental results relating crystal operators on the \(GL_\ell\) structure with combinatorially defined operations on the \(GL_n\) crystal [5–8, 16, 18]. Defining the dual to a perforated tableau is especially simple, so we mention a few of these connections, taking up the general issue is outside the scope of this paper and will have to be taken up in later work.

**Definition 1.1** Let \(Bi([\ell]^{\otimes k}, [n]^{\otimes k})\) denote the set of length \(k\) biwords over \([\ell]\) and \([n]\):

\[
B \in Bi([\ell]^{\otimes k}, [n]^{\otimes k}), \quad B = \left( \begin{array}{c}
a_1 & a_2 & \cdots & a_k \\
b_1 & b_2 & \cdots & b_k \\
\end{array} \right) = \left( \begin{array}{c}
\tau \\
\omega \\
\end{array} \right),
\]

where \(\tau \in [\ell]^{\otimes k}\) and \(\omega \in [n]^{\otimes k}\). We restrict the words in the top row \(\tau \in [\ell]^{\otimes k}\) to those that are weakly increasing (so \(a_i \leq a_{i+1}\)), and we restrict words \(\omega \in [n]^{\otimes k}\) to those that satisfy \(b_i \geq b_{i+1}\) whenever \(a_i = a_{i+1}\).

We say the **dual** of a biword \((\tau, \omega)^T \in Bi([\ell]^{\otimes k}, [n]^{\otimes k})\) is the biword \((\hat{\omega}, \hat{\tau})^T\), obtained by swapping the top and bottom rows of \((\tau, \omega)^T\), and reordering columns so that \(\hat{\omega}\) is weakly increasing, and under each constant factor of \(\hat{\omega}\) the corresponding factor of \(\hat{\tau}\) is weakly decreasing (that is, so that \((\hat{\omega}, \hat{\tau})^T \in Bi([n]^{\otimes k}, [\ell]^{\otimes k})\). Thus, if

\[
\left( \begin{array}{c}
\tau \\
\omega \\
\end{array} \right) = \left( \begin{array}{c}
1 & 1 & 2 & 2 & 3 & 3 & 3 & 4 & 4 & 4 \\
2 & 1 & 2 & 2 & 3 & 3 & 1 & 3 & 3 & 1 \\
\end{array} \right) \in Bi([4]^{\otimes 11}, [3]^{\otimes 11}),
\]

we swap rows and reorder columns so that

\[
\left( \begin{array}{c}
\hat{\omega} \\
\hat{\tau} \\
\end{array} \right) = \left( \begin{array}{c}
1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\
4 & 3 & 1 & 2 & 2 & 1 & 4 & 4 & 3 & 3 \\
\end{array} \right) \in Bi([3]^{\otimes 11}, [4]^{\otimes 11}).
\]

A \(GL_3 \times GL_4\) bicrystal structure is obtained on the pair \((\hat{\omega}, \hat{\tau})\) letting the \(GL_3\) crystal operators act on \(\omega\) and \(GL_4\) operators act on \(\hat{\tau}\). One shows that these actions commute on the dual pairs \((\hat{\omega}, \hat{\tau})\).

In the matrix setting, duality is even simpler to describe. We already had a \(GL_3\) crystal structure defined by operators acting on the columns of the \(4 \times 3\) matrix \(M\) given above. A \(GL_3 \times GL_4\) bicrystal structure is obtained by defining the corresponding \(GL_4\) structure applied to the columns of \(M^T\), the transpose of \(M\) (or appropriately defined versions of row operations on \(M\)). Again, one shows that these actions commute [5, 18].

For RSK \((P, Q)\) pairs, the dual of \((P, Q)\) is simply \((Q, P)\). As illustrated in our example

\[
P = \begin{bmatrix}
1 & 1 & 1 \\
2 & 3 \\
3 \\
\end{bmatrix}, \quad Q = \begin{bmatrix}
1 & 1 & 2 & 3 & 4 & 4 & 4 \\
2 & 3 \\
3 \\
\end{bmatrix},
\]

\(Q\) has content in \([4]\), and the induced \(GL_4\) structure on \((Q, P)\) is given by acting on the word obtained from \(Q\) by reading its rows from right to left, top to bottom. One proves that this action commutes with the \(GL_3\) crystal structure on \(P\) described above.

We now determine the dual of a perforated tableau. Given a ptableau \(T\), its **dual** \(\hat{T}\) by letting the content in row \(i\) of \(T\), read right to left, name the **rows** in which a horizontal strip of \(i\)'s appears in \(\hat{T}\). Thus, if

\[
T = \begin{bmatrix}
1 & 3 & 4 \\
1 & 2 & 2 \\
3 & 3 & 4 & 4 \\
\end{bmatrix},
\]
the first row (read right to left) says that $\hat{T}$ has a strip of 1’s (moving left to right) appearing in rows 4, 3 and 1, and so on:

$$\hat{T} = \begin{bmatrix} 1 & 2 \\ 2 & 2 \\ 1 & 3 & 3 \\ 1 & 3 & 3 \end{bmatrix}.$$  

We obtain a $GL_4 \times GL_3$ bicrystal structure on the pair $(T, \hat{T})$ by the induced $GL_3$ structure on $T$, and the $GL_4$ crystal structure on $\hat{T}$.

Assuming facts stated for the other models above, we can prove the bicrystal structures on ptableaux commute. This follows from the easily proved bijection $\Phi$ from ptableaux to matrices: given $T$, the $(i, j)$ entry of the associated matrix $M$ is the number of times an $i$ appears in row $j$ of $T$. For our example,

$$T = \begin{bmatrix} 1 & 3 & 4 \\ 1 & 2 & 2 \\ 3 & 3 & 4 & 4 \end{bmatrix} \quad \Phi \quad M = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 1 & 0 & 2 \\ 1 & 0 & 2 \end{bmatrix}.$$  

It is not hard to show that if, under this bijection, the ptableau $T$ corresponds to matrix $M$, then the ptableau $\hat{T}$ corresponds to $M^T$, the transpose of $M$.

$$T = \begin{bmatrix} 1 & 3 & 4 \\ 1 & 2 & 2 \\ 3 & 3 & 4 & 4 \end{bmatrix} \quad \Rightarrow \quad \hat{T} = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 \\ 1 & 3 & 3 \\ 1 & 3 & 3 \end{bmatrix}.$$  

Indeed, we can recover $\hat{T}$ even more directly from the parsed word $\omega = 21|22|331|331$ that generated all our examples. The content of row $i$ of $\hat{T}$ is merely the $i$-th factor of $\omega_P$, read right to left. This content is then arranged minimally to ensure the resulting tableau filling is column strict.

A few comments are in order here. First, these observations emphasize that a perforated tableau $T$ associated to parsed word $\omega_P$ is not merely a rearrangement of the factors of $\omega_P$, although its dual $\hat{T}$, is. This is significant for a number of reasons. The number of rows of $\hat{T}$ depends on the number of factors into which the associated word $\omega$ is parsed. That is, there is an intrinsic $GL_n$ crystal structure on words $\omega \in [n]^{\otimes k}$, but the $GL_\ell \times GL_n$ bicrystal structure depends on the choice of parsing in the case of $\omega_P$ or, equivalently, on the choice of compatible top row $\tau$ in the associated biword. One obtains different bicrystal structures, though all of them yield isomorphic $GL_n$ structures.

We will see that one consequence of this is that if $T$ is the ptableau associated to a parsed word $\omega_P$, and $T'$ is the ptableau associated to $\omega_P'$ (the same word, but with a different parsing), then $T$ and $T'$ will be plactly equivalent, meaning that they lie in the same

---

The Springer logo is visible at the bottom right corner of the page.
location in isomorphic irreducible crystal graphs, and (by means of our Theorem 6.2 below) their respective highest weight elements will be partitions of the same shape. This need not be true of their duals \( \hat{T} \) and \( \hat{T}' \), which might not even have the same number of rows, despite both being more directly determined by the word \( \omega \).

We should mention that in recently available work of Gerber and Lecouvey [8] (which appeared subsequent to the authors’ earlier draft of this work [1]), a crystal model consisting of a tensor product of column tableaux is proposed. They define a crystal operator from a word in \([n]^{\otimes k}\) computed from their model, and give many other significant results, including a definition of duality that allows them to compute the maps mentioned above. Most of these results are outside the scope of our work here.

2 Definitions, Perforated Tableaux

In all that follows our Young diagrams (or just diagrams) will be rectangular. A tableau is a filling of boxes of the diagram with positive integers, possibly leaving some boxes blank (or perforated). We refer to boxes in a tableau that are not blank as having content.

**Definition 2.1** Given a box \([b]\) in a tableau \(T\), the northwest shadow of \([b]\) is the collection of boxes in \(T\) to the west, north, or both, of \([b]\).

**Definition 2.2** In a tableau, a collection of boxes forms a horizontal strip if no two boxes in the strip are in the same column, and any box in row \(k\) of the strip is in a column to the right of any box of the strip in row \(k + \ell\), for \(\ell > 0\). Given a horizontal strip, we call the lowest, left-most box the head of the strip, and the highest, rightmost box the tail. If the collection of all the appearances of an integer \(i\) in a tableau form a horizontal strip, we call this the \(i\)-strip of the tableau.

In this paper, we picture horizontal strips as if they “grow” from the lower left to right, bottom to top (from head to tail). This will seem natural when we introduce a map from words in \([n]^{\otimes k}\) to perforated tableaux, in which strips are filled from head to tail as the word is read in (via our map) from left to right.

**Definition 2.3** A rectangular tableau \(T\) will be called a perforated tableau (or ptableau, for short) if:

1. All boxes in \(T\) contain positive integers (have content), or are blank (unfilled, denoted \(\square\)).
2. For any positive integer \(i\), the \(i\)’s in \(T\) form a horizontal strip.
3. For positive integers \(i\) and \(j\), with \(i < j\), every element in the horizontal strip of \(j\)’s lies outside the northwest shadow of any entry in the horizontal strip of \(i\)’s.
4. \(T\) has no column entirely composed of blanks (rows of all blanks are allowed).

If, in addition to the above requirements, a perforated tableaux \(T\) satisfies:

5. For any positive integer \(i\) appearing in \(T\), the head of the \(i\)-strip of \(T\) lies in a row strictly below the row containing the tail of \((i - 1)\)-strip, then we say \(T\) is a minimally parsed perforated tableau.
Here is an example of a minimally parsed perforated tableau:

\[ T = \begin{array}{cccc}
\hline
& 1 & 1 & 1 \\
1 & 1 & 1 & 2 \\
1 & 1 & 2 & 2 & 3 & 3 & 3 & 3 \\
1 & 1 & 1 & 2 & 2 & 2 & 3 & 4 & 4 \\
\hline
\end{array} \]

with \( \text{wt}(T) = (3, 5, 7, 10) \).

**Notational Convention** We need to distinguish between an actual box of a ptableau, and the *value* (an integer or blank) of that box. We denote a box with content \( a \) by \([a]\); we denote a blank (empty) box by □.

**Definition 2.4** We define the *weight* of a perforated tableau \( T \) to be the sequence \( \text{wt}(T) = (m_1, \ldots, m_n) \), where \( m_i \) is the number of filled boxes in row \( i \) of \( T \).

**Definition 2.5** Let \( \nu = (\nu_1, \ldots, \nu_n) \) be a partition (so \( \nu_1 \geq \nu_2 \geq \cdots \geq \nu_n \)). Let \( \text{SSYT}(\nu) \) denote the set of semistandard Young tableaux of shape \( \nu \). All tableaux in \( \text{SSYT}(\nu) \) will have \( \nu_1 \) columns, with content in every box of row 1. In row \( i \), for \( 2 \leq i \leq n \), there is content in the first \( \nu_i \) boxes, followed by \( (\nu_1 - \nu_i) \) blanks.

So, an example of a semistandard Young tableau of weight \((4, 3, 3, 1)\) is:

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

while one of weight \((4, 3, 3, 1, 0)\) is:

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

Note that, typically, a SSYT is drawn without including the blank boxes, but, by including the blank boxes, we see that SSYT are a special case of perforated tableaux.

**Definition 2.6** Suppose \( S \) and \( T \) are tableaux with \( n \) rows. We say \( S \) and \( T \) are *row equivalent* if we can transform \( S \) into \( T \) by a finite number of steps in which a blank in a given row is swapped with an adjacent non-blank in the same row, such that the result is a valid perforated tableau. Note that if \( S \) and \( T \) are row equivalent, then \( \text{wt}(S) = \text{wt}(T) \).

Row equivalence is clearly an equivalence relation on tableaux.

**Definition 2.7** A ptableau \( T \) is *left-justified* if all of \( T \)'s content is as far to the *left* as possible. That is, in any ptableau \( T' \) that is row equivalent to \( T \), each entry of \( T \) is weakly to the left of the corresponding entry in \( T' \).

Similarly, a ptableau \( T \) is *right-justified* if all content of \( T \) is as far right as possible.

The following are then easily proved:

**Lemma 2.8** 1. *Given any ptableau \( T \), there is a unique ptableau \( ^*T \) that is left-justified and row equivalent to \( T \).*

2. *Given a left-justified ptableau \( ^*T \), there is a unique right-justified ptableau \( T^* \) that is row equivalent to \( ^*T \).*

**Definition 2.9** Let \( \text{PTab} \) denote the set of row equivalence classes of ptableaux, each of which contains ptableaux that are row-equivalent to the representative \( ^*T \) and that have the
same number of columns as $^*T$. Let $\text{PTab}_n$ denote the subset of $\text{PTab}$ composed of ptableaux with $n$ rows (some of those rows may be entirely blank).

Let $\text{PTab}_{(\ell,n)}$ denote the subset of $\text{PTab}_n$ whose content is from the set $[\ell]$. We require that all members of an equivalence class have the same number of columns to rule out such examples as:

$$ T = \begin{array}{ccc|ccc}
1 & 2 & 3 & 1 & 1 & 2 \\
\end{array} $$

Strictly speaking, this is a perforated tableau, but it would seemingly be row-equivalent to:

$$ \begin{array}{ccc|ccc}
\hline
1 & 1 & 1 & 2 \\
1 & 2 & 2 & 3 \\
\hline
\end{array} $$

were we to preserve the blanks in $T$. But this is not a perforated tableau because by definition, ptableaux cannot have columns of blanks. We resolve this by using the left-justified form of $T$, with blank columns removed:

$$ ^*T = \begin{array}{ccc|ccc}
1 & 1 & 1 & 2 \\
1 & 2 & 2 & 3 \\
\end{array} $$

and then only admit into the equivalence class of $^*T$ those perforated tableaux with the same number of columns (in this case, 5) as $^*T$.

Note that, by Lemma 2.8, the right-justified ptableau

$$ T^* = \begin{array}{ccc|ccc}
1 & 1 & 1 & 2 \\
1 & 2 & 2 & 3 \\
\end{array} $$

is in the same $\text{PTab}$ equivalence class as $^*T$, but that this class would not include the tableau:

$$ \begin{array}{ccc|ccc}
1 & 1 & 1 & 2 \\
1 & 2 & 2 & 3 \\
\end{array} $$

because it has more columns than $^*T$.

We regard row-equivalence classes of ptableaux, under Definition 2.9, as a single object, which (to avoid unnecessary circumlocution) we will still refer to as “a” ptableau. We think of this as a single tiled grid (with the number of columns defined by $^*T$) in which the tiles (the content) are free to slide right or left, subject to the constraint of column strictness. Thus, we regard

$$ \begin{array}{ccc|ccc}
1 & 1 & 4 \\
1 & 1 & 2 & 5 \\
1 & 2 & 3 & 4 & 5 & 6 \\
\end{array} $$

and

$$ \begin{array}{ccc|ccc}
1 & 1 & 4 \\
1 & 1 & 2 & 5 \\
1 & 2 & 3 & 4 & 5 & 6 \\
\end{array} $$

to be equal as ptableaux since they are row equivalent. Furthermore,

$$ \begin{array}{ccc|ccc}
1 & 1 & 4 \\
1 & 1 & 2 & 5 \\
1 & 2 & 3 & 4 & 5 & 6 \\
\end{array} $$

is the (unique) left-justified representative of this equivalence class. So, while a ptableau $T$ can potentially be represented in multiple (row-equivalent) forms, $^*T$ and $T^*$ are unique representatives.

## 3 From Words $[n] \otimes k$ and Biwords to Perforated Tableaux

**Definition 3.1** Given a word $\omega \in [n] \otimes k$, a **subword** of $\omega$ is sequence of letters in $\omega$, in order but not necessarily contiguous. A factor of $\omega$ is a contiguous subword of $\omega$. 
Below we define a map $\text{Perf}$ from words $[n]^{\otimes k}$ to perforated tableaux. The map will be determined by a word $\omega \in [n]^{\otimes k}$, and a “parsing” of $\omega$ into weakly decreasing factors. Condition (5) of Definition 2.3 above characterizes the form of the ptableau $\text{Perf}(\omega)$ when $\omega$ has been parsed into the minimal number of such factors, though condition (5) can be verified independently.

**Definition 3.2**

Given some word $\omega = \omega_1 \omega_2 \cdots \omega_{k-1} \omega_k \in [n]^{\otimes k}$, a parsing of the word is any partition of $\omega$ into a disjoint collection of weakly decreasing factors, when read left to right. Trivial parsings (with empty factors) are allowed. The minimal parsing of $\omega$ is the (unique) parsing of $\omega$ into the fewest possible number of weakly decreasing factors.

For example, if $\omega$ is

$$44334422334411122233444 \in [n]^{\otimes 24},$$

we parse it into weakly decreasing factors by denoting the breaks between factors with the symbol “|”:

$$4433|4422333|444111|222233||444.$$

(the “||” above denotes an empty factor). The minimal parsing is:

$$4433|4422|333|444111|222233|444.$$

For future reference, we record the following:

**Definition 3.3**

Given some word $\omega \in [n]^{\otimes k}$, we denote a parsing of $\omega$ by the symbol $P$, and denote the word $\omega$, with its associated parsing, by $\omega_P$. The minimal parsing of a word $\omega$ is denoted $\omega_{P_{\min}}$.

We let $P([n]^{\otimes k})$ denote the set of all parsings of words in $[n]^{\otimes k}$.

If a parsed word $\omega_P \in P([n]^{\otimes k})$, breaks $\omega$ into $\ell$-many weakly decreasing factors, let $\mathcal{B}_i(\omega_P)$ denote the unique biword in $\mathcal{B}_i([\ell]^{\otimes k}, [n]^{\otimes k})$ obtained by writing a string of $i$’s over the $i$-th factor.

Thus, our example

$$\omega_P = 4433|4422|333|444111|222233||444$$

would be associated to the biword

$$\mathcal{B}_i(\omega_P) = \left(\begin{array}{cccccccccccccccccccc}
1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 5 & 5 & 5 & 5 & 6 & 6 & 6 & 6 & 7 & 7 & 7 & 9 & 9 & 9 & 9 \\
4 & 4 & 3 & 3 & 4 & 4 & 2 & 2 & 2 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 4 & 4 & 4 & 4 & 4
\end{array}\right).$$

Consequently, we have the remarkable fact that parsed words carry the same information (in the context of models for crystal operators) as biwords, RSK $(P, Q)$ pairs, and matrix models $\text{Mat}_{\ell \times n}$. Below, we define a map from parsed words to ptableaux. It will be obvious that we could have defined them from biwords directly, but it will be both notationally convenient to use parsed words, and it will allow us to state with precision how the crystal structure we define on ptableaux commutes with this map, using the crystal operators on words (and ignoring the parsing).

We define the map $\text{Perf} : P([n]^{\otimes k}) \to \text{PTab}_n$ taking a parsed word $\omega_P \in P([n]^{\otimes k})$ to an equivalence class $\text{Perf}(\omega_P) = T \in \text{PTab}_n$ determined by its unique left-justified perforated tableau $^*T$, constructed from $\omega$ and its parsing $P$. It will be apparent that $^*T$ clearly fulfills the requirements of Definition 2.3, so we omit the proof of this claim.
Given a word \( \omega P = \omega_1 \omega_2 \cdots \omega_{k-1} \omega_k \in P([n]^{\otimes k}) \), we represent the parsing of \( \omega \) by:

\[
\omega P = h_1|h_2| \cdots |h_\ell,
\]

where each \( h_s \) is a weakly decreasing factor of \( \omega P \). Given a factor \( h_s \), let \( h^i_s \) denote the number of \( i \)'s in \( h_s \).

We construct from \( \omega P \) a left-justified ptableau \( ^*T \): Begin with a blank tableau of \( n \) rows (and an as-yet unspecified number of columns), with the left-most column designated as column 1, the next column as column 2, etc. Given the factor \( h_1 \), we put \( h^n_1 \)-many 1’s in row \( n \), starting in column 1, then \( h^{n-1}_1 \)-many 1’s in row \( n-1 \), starting in column \( h^n_1 + 1 \), then \( h^{n-2}_1 \)-many 1’s in row \( n-2 \), starting in column \( h^n_1 + h^{n-1}_1 + 1 \), etc., until we have constructed a horizontal strip of 1’s. We then put \( h^k_2 \)-many 2’s in row \( k \), starting from row \( k = n \) and moving up, subject to the constraint that each row of 2’s is as far left as possible, and the filling is column-strict (leaving blanks in the tableau as needed to achieve this). We continue in the same manner for \( h_3 \) through \( h_\ell \). The rightmost column containing an integer is then the rightmost column of the perforated tableau. We denote the output of this construction by \( ^*T \) since it is clearly left-justified. The ptableau \( ^*T \) determines, by definition, a row-equivalence class \( \text{Perf}(\omega P) \in \text{PTab} \).

As an example, let

\[
\omega = 44433222111444333221443344 \in \{4\}^{26}.
\]

The minimal parsing of \( \omega \) is

\[
\omega_{P_{\text{min}}} = \underbrace{44433222111}_{h_1} | \underbrace{444333221}_{h_2} | \underbrace{4433}_{h_3} | \underbrace{44}_{h_4}.
\]

To construct \( ^*T \in \text{Perf}(\omega_{P_{\text{min}}}) \), we first construct the horizontal strip of 1’s determined by \( h_1 = 44433222111 \). Working left to right in \( h_1 \), we place three 1’s in row 4, then two 1’s in row 3, three 1’s in row 2, and finally three 1’s in row 1:

\[
\begin{array}{|c|c|c|}
\hline
1 & 1 & 1 \\
\hline
1 & 1 \\
\hline
1 & 1 & 1 \\
\hline
\end{array}
\]

We then insert a maximally left-justified horizontal strip of 2’s determined by \( h_2 \). Notice that this increases the number of columns in the ptableau.

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

We fill in the rest of the perforated tableau using \( h_3 \) and then \( h_4 \), giving the left-justified \( ^*T \in \text{Perf}(\omega_{P_{\text{min}}}) \) where:

\[
^*T = \begin{array}{|c|c|c|c|}
\hline
1 & 1 & 1 & 2 \\
\hline
1 & 1 & 2 & 2 & 3 & 3 \\
\hline
1 & 1 & 2 & 2 & 3 & 3 \\
\hline
\end{array}
\]

In terms of a biword (induced by some parsed word \( \omega P \)):

\[
\begin{pmatrix}
a_1 & a_2 & \cdots & a_k \\
\omega_1 & \omega_2 & \cdots & \omega_k
\end{pmatrix},
\]

we obtain \( \text{Perf}(\omega P) \) from this biword by placing each entry \( a_i \) in the top row of the biword into row \( \omega_i \) of the ptableau, moving content to the right, as necessary, to avoid column strictness violations.
Definition 3.4 Define $BiPerf : Bi([\ell]^{\otimes k}, [n]^{\otimes k}) \to PTab_n$ via

$$BiPerf (\frac{\tau}{\omega}) = Perf (\omega_P),$$

where

$$\omega_P = Bi^{-1}(\frac{\tau}{\omega}) = Perf (\omega_P),$$

and $Bi$ is defined as in Definition 3.3.

Lemma 3.5 Let Perf and $BiPerf$ be the maps defined above. Then recalling Definition 2.9:

1. $BiPerf$ is a bijection from biwords $\bigcup_{k,\ell \geq 1} Bi([\ell]^{\otimes k}, [n]^{\otimes k})$ to equivalence classes in $PTab_n$, where, in particular, $BiPerf$ maps elements in $Bi([\ell]^{\otimes k}, [n]^{\otimes k})$ to $PTab(\ell, n)$.
2. Regarding each element $\omega \in [n]^{\otimes k}$ as a minimally parsed element of $P([n]^{\otimes k})$, then $Perf$ is a bijection from $\bigcup_{k \geq 1} [n]^{\otimes k}$ to row equivalence classes of minimally parsed ptableaux in $PTab_n$.
3. The image $\omega_P \mapsto T = Perf (\omega_P)$ determines a bijection from letters in $\omega$ to the content of the tableau $T$, because row equivalence fixes the content and order of entries in any row. We also denote this bijection as “Perf”.
4. If a letter $\omega_i$ lies to the right of a letter $\omega_j$ in a word $\omega_P \in [n]^{\otimes k}$, then $Perf (\omega_j)$ lies outside the northwest shadow of $Perf (\omega_i)$ in $T$.

Given the bijections between biwords, elements of Mat$_{\ell \times n}$ and RSK $(P, Q)$ pairs, we could extend bijections between ptableaux and these objects as well.

We will need the following technical result in the next section:

Lemma 3.6 Let $T \in Perf (\omega_P)$ for some $\omega_P \in P([n]^{\otimes k})$. Then the number of columns in $^*T$ equals the length of the longest weakly decreasing subword in $\omega_P$.

Proof Let $m$ equal the length of the maximal weakly decreasing subword of $\omega_P$ (in this setting, the parsing is irrelevant). Recall the bijection between the entries in $\omega_P$ and the content of $^*T \in Perf (\omega_P)$ given in Lemma 3.5. Any weakly decreasing subword of $\omega_P$ will correspond, under this bijection, to distinct boxes of content in $^*T$ appearing in different columns. Hence the number of columns of $^*T$ is at least $m$. Clearly, the number of columns cannot exceed $m$; if it did, there would be a sequence of entries in $^*T$ appearing in distinct columns that would correspond to a weakly decreasing subword in $\omega_P$ of length greater than $m$, contradicting $m$’s maximality.

4 Crystal Structures on $P([n]^{\otimes k})$ and PTab$_n$

4.1 Crystal Operators on $P([n]^{\otimes k})$

We briefly review the crystal graph structure on $[n]^{\otimes k}$ (the parsings are irrelevant to these considerations), following the Kashiwara convention on crystal operators. See [5] Lemma 2.33, p. 21, (defined there with an alternate orientation from ours):
Proposition 4.1 Suppose $\omega = \omega_1 \omega_2 \cdots \omega_k \in [n]^{\otimes k}$. For any $s \in [n]$, define $\phi_i(s) = \delta_i(s)$ (Kronecker delta function), and $\epsilon_i(s) = \delta_i(s) + 1$. Then define
\[
c_i^e(\omega, j) = \left( \sum_{s=1}^{j} \epsilon_i(\omega_s) - \sum_{s=1}^{j-1} \phi_i(\omega_s) \right),
\]
\[
c_i^f(\omega, j) = \left( \sum_{s=j}^{k} \phi_i(\omega_s) - \sum_{s=j+1}^{k} \epsilon_i(\omega_s) \right),
\]
and, for $1 \leq i < n$,
\[
\epsilon_i(\omega) = \max_{j=1}^{k} c_i^e(\omega, j), \quad \phi_i(\omega) = \max_{j=1}^{k} c_i^f(\omega, j).
\]
Then for $s \in [n]$, define $e_i(s) = i$ if $s = i + 1$, and NULL otherwise, and $f_i(s) = i + 1$ if $s = i$, and NULL otherwise. Extend these definitions to the word $\omega \in [n]^{\otimes k}$ by setting
\[
e_i(\omega) = \omega_1 \omega_2 \cdots \omega_{j-1} e_i(\omega_j) \omega_{j+1} \cdots \omega_k,
\]
with $j$ the smallest index at which $\epsilon_i(\omega) = c_i^e(\omega, j)$, and
\[
f_i(\omega) = \omega_1 \omega_2 \cdots \omega_{j-1} f_i(\omega_j) \omega_{j+1} \cdots \omega_k,
\]
with $j$ the largest index at which $\phi_i(\omega) = c_i^f(\omega, j)$, or NULL if $e_i(\omega_j)$ or $f_i(\omega_j)$ is NULL.

These definitions determine an $A_{n-1}$ crystal structure on $[n]^{\otimes k}$, as a $k$-fold tensor product of the standard crystal of $[n]$.

Definition 4.2 A crystal graph on $[n]^{\otimes k}$ is a subset $B \subseteq [n]^{\otimes k}$ that is closed under the crystal operators $e_i, f_i$, for $1 \leq i < n$. We regard the words in $B$ as the nodes of the graph; two words $w, w' \in B$ are connected by an edge if $w' = e_i(w)$ for some $i$ or, equivalently, if $w = f_i(w')$. A connected crystal graph is called irreducible.

Definition 4.3 A word $\omega \in [n]^{\otimes k}$ is highest weight if and only if $e_i(\omega) = $ NULL for all $i$, $1 \leq i < n$. Likewise, a word $\omega' \in [n]^{\otimes k}$ is lowest weight if and only if $f_i(\omega') = $ NULL for all $i$, $1 \leq i < n$.

It is known that if $\omega \in [n]^{\otimes k}$ is highest weight, then its weight, $wt(\omega)$, is a partition. (Recall that the weight of a word $\omega$ is the composition $(a_1, a_2, \ldots, a_n)$, where $a_i$ is the number of $i$’s in $\omega$.) Every irreducible crystal graph possesses a unique element of highest weight ([5], p. 34). Thus we need to distinguish between a highest weight element of an irreducible crystal graph, and the actual weight of the highest weight element. This leads to the following:

Definition 4.4 Suppose $\omega \in [n]^{\otimes k}$ is highest weight. We will denote the connected component containing $\omega$ (an irreducible crystal graph) by $B^v(\omega)$, where $v = wt(\omega)$.

If $\omega, \omega' \in [n]^{\otimes k}$ are highest weight words such that $wt(\omega) = wt(\omega') = v$, then the irreducible crystal graphs $B^v(\omega)$ and $B^v(\omega')$ are necessarily isomorphic (see [5], p. 52). We refer to words identified by means of such an isomorphism as plactically equivalent.
4.2 Crystal Operators on Perforated Tableaux

**Definition 4.5** Let \( T \in \text{PTab}_n \) and choose \( i \), for \( 1 \leq i < n \). Let \( T[i, i + 1] \) denote the ptableau formed by the entries of \( T \) in rows \( i \) and \( i + 1 \).

Note that, as columns of blanks are not allowed in a ptableau, we omit any such columns in the determination of \( T[i, i + 1] \). For example, if

\[
T = \begin{array}{|c|c|c|}
1 & 1 & 2 \\
1 & 1 & 2 \\
1 & 1 & 2 \\
\end{array}
\quad \text{then} \quad T[2, 3] = \begin{array}{|c|c|c|}
1 & 1 & 2 \\
1 & 1 & 2 \\
1 & 1 & 2 \\
\end{array}
\]

**Lemma 4.6** Let \( T \in \text{PTab}_n \). If content in row \((i + 1)\) of \( \ast T[i, i + 1] \) is uncovered, then the corresponding content in \( \ast T \) is also uncovered.

**Proof** Replace rows \( i \) and \( i + 1 \) in \( \ast T \) with \( \ast T[i, i + 1] \), with its content as far left as possible.

For example:

\[
\begin{array}{|c|c|c|}
1 & 1 & 2 \\
1 & 1 & 2 \\
1 & 1 & 2 \\
\end{array}
\quad \text{→} \quad \begin{array}{|c|c|c|}
1 & 1 & 2 \\
1 & 1 & 2 \\
1 & 1 & 2 \\
\end{array}
\]

The resulting tableau will typically no longer be a valid ptableau, but we shall sequentially adjust content in it to resolve this in such a way that (1) the end result is \( \ast T \), and (2) content uncovered in \( \ast T[i, i + 1] \) will remain uncovered during the adjustment. Note that we have not changed the content, nor the order of the non-blank entries in rows \( i \) and \( i + 1 \) of \( \ast T \), and have only changed their relative positions in the rows.

Begin with the 1’s in row \( i + 1 \). If the rightmost \( \boxed{1} \) of \( \ast T \) appearing below row \( i + 1 \) is in column \( \ell \), then move all content of rows \( i \) and \( i + 1 \) to the right, so that the first entry in row \( i + 1 \) is in column \( \ell + 1 \). In this example, \( \ell = 2 \) and we move all content in rows 2 and 3 two columns to the right:

\[
\begin{array}{|c|c|c|}
1 & 1 & 2 \\
1 & 1 & 2 \\
1 & 1 & 2 \\
\end{array}
\quad \Rightarrow \quad \begin{array}{|c|c|c|}
1 & 1 & 2 \\
1 & 1 & 2 \\
1 & 1 & 2 \\
\end{array}
\]

Call this the “1-adjustment”. If, after this move, the rightmost \( \boxed{2} \) appearing below row \( i + 1 \) appears in column \( \ell' \), then we move all content in rows \( i \) and \( i + 1 \) with values greater than 1 to the right until the resulting (partially adjusted) ptableau is column-strict and left-justified in columns 1 through \( \ell' \). Call this shift the “2-adjustment”. (In our example, the 2-adjustment isn’t necessary as the 2’s in rows 3 and 4 are already column-strict.) We continue inductively until we have adjusted all content from its original location in \( \ast T[i, i + 1] \) to where it must appear in \( \ast T \), in order to recreate column-strictness, maintaining the requirement that if an \( s \)-adjustment moves content in rows \( i \) and \( i + 1 \), it moves all content with values greater than or equal to \( s \) an equal number of columns to the right. In our example, performing the 4-adjustment moves all content greater than or equal to 4 in rows 2 and 3 two columns to the right and results in \( \ast T \):

\[
\begin{array}{|c|c|c|}
1 & 1 & 2 \\
1 & 1 & 2 \\
1 & 1 & 2 \\
\end{array}
\quad \text{4}-\text{adjustment:} \quad \Rightarrow \quad \begin{array}{|c|c|c|}
1 & 1 & 2 \\
1 & 1 & 2 \\
1 & 1 & 2 \\
\end{array}
\]
Now let \( \boxed{c} \) denote an uncovered entry in row \( (i + 1) \) of \( *T[i, i + 1] \). All content in row \( i \) in a column strictly left of the column containing \( \boxed{c} \) has value smaller than \( c \), and content in row \( i \) to the right has value greater than or equal to \( c \); otherwise, we could have covered \( \boxed{c} \) in \( *T[i, i + 1] \). However, by the process noted above, at each stage as we adjust \( *T \) from \( *T[i, i + 1] \), for \( s \leq c \), the \( s \)-adjustment moves all content with values greater than or equal to \( s \) the same number of entries to the right. In particular, if \( \boxed{c} \) is uncovered before such an \( s \)-adjustment, it remains uncovered after it. For any \( s > c \), the \( s \)-adjustment will not move \( \boxed{c} \) at all. Thus, no content in row \( i \) will be moved over \( \boxed{c} \) in \( *T \), and the lemma is proved.

**Definition 4.7** Suppose \( T \in \text{PTab}_n \) and that \( \boxed{c} \) is an entry in row \( i + 1 \) of \( T \) such that in \( *T[i, i + 1] \), \( \boxed{c} \) is the rightmost entry with a blank above it. In this case, we define \( e_i(T) \) to be the ptableau obtained by swapping \( \boxed{c} \) with the blank above it (in row \( i \)) of \( *T \) (by virtue of Lemma 4.6). If there is no such entry \( \boxed{c} \) in \( *T[i, i + 1] \), then \( e_i(T) = \text{NULL} \).

We define \( \epsilon_i(T) \) to be the number of blanks in row \( i \) (the top row) of \( *T[i, i + 1] \).

We give an analogous definition of operators \( f_i : \text{PTab}_n \rightarrow \text{PTab}_n \cup \{\text{NULL}\} \) below.

Recall the definition of the dual \( \hat{T} \) of a ptableau \( T \) given in the introduction. There are related results \([8, 17]\) that use the bicrystal structure on a crystal graph model to realize a crystal operator on one object as some other combinatorially defined transformation on its dual. This includes using \textit{jeu de taquin} operations on dual elements, to compute commutators of tensor products of factors of words. That is not (quite) what is asserted here. Rather, in the case of crystal operators \( e_i \) and \( f_i \) on ptableaux, the operators themselves can be seen as a sort of \textit{jeu de taquin} operation on the ptableau \( T \). This may very well correspond to an interesting operation on the dual \( \hat{T} \). However, those previously cited results do not imply our theorems here, and second, our goal is to replace the more complicated definitions of crystal operators (on words, matrices, SSYT) with the simpler operators defined on ptableaux.

Computing these (to be shown) crystal operators on ptableaux generally requires less computation than computing them on words. Computation of the crystal operators \( f_i, e_i \) on words in \([n]^{\otimes k}\) requires counting \( i \)'s and \((i + 1) \)'s across the entire length of the word (computing in the other models is not substantially simpler), while the computation of operators on ptableaux requires working only with the columns of rows \( i \) and \( i + 1 \) in \( T \) that contain a blank in row \( i \).

### 4.3 Examples

As an example, consider \( T \) and \( T[2, 3] \) as above.

We calculate \( e_2(T) \):

\[
*T[2, 3] = \begin{array}{cccccc}
1 & 1 & 2 & 6 \\
1 & 1 & 2 & 4 & 5 & 6 & 7 \\
\end{array}
\]

and so the 6 in the sixth column of \( *T[2, 3] \) is the rightmost entry under a blank. Noting Lemma 4.6 (ensuring the blank persists in \( *T \)), we circle this entry in \( *T \) below:

\[
*T = \begin{array}{ccccc}
1 & 1 & 2 & 4 & 5 \\
1 & 1 & 2 & 4 & 5 & 6 & 7 \\
1 & 1 & 2 & 3 & 3 & 4 & 6 & 6 \\
\end{array}, \quad \text{and so} \quad e_2(T) = \begin{array}{cccccc}
1 & 1 & 2 & 4 & 5 \\
1 & 1 & 2 & 4 & 5 & 6 & 7 \\
1 & 1 & 2 & 3 & 3 & 4 & 6 & 6 \\
\end{array}.
\]

By definition, \( e_2(T) = 3 \) because there are three blanks in the top row of \( T[2, 3] \).
Thus, $T \in \text{Perf}(\omega)$ for the word: $\omega_P = 443322|432|444|4311|31|4432|3$.

Anticipating Theorem 5.1 below, we note: $e_2(\text{Perf}(\omega_P)) = \text{Perf}(e_2(\omega_P))$.

The definitions of crystal operators on ptableaux make many features of the theory of crystal graphs more visually clear. For example, the form of the 1-root string (see [5], p.23),

$$322|3311|222|3 \overset{e_1}{\rightarrow} 322|3311|221|3 \overset{e_1}{\rightarrow} 321|3311|221|3 \overset{e_1}{\rightarrow} 311|3311|221|3$$

is computable, but not perhaps obvious. However, represented as perforated tableaux, the result of successive applications of $e_1$ is easy to see:

$$*T = \begin{array}{ccc}
1 & 1 & 1 \\
1 & 3 & 3 \\
1 & 2 & 4 \\
\end{array} \overset{e_1}{\rightarrow} \begin{array}{ccc}
1 & 1 & 3 \\
1 & 3 & 3 \\
1 & 2 & 4 \\
\end{array} \overset{e_1}{\rightarrow} \begin{array}{ccc}
1 & 2 & 2 \\
1 & 3 & 3 \\
1 & 2 & 4 \\
\end{array} \overset{e_1}{\rightarrow} \begin{array}{ccc}
1 & 1 & 2 \\
1 & 2 & 2 \\
1 & 2 & 4 \\
\end{array},$$

and the action of $e_1$ becomes even more evident when restricting our attention to $*T[1, 2]$, where applying $e_1$ just moves left-justified content under a blank up:

$$*T[1, 2] = \begin{array}{ccc}
2 & 2 & 2 \\
1 & 3 & 3 \\
1 & 3 & 3 \\
\end{array} \overset{e_1}{\rightarrow} \begin{array}{ccc}
2 & 2 & 2 \\
1 & 3 & 3 \\
1 & 3 & 3 \\
\end{array} \overset{e_1}{\rightarrow} \begin{array}{ccc}
2 & 2 & 2 \\
1 & 3 & 3 \\
1 & 3 & 3 \\
\end{array} \overset{e_1}{\rightarrow} \begin{array}{ccc}
1 & 2 & 2 \\
1 & 3 & 3 \\
1 & 3 & 3 \\
\end{array}.$$

Below are, we claim, two (isomorphic) crystal graphs of ptableaux. The crystal on the left is the image under Perf of a crystal in $[3]^{\otimes 4}$ with highest weight $1112$. This crystal can be realized explicitly as a crystal of SSYT under a row-reading map. However, the crystal on the right corresponds to highest weight $1211 \in [3]^{\otimes 4}$, which does not have a corresponding SSYT model.
5 First Results

The following theorem will imply that the bijections mentioned in the introduction, and proved in Lemma 3.5 along with the definitions for crystal operators on ptableaux of the previous section, induce a $GL_n$ crystal graph structure on ptableaux. While one could simply define crystal operators on ptableaux by transfer of structure [5, 17], we would still need to verify that this transfer is actually realized by the simpler jeu de taquin style maps constructed above, saving ourselves little work, and so we present the theorem below.

**Theorem 5.1** Let $e_i : P([n]^{\otimes k}) \to P([n]^{\otimes k}) \cup \{\text{NULL}\}$ be the standard crystal operator and let $\omega_P \in P([n]^{\otimes k})$, with some parsing $\mathcal{P}$. Then,

$$e_i (\text{Perf}(\omega_P)) = \text{Perf}(e_i(\omega_P)).$$

That is, the standard $e_i$ crystal operator on words in $[n]^{\otimes k}$ commutes with the $e_i$ operator on ptableaux (Definition 4.7).

The above equality also implies that if $\omega_P$ is some parsing of a word $\omega \in [n]^{\otimes k}$, then $\mathcal{P}$ remains a valid parsing of $e_i(\omega)$, so that we can write $e_i(\omega_P) = e_i(\omega)\mathcal{P}$.

The effect of $e_i$ on some word $\omega_P$ (if the output is not NULL), is to change a letter $(i + 1)$ in $\omega_P$ into an $i$ in $e_i(\omega_P)$. Thus, the difference between Perf$(\omega_P)$ and Perf$(e_i(\omega_P))$ is that exactly one entry in row $i + 1$ of Perf$(\omega_P)$ is moved to row $i$ in Perf$(e_i(\omega_P))$. Theorem 5.1 says that our definition of $e_i$ on ptableaux chooses the correct entry to move, by swapping it with a blank lying directly above it.

Before proving the theorem, we give an example. Given some parsed word $\omega_P$, let $\omega_P[i, i + 1]$ denote the sequence of $i$’s and $(i + 1)$’s appearing in $\omega_P$, in order, maintaining the parsing. So, suppose, for some word, we had:

$$\omega_P[3, 4] = 44433|444333|4433|44443|433.$$

Recall the definition of $c_e^i(\omega, j)$ Theorem 4.1. We write the running totals of $c_e^3(\omega_P, j)$ (from left to right) under the entries in $\omega_P$, putting the occurrence of any new maximum in bold:

$$\omega_P[3, 4] = \textbf{44433}3|\textbf{444333}|\textbf{433}|\textbf{44443}|\textbf{443}$$

$$\textbf{12332}|23\textbf{4432}|23\textbf{32}|23\textbf{455}|566$$

Let $T = \text{Perf}(\omega_P)$. $T$ is constructed, under Perf, by reading the letters of $\omega_P$, left to right. We say a box $\boxed{a}$ in $T$ is “read into” the tableau “before” some other box $\boxed{b}$ whenever $\text{Perf}^{-1}(\boxed{a})$ lies to the left of $\text{Perf}^{-1}(\boxed{b})$ in $\omega_P$. We label the entries of $^*T[i, i + 1]$ with subscripts, indicating the order (from left to right) that they are read into the tableau:

$$^*T[3, 4] = \begin{array}{ccccccccc}
1 & 1 & 1 & 2 & 2 & 2 & 3 & 4 & 5 \\
1 & 2 & 3 & 2 & 3 & 1 & 2 & 1 & 1
\end{array}$$

We see that each new maximum for $c_e^3(\omega_P, j)$ corresponds, under Perf, to an entry in $^*T[3, 4]$ in row 4 with no content above it. Thus, $e_3(\omega_P)$ changes the rightmost 4 to a 3, while $e_2(^*T[3, 4])$ (and hence, $e_2(T)$ as well) moves the corresponding entry (522, in the penultimate column) from row 4 to row 3. Note also that the content to be moved is immediately apparent from the form $^*T[3, 4]$, without the computation needed in $\omega_P[3, 4]$. 

\[\text{Springer}\]
Proof Let \( \omega = \omega_1 \omega_2 \cdots \omega_{k-1} \omega_k \in [n]^\otimes k \) and choose some parsing \( \omega P = h_1 | h_2 | \cdots | h_s \). We show \( e_i(\text{Perf}(\omega P)) = \text{Perf}(e_i(\omega P)) \), and that \( P \) is still a valid parsing of \( e_i(\omega) \). Let \( T = T_{\omega P} = \text{Perf}(\omega P) \).

It is clear that \( *T[i, i + 1] = \text{Perf}(\omega P[i, i + 1]) \). Further, since \( *T[i, i + 1] \) has only two rows, the bottom row (row \( i + 1 \)), is, from left to right, a block of content followed by a block of blanks. In row \( i \), content is as far left as possible, so that, if any box \( c' \) has a blank to its left, it is to maintain column-strictness. Thus, in the diagram below, \( c \leq c' \) and \( c'' \leq c''' \) with blanks above the entries \( c \) and \( c'' \):

\[
*T[i, i + 1] = \cdots c \quad a \quad \cdots c'' \quad \cdots b \quad \cdots
\]

The bijection induced from Perf taking letters in \( \omega P \) to boxes in \( T_{\omega P} \) can be restricted to Perf : \( \omega P[i, i + 1] \to T[i, i + 1] \). Suppose \( c \) in row \( i + 1 \) of \( *T[i, i + 1] \) is an uncovered entry, meaning it has a blank directly above it. Let \( x = \text{Perf}^{-1}(c) \). Since \( *T[i, i + 1] \) is left-justified, and column-strict, the letters to the left of \( x \) in \( \omega P[i, i + 1] \) map bijectively, under Perf, to the entries left of \( c \) in rows \( i \) or \( i + 1 \) of \( *T[i, i + 1] \). Then (again recalling the definition of \( c_x^i(\omega, j) \) from Theorem 4.1):

\[
0 \leq (\text{number of entries in row } (i+1) \text{ weakly left of } c) \\
- (\text{number of entries in row } i \text{ left of } c) \\
= (\text{number of } (i+1) \text{'s weakly left of } x) - (\text{number of } i \text{'s left of } x) \\
= c_x^i(\omega P, j),
\]

where we assume \( x \) appears in column \( j \) of \( \omega P \).

Thus, uncovered entries in row \( i + 1 \) of \( *T[i, i + 1] \) correspond to positive values of \( c_x^i(\omega P, j) \), with the largest such value corresponding to the rightmost uncovered entry.

We claim only the first occurrence of a new maximum value of \( c_x^i(\omega P, j) \) (as we compute values, left to right, across the word \( \omega P \)) will correspond to an uncovered entry of \( *T[i, i + 1] \) in row \( (i+1) \). Indeed, if an entry \( b \) in row \( i + 1 \) is covered, then the entries directly above or to its left will have been read in prior to it, (although some entries that were read in before \( b \) may lie in row \( i \) to the right of \( b \)). Thus, the value of \( c_x^i(\omega P, j) \) corresponding to \( b \) either cannot exceed that of an uncovered entry to its immediate left, or is negative,(if there is no such uncovered entry). Therefore the first occurrence of the maximal value of \( c_x^i(\omega P, j) \) cannot occur at \( b \).

Thus we see that the first appearance of the maximum value of \( c_x^i(\omega P, j) \) can only appear in an entry corresponding to the rightmost uncovered entry of \( *T[i, i + 1] \) in row \( (i+1) \). This rightmost uncovered entry will be moved into row \( i \) in the computation of \( e_i(T) \) precisely when the corresponding entry in \( \omega P \) (under Perf) changes an \( i + 1 \) to an \( i \) in computing \( e_i(\omega P) \), proving the result.

\[ \square \]

Definition 5.2 Define a map Rot : PTab\( (\ell,n) \) → PTab\( (\ell,n) \) (recall the definition of PTab\( (\ell,n) \) from Definition 2.9) with Rot\( (T) \) the ptableau obtained from \( T \) by first rotating \( T \) by 180°, and then replacing each entry \( t \) by \( \ell - t + 1 \).
For example, if \( T = \begin{array}{cccc}
1 & 1 & 2 & 5 \\
1 & 1 & 2 & 4 \\
1 & 1 & 2 & 3
\end{array} \in \text{PTab}_{(8,4)} \), then
\[
\text{Rot}(T) = \begin{array}{cccc}
1 & 1 & 2 & 3 \\
2 & 3 & 4 & 5 \\
3 & 4 & 5 & 6 \\
4 & 5 & 5 & 7
\end{array}.
\]

Similarly,

**Definition 5.3** Define a map \( \text{Rot} : [n]^{\otimes k} \rightarrow [n]^{\otimes k} \) by first defining, for each \( \ell \in [n] \):
\[
\overline{\ell} = n - \ell + 1,
\]
and then defining
\[
\text{Rot}(\omega) = \text{Rot}(\omega_1\omega_2\ldots\omega_k) = \overline{\omega_k} \omega_{k-1} \ldots \overline{\omega_1}.
\]

We state the following easy facts, without proof:

**Proposition 5.4** For any \( T \in \text{PTab}_n \), we have \( *\text{Rot}(T) = \text{Rot}(T^*) \) and \( \text{Rot}(T^*) = \text{Rot}(T)^* \). Also, the map \( \text{Rot} \) is an involution, and finally
\[
\text{Perf}(\text{Rot}(\omega)) = \text{Rot}(\text{Perf}(\omega)).
\]

Analogously to Definition 4.7, we define the ptableau crystal operator \( f_i \) (\( 1 \leq i < n \)). Note that a result for right-justified ptableaux, analogous to Lemma 4.6, ensures that the following is well-defined:

**Definition 5.5** Suppose \( T \in \text{PTab}_n \) and that \( \boxed{c} \) is an entry in row \( i \) of \( T \) such that in \( T[i, i+1]^* \) (the right-justified form of \( T[i, i+1] \)) \( \boxed{c} \) is the left-most entry with a blank below it. In this case, we define \( f_i(T) \) to be the ptableau obtained by swapping \( \boxed{c} \) with the blank below it (in row \( i+1 \)) of \( T^* \). If there is no such entry \( \boxed{c} \), then \( f_i(T) = \text{NULL} \).

We define \( \phi_i(T) \) to be the number of blanks in row \( i+1 \) of \( T[i, i+1]^* \).

**Theorem 5.6** Given the map \( f_i : \text{PTab}_n \rightarrow \text{PTab}_n \) above, we have
\[
\text{Perf}(f_i(\omega_P)) = f_i(\text{Perf}(\omega_P)).
\]

**Proof** We note (from the definitions of the maps \( e_i \) and \( f_i \)) that \( \text{Rot}(f_i(\omega_P)) = e_{n-i}(\text{Rot}(\omega_P)) \) and \( \text{Rot}(e_{n-i}(T)) = f_i(\text{Rot}(T)) \), from which we have
\[
\text{Perf}(f_i(\omega_P)) = \text{Perf}(\text{Rot}(f_i(\omega_P)))
= \text{Perf}(\text{Rot}(e_{n-i}(\text{Rot}(\omega_P))))
= \text{Rot}(\text{Perf}(e_{n-i}(\text{Rot}(\omega_P))))
= \text{Rot}(\text{Perf}(\text{Rot}(\omega_P)))) \quad \text{by Theorem 5.1}
= f_i(\text{Perf}(\text{Rot}(\omega_P))))
= f_i(\text{Perf}(\omega_P)).
\]
The following Corollary highlights the convenience of computing crystal operators in \( \text{PTab}_n \). Unlike the complicated calculations for \( \phi_i \) and \( \epsilon_i \) for words in \([n]^{\otimes k}\), we can calculate the corresponding values for ptableaux with a simple count in \( T[i, i+1] \), without needing left- or right-justification:

**Corollary 5.7** Suppose \( T \in \text{PTab}_n \). Then we have

\[
\phi_i(T) = \text{the number of blanks in row } i+1 \text{ of } T[i, i+1],
\]

and

\[
\epsilon_i(T) = \text{the number of blanks in row } i \text{ of } T[i, i+1].
\]

The reader should compare the simplicity of the definitions for \( \phi_i(T), \epsilon_i(T) \) with the corresponding definitions of \( \phi_i(\omega) \) and \( \epsilon_i(\omega) \) in Proposition 4.1.

**Corollary 5.8** Let \( \omega_P \) be highest weight, let \( T_{\omega_P} = \text{Perf}(\omega_P) \), and \( \nu = \text{wt}(\omega_P) = \text{wt}(T_{\omega_P}) \). Let \( \mathbb{B}^\nu(T_{\omega_P}) \) be the closure of the images of (highest weight) \( T_{\omega_P} \) under the ptableaux operators \( e_i \) and \( f_i \), for \( 1 \leq i < n \). Then the restriction \( \text{Perf} : \mathbb{B}^\nu(\omega_P) \rightarrow \mathbb{B}^\nu(T_{\omega_P}) \) is an isomorphism of irreducible \( GL_n \) crystals. (Recall the definition of \( \mathbb{B}^\nu(\omega_P) \) from Definition 4.4.)

**Proof** The set \( \mathbb{B}^\nu(\omega_P) \) is a connected, irreducible subcrystal of \([n]^{\otimes k}\). Since \( \text{Perf} \) is a bijection from \( \mathbb{B}^\nu(\omega_P) \) that commutes with the crystal operators, and \( \text{wt}(T_{\omega_P}) = \text{wt}(\omega_P) \) as noted above, they are isomorphic crystals (See [5], p. 54).

6 **Highest Weights**

**Definition 6.1** We say a ptableau \( T \in \text{PTab} \) is partition-shaped if no blank box in \( ^*T \) has content to the right or below the blank (so all content is in the upper left corner of \( ^*T \), with no “holes”). We say a ptableau is anti-partition-shaped if in \( T^* \), no blank has content to the left or above it or, equivalently, if \( \text{Rot}(T) \) is partition-shaped.

For example, the ptableau

\[
T = \begin{array}{cccccc}
1 & 1 & 4 \\
1 & 2 & 5 \\
2 & 3 & 4 & 5 & 6
\end{array}
\]

is not partition-shaped since

\[
^*T = \begin{array}{cccc}
1 & 1 & 4 \\
1 & 2 & 5 \\
1 & 2 & 3 & 4 & 5 & 6
\end{array}
\]

and the leftmost blanks in the first two rows have content to their right. However, the ptableau

\[
T' = \begin{array}{cccc}
1 & 1 & 1 & 4 & 4 \\
2 & 4 & 4 & 6 \\
3 & 5
\end{array}
\]

is partition-shaped.
is partition-shaped since
\[ *T' = \begin{array}{cccc}
1 & 1 & 1 & 1 \\
4 & 4 & 6 & \\
3 & 5 & & 
\end{array} \, . \]

**Theorem 6.2** A ptableau \( T \in \text{PTab}_n \) is partition-shaped if and only if it is highest weight, meaning \( e_i(T) = \text{NULL} \) for all \( i \), \( 1 \leq i < n \).

Similarly, a ptableau \( T \) is anti-partition-shaped if and only if it is lowest weight, meaning \( f_i(T) = \text{NULL} \) for all \( i \), \( 1 \leq i < n \).

Thus, in the examples above, the ptableau \( T' \) is highest weight, while \( T \) is not.

**Proof** Clearly, if \( *T \) is partition-shaped, then it is highest weight, since \( e_i \) on \( *T \), (if not NULL), swaps a blank in row \( i \) with content directly below it in row \( (i + 1) \). When \( *T \) is partition-shaped, there are no such blanks. Equivalently, by Corollary 5.7 all \( \epsilon_i(T) = 0 \).

For the converse, we note that by Theorem 5.1 and Definition 4.7, if \( e_i(T) = \text{NULL} \) for all \( i \), each \( *T[i, i + 1] \) must be partition-shaped; otherwise there would be some blank over content, contradicting \( e_i(T) = \text{NULL} \). Hence \( T \) itself is partition shaped.

The proof of the second statement regarding lowest weights is proved analogously (indeed, isomorphically under the map Rot).

It is well-known that a word \( \omega \in [n]^{\otimes k} \) is a highest weight if and only if it is a Yamanouchi word: if \( \omega = \omega_1 \omega_2 \cdots \omega_{k-1} \omega_k \), then for each \( i \in [n-1] \) and each \( s, 1 \leq s \leq k \), the segment \( \omega_1 \omega_2 \cdots \omega_{s-1} \omega_s \) contains at least as many \( i \)'s as \( (i + 1) \)'s. The characterization of highest weight ptableau given above is, by contrast, generally easier to use. For example, to see if the minimally parsed word

\[ \omega = 111|22111|2221|33211|332|33 \]

is highest weight, we could check that all \( e_i \) are NULL on \( \omega \), or we could compute all the counts of \( i \)'s over \( (i + 1) \)'s across the length of the word to verify that the word is Yamanouchi, or, much more simply, we can make the associated ptableau:

\[ \text{Perf}(\omega) = \begin{array}{cccc}
1 & 1 & 1 & 1 \\
2 & 2 & 2 & 3 \\
3 & 3 & 3 & 4 \\
4 & 4 & 5 & 6 \\
5 & 5 & 6 & 6 
\end{array} \, . \]

and observe immediately that it is highest weight (unperforated). Indeed, viewing the \( e_i \) operators as moving blanks down to lower rows in a ptableau, and the \( f_i \) as moving blanks up, it is quite natural that highest weight ptableaux have no interior blanks; the blanks have been moved down as far as possible.

### 7 Tensor Products and the Littlewood-Richardson Rule

#### 7.1 Tensor Products in \([n]^{\otimes k}\)

Tensor products of crystals in \([n]^{\otimes k}\) are constructed by concatenation of words. If \( C_1 \) and \( C_2 \) are two crystal graphs (from, say, \([n]^{\otimes k}\) and \([n]^{\otimes s}\), respectively), then the nodes of the crystal graph \( C_1 \otimes C_2 \subseteq [n]^{\otimes (k+s)} \) are the concatenated words of \( C_1 \) and \( C_2 \), with the crystal operator acting on these words. The set of nodes in \( C_1 \otimes C_2 \) is closed under the crystal operators, and hence decomposes as a disjoint sum of irreducible crystals (see [5], p. 18).
Let $B^\nu(\omega)$ and $B^\mu(\omega')$ be two irreducible crystals in $[n]^{\otimes k}$ with highest weight elements $\omega$ and $\omega'$, respectively, with weights $wt(\omega) = \nu$ and $wt(\omega') = \mu$. A classical problem is to determine the number of irreducible crystal graphs of a given highest weight $\lambda$ in the tensor product $B^\nu(\omega) \otimes B^\mu(\omega')$. We denote this multiplicity by the Littlewood-Richardson coefficient, $c^\lambda_{\mu\nu}$. In order to determine $c^\lambda_{\mu\nu}$ combinatorially, it is not enough to use representatives of isomorphism classes, such as SSYT.

The issue is the SSYT crystal model is not closed under tensor products. As noted above, if $\nu$ is a partition of $n$ rows and $k$ parts, the map $RR : SSYT(\nu) \to [n]^{\otimes k}$ picks out only one representative irreducible crystal of highest weight $\nu$, with all other graphs with the same highest weight identified only up to isomorphism. So, for example, the pair of SSYT:

\[
T = \begin{array}{ccc}
1 & 1 & 3 \\
2 & 3 & \end{array}
\quad \quad \quad \quad \quad \quad T' = \begin{array}{ccc}
1 & 2 & 3 \\
2 & 3 & \end{array}
\]

correspond, under the row-reading map $RR$ to the words:

\[
T = \begin{array}{ccc}
1 & 1 & 3 \\
2 & 3 & \end{array} \quad \quad \quad \quad \quad \quad T' = \begin{array}{ccc}
1 & 2 & 3 \\
2 & 3 & \end{array}
\]

\[
\downarrow \quad \quad \quad \quad \quad \quad \downarrow
\]

\[
RR(T) = 31132 \quad RR(T') = 32132
\]

We can certainly form the tensor product of the words:

\[
RR(T) \otimes RR(T') = 31132 \otimes 32132 = 31132|32132,
\]

(the parsing is there for clarity, and does not affect the crystal structure). However, this word has weakly decreasing factors (from left to right) of size 3,2,3,2, and hence is clearly not the row-reading word of any SSYT (for which the lengths of factors must be weakly decreasing). Of course, there are tableau methods to replace $31132|32132$ with a the row-reading word of some SSYT, but this requires computing an isomorphic proxy for this tensor product.

To compute Littlewood-Richardson coefficients combinatorially, one must employ a different combinatorial model. Classically, this is achieved by enumerating Littlewood-Richardson fillings of the skew shape $\lambda/\mu$ with content $\nu$. Hence, in the setting of $[n]^{\otimes k}$, we use one class of objects (semistandard Young tableau) to build representative models of crystals, but to analyze tensor products, we need to build a different model (fillings of skew shapes). Using ptableaux, we can accomplish both tasks simultaneously.

### 7.2 Tensor Products of Ptableaux

**Definition 7.1** We adopt the following convention. Given two words $\omega, \omega' \in [n]^{\otimes k}$, with parsings $\mathcal{P}$ and $\mathcal{P}'$, resulting in $\omega_{\mathcal{P}}$ and $\omega'_{\mathcal{P}'}$, we will assign to $\omega_{\mathcal{P}} \otimes \omega'_{\mathcal{P}'}$, the parsing $\omega_{\mathcal{P}} \otimes \omega'_{\mathcal{P}'} = \omega_{\mathcal{P}} | \omega'_{\mathcal{P}'}$.

The tensor product of ptableaux is defined quite easily. Let $T \in \text{PTab}(\ell,n)$ and $U \in \text{PTab}(m,n)$. We form the concatenation $T \otimes U \in \text{PTab}((\ell+m),n)$ by:

1. Replace each box $[a]$ in $U$ by $[a + \ell]$. Call this ptableau $U^{+\ell}$.
2. Construct $T \otimes U$ by appending the rows of $U^{+\ell}$ to the right of the rows of $^*T$, from the bottom to the top, at each stage left-justifying appended rows.
For example, if \( T = \begin{array}{ccc}
1 & 1 & 2 \\
1 & 2 & 3 \\
2 & 3 & 4
\end{array} \) \( U = \begin{array}{ccc}
1 & 1 & 2 & 4 \\
1 & 2 & 3 & 4 \\
3 & 3 & 3
\end{array} \) \( \in \) \( \text{PTab}(4, 3) \), then
\[
U^{+4} = \begin{array}{cccc}
5 & 5 & 6 & 8 \\
6 & 6 & 7 & \\
7 & 7 & & \\
& & & \\
\end{array}
\in \text{PTab}(8, 3), \text{ and so }
\]
\[
T \otimes U = \begin{array}{cccc}
1 & 1 & 2 & 3 \\
1 & 2 & 3 & 3 \\
2 & 3 & 4 & 7 \\
& & & 7
\end{array}
\].

Note that it is possible that a blank appearing in \( U \) will not persist in \( T \otimes U \).

The following is easily proved:

**Lemma 7.2** Concatenation of words in \([n]^{\otimes k}\) commutes with \( \text{Perf} \) in \( \text{PTab}_n \). That is,
\[
\text{Perf}(\omega_P \otimes \omega'_P) = \text{Perf}(\omega_P) \otimes \text{Perf}(\omega'_P).
\]

As noted earlier, a word \( \omega \in [n]^{\otimes k} \) is of highest weight if and only if it is a *Yamanouchi* word, that is, if \( \omega = \omega_1 \omega_2 \cdots \omega_{k-1} \omega_k \), then for each \( i \in [n-1] \) and each \( \ell, 1 \leq s \leq k \), the segment \( \omega_1 \omega_2 \cdots \omega_{s-1} \omega_s \) contains at least as many \( i \)'s as \( (i+1) \)'s. Thus, a concatenation of words \( \omega \otimes \omega' \) is highest weight precisely when it is Yamanouchi. In the ptableau setting, this is essentially a corollary to Theorem 6.2:

**Theorem 7.3** Let \( T, T' \in \text{PTab}_n \). Then \( T \otimes T' \) is highest weight if and only if \( T \otimes T' \) is partition-shaped. In particular, this implies that \( T \) is highest weight (and hence itself partition-shaped).

**Proof** By Theorem 6.2, \( T \otimes T' \) is highest weight if and only if \( T \otimes T' \) is partition-shaped. Since \( T \) forms the interior, upper-left corner of \( T \otimes T' \), \( T \otimes T' \) will be highest weight if and only if \( T \) already has no interior blanks (in its left-justified form), in other words, \( T \) itself is highest weight. \( \square \)

For example, suppose \( T = \begin{array}{cccc}
1 & 1 & 2 & 3 \\
1 & 2 & 3 & 3 \\
2 & 3 & 4 \\
\end{array} \) and \( T' = \begin{array}{ccc}
1 & 2 & 2 \\
1 & 2 & 3 \\
2 & 3 & 3 \\
\end{array} \).

Then
\[
T \otimes T' = \begin{array}{cccccccc}
1 & 1 & 1 & 2 & 2 & 3 & 4 & 4 \\
1 & 2 & 3 & 3 & 3 & 5 & 5 & 5 \\
2 & 3 & 4 & 5 & 6 & 7 & 6 & 6 \\
& & & & & & & 7
\end{array}
\],

which is partition-shaped, hence highest weight.

**Theorem 7.4 (Littlewood-Richardson Rule for Ptableaux)** Let \( \mathbb{B}^\mu(T_{\max}) \) and \( \mathbb{B}^\nu(U) \) be two irreducible crystals in \( \text{PTab}_n \) of highest weights \( \mu \) and \( \nu \), with highest weight ptableaux \( T_{\max} \) and \( U \), respectively. Then
\[
c_{\mu\nu}^\lambda = \# \{ T \in \mathbb{B}^\nu(U) : T_{\max} \otimes T \text{ is partition-shaped, of shape } \lambda \}.
\]

In particular, the ptableaux \( T_{\max} \otimes T \) are the highest weight elements in the irreducible constituents of \( \mathbb{B}^\mu(T_{\max}) \otimes \mathbb{B}^\nu(U) \) of highest weight \( \lambda \).
7.3 Connections with the Classical Littlewood-Richardson Rule

Theorem 7.4 should be viewed as a generalization of the classical Littlewood-Richardson rule for enumerating Littlewood-Richardson coefficients $c_{\mu\nu}^\lambda$. Below, we make this connection precise, and in doing so, highlight an advantage that the ptableaux model has over the SSYT model for crystals in $[n]^\otimes k$.

Let $B_{\mu}$ and $B_{\nu}$ denote two irreducible crystals of SSYT, of highest weight $\mu$ and $\nu$, respectively. In the literature one finds expressions such as:

$$B_{\mu} \otimes B_{\nu} \cong \bigoplus \lambda c_{\mu\nu}^\lambda B_{\lambda}$$

which denote decompositions of a tensor product into irreducible constituents (up to isomorphism). However, as noted above, there is no actual definition of a tensor product of SSYT crystals. At best, we identify each factor of $B_{\mu} \otimes B_{\nu}$ with their row reading image (a canonical crystal of words) and compute the tensor product by concatenation of words. But, by virtue of this, we must replace the actual decomposition of $B_{\nu} \otimes B_{\mu}$ into irreducible constituents with isomorphic copies in the sum $\bigoplus \lambda c_{\mu\nu}^\lambda B_{\lambda}$ (the use of “$\cong$” is essential). That is, typically no crystal in $B_{\mu} \otimes B_{\nu}$ equals $B_{\lambda}$, even when identifying it with a crystal of words.

This is not problematic in itself and, indeed, the point of the Littlewood-Richardson rule is to count multiplicities of isomorphic crystals. The trouble with the SSYT model is that it is of no help in evaluating $c_{\mu\nu}^\lambda$, or analyzing its properties. Thus, one needs an entirely new combinatorial model, namely Littlewood-Richardson fillings.

The Littlewood-Richardson Rule enumerates $c_{\mu\nu}^\lambda$ by counting the set of Littlewood-Richardson fillings of the skew shape $\lambda/\mu$ with content $\nu$. To find such a filling, remove the partition $\mu$ from the partition $\lambda$ leaving the skew shape $\lambda/\mu$. In the remaining boxes, using the partition $\nu = (\nu_1, \ldots, \nu_n)$, we place in the skew shape $\nu_1$-many 1’s, $\nu_2$-many 2’s, etc., filling the entire skew shape, so that the filling is semistandard, and such that the filling satisfies the word condition, meaning that the row reading word of the filling (in the skew shape) is a Yamanouchi word. The number of such fillings equals $c_{\mu\nu}^\lambda$.

We will demonstrate how this classical construction is a special case of our Theorem 7.4 above. Recall the row reading we adopted for SSYT. We obtain a word $\omega$ from a SSYT $P$ by writing the content in a row from right to left, from the top row to the bottom.

Definition 7.5 We say a ptableau $T \in \text{PTab}_n$ satisfies the word condition on ptableaux if and only if $T = \text{Perf}(\omega)$, where $\omega$ is the row-reading word of a SSYT $P$.

We use the term “word condition” out of tradition as it is used in the context of the Littlewood-Richardson rule. However, the word condition on ptableaux does not correspond to the Yamanouchi condition on words in $[n]^\otimes k$. That is, if a ptableau $T$ satisfies the word condition, this does not imply that $T = \text{Perf}(\omega)$ for some Yamanouchi word $\omega$. Further, while Yamanouchi words (being highest weight elements in $[n]^\otimes k$) map to highest weight ptableaux (partition-shaped by Theorem 6.2), when we say a ptableaux itself satisfies the word condition, it need not be highest weight, and need only satisfy the classical “word condition” on Littlewood-Richardson fillings.

Lemma 7.6 Suppose $T$ satisfies the word condition for ptableaux. Then $T$ lies in a connected crystal graph such that all ptableaux in the graph satisfy the word condition.

Proof This follows by noting that the maps from SSYT to words, and then from words to ptableaux, are isomorphisms of crystal graphs [5].

\[ \text{Springer} \]
The following is easily shown:

**Lemma 7.7** A ptableau $T$ satisfies the word condition on ptableaux (Definition 7.3) if and only if the number of $i$'s appearing in $T$ in rows $i$ through $i + k$ is greater than or equal to the number of $(i + 1)$'s appearing in rows $(i + 1)$ through $i + k + 1$, for $k \geq 0$.

Let $\nu = (\nu_1, \ldots, \nu_k)$ be a partition, and let $T_\nu$ denote the ptableau whose first row contains $\nu_1$-many 1's, whose second row contains $\nu_2$-many 2, etc. So, for example, if $\nu = (4, 3, 3, 1)$ then:

$$T_\nu = \begin{array}{cccc}
1 & 1 & 1 & 1 \\
2 & 2 & 2 & \\
3 & 3 & 3 & \\
4 & \\
\end{array}$$

**Corollary 7.8** If a ptableau $T \in \PTab_n$ satisfies the word condition, it lies in some $B^\nu(T_\nu)$, the crystal with highest weight $T_\nu$, where $\nu = (\nu_1, \nu_2, \ldots)$, and $\nu_1$ is the number of 1's appearing in $T$, $\nu_2$ is the number of 2's appearing in $T$, etc. In particular, the irreducible crystal $B^\nu(T_\nu)$ is composed of ptableaux satisfying the word condition on ptableaux.

**Proof** Suppose $T$ is right-justified (the word condition is clearly invariant under row equivalence). The word condition implies that we only find 1's in row 1 of $T$, only 1's and 2's in rows 1 and 2 of $T$, etc. Thus, in row 2, by the word condition the 2’s in row 2 do not extend left past the 1’s in row 1, and since $T$ is right-justified, the 1’s in rows 1 and 2 have no gaps in the columns in which they appear. But continuing, since the 3’s in row 3 cannot extend to the left past the 2’s in row 2, then the 2’s in row 3 cannot extend to the left past the 1’s in row 2, so that the 1’s appearing in rows 1,2, and 3 appear in consecutive columns, etc. Thus, the horizontal strip of 1’s is maximal, in that there is no column in $T$ not containing a 1. But then those 1’s may only move directly upward (under the action of some crystal operator $e_i$), so that in the highest weight ptableau of $T$, the top row consists entirely of 1’s. After performing these operations (moving 1’s upward), we argue inductively to conclude the highest weight is of the form $T_\nu$.

Since $T_\nu$ satisfies the word condition, then by Lemma 7.3 $B^\nu(T_\nu)$ is composed of ptableaux satisfying the word condition on ptableaux.

With these results, we see that the classical Littlewood-Richardson rule is a special case of Theorem 7.4 above. Consider some Littlewood-Richardson filling of skew shape $\lambda/\mu$:

Here, $\mu = (6, 3, 1, 0)$, $\lambda = (10, 8, 5, 4)$, and $\nu = (7, 5, 3, 2)$. We can fill the blanks of shape $\mu$ with any semistandard content; for now we use $T_\mu$, and distinguish it by circling the content:

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

© Springer
Viewing the above as a (highest weight) perforated tableau, we note that, up to a relabeling of content (we would have to add 3 to each entry of $T$ under our definition of tensor product), we may regard this tableau as a tensor product:

That is, the filling of the skew shape $\lambda/\mu$ is determined by the tableau

Since $T$ satisfies the word condition by Lemma 7.3, then, by Corollary 7.8, we see that $T$ is in the crystal whose highest weight is $T\nu$. Thus, by Theorem 7.4 the number of $T \in B\nu(T\nu)$ such that $T\mu \otimes T$ is partition shaped (with left-justified content of shape $\lambda$) equals the number of Littlewood-Richardson fillings of $\lambda/\mu$ of content $\nu$, and each filling is, up to a relabeling, row equivalent to some such ptableau $T$.

8 Evacuation, Involutions and Highest Weights in PTab

We now present combinatorial algorithms for computing with highest weight elements in ptableaux crystals. These results show how to connect crystal operators on perforated tableaux to well-known algorithms, including Schützenberger evacuation and the tableau switching of Benkart, Sottile, and Stroomer (see [3]). The basis of these connections will be Lemma 8.3 below, showing that the Schützenberger evacuation map can be “factored” as a product of ptableaux crystal operators.

8.1 Evacuation and SSYT

Crystal graphs enjoy a number of symmetries. One that has been extensively studied is the **Lusztig Involution**, denoted here $\phi_{Lus}$, defined by the condition that if $c_{max}$ is the highest weight element in some irreducible crystal graph $C$, then the image of the Lusztig Involution is

The Lusztig involution is defined by the relations above. However, when nodes of crystal graphs are determined by combinatorial models, various combinatorial algorithms for computing $\phi_{Lus}$ have been of interest.

In the SSYT model of crystals, the Lustig image of a SSYT $T$, $\phi_{Lus}(T)$, is computed by the **Schützenberger involution** on SSYT (see [4]). The Schützenberger involution has two steps. The first is the **evacuation** procedure (described below) producing an anti-partition shaped tableau. This is followed by rotation, which is the Rot map defined above, applied to the output of the evacuation step, regarding the output as a perforated tableau (but whose left-justified image under Rot becomes a SSYT once again).

The evacuation step of the Schützenberger involution is performed with a series of **jeu de taquin** slides which pass a blank through a given SSYT $T$. Since performing jeu de taquin slides on a ptableau produces another ptableau, while the image of a semistandard tableau...
after jeu de taquin need not be semistandard (in that it may have interior blanks), it will be convenient to regard \( T \) (temporarily) as a left-justified ptableau (so \( T = ^*T \)) during the evacuation step.

**Definition 8.1** (See [14, 15]) Define \( \phi_{\text{Lus}} \) on crystals of SSYT as follows: First, define the map \( \text{Evac} : \text{PTab}_n \to \text{PTab}_n \). Given a SSYT \( T \), regarded as a left-justified ptableau, an **outer corner** of \( T \) is a blank appearing in \( ^*T \) with no other blanks immediately to its left, or immediately above it, and all blanks below it and to its right. Thus, in \( ^*T \), an outer corner will have content above it (labeled \( b \) below), and to its left (labeled \( c \) below):

\[
\begin{array}{ccc}
... & * & a \\
... & * & b \\
... & c & ...
\end{array}
\]

Fix some choice of outer corner, and call this the **distinguished blank**. Then perform **inward jeu de taquin** on this blank as follows: Let \( c \) be the content to the left of the distinguished blank (as in the figure above), and \( b \) be the content immediately above it. If \( b \geq c \), swap the distinguished blank with \( b \). If not, (so \( b < c \)), swap the blank with \( c \). Either of these swaps is called a **jeu de taquin slide**, or just a **slide**. At this point, the distinguished blank will either:

1. Have content above it and to its left, in which case we again perform a jeu de taquin slide on the distinguished blank.
2. Have content **only** above it, or **only** to its left, in which case we swap the blank with this single content entry.
3. Have only blanks to its left and above it, in which case we do nothing, and consider the blank “fixed”.

We repeat this process, performing slides on the distinguished blank until it has been fixed. We then identify a new outer corner as the distinguished blank, and perform inward jeu de taquin on it. This is repeated until there are no outer corners remaining, resulting in a ptableau \( \text{Evac}(T) \). To complete the Schützenberger involution, we then apply the map \( \text{Rot} \) to \( \text{Evac}(T) \):

\[
\phi_{\text{Lus}}(T) = \text{Rot}(\text{Evac}(T)).
\]

Berenstein and Zelevinsky (see [4]) proved \( \phi_{\text{Lus}}(T) = \text{Rot} \circ \text{Evac}(T) \) on the irreducible crystal containing the SSYT \( T \). They also showed that \( \text{Evac}(T) \) is independent of the sequence of outer corners chosen in \( T \).

The evacuation portion of the algorithm above starts with an anti-partition shape of blanks in the lower-right corner of the diagram that migrate upwards, under inward jeu de taquin, to form a partition-shaped collection of blanks in the upper left, of the same corresponding shape. Equivalently, if a (partition-shaped) ptableau \( T \) has weight \((a_1, \ldots, a_k)\) (where \( a_i \geq a_{i+1} \) since it is partition shaped), then the ptableau \( \text{Evac}(T) \) will have weight \((a_k, a_{k-1}, \ldots, a_1)\).

For example, given a semistandard Young tableau \( T \):

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

we compute \( \text{Evac}(T) \) by first performing inward jeu de taquin on the outer corner in row 2:

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

...
Continuing inner jeu de taquin, we obtain:

$$\text{Evac}(T) = \begin{array}{cccc}
1 & 2 & 3 \\
2 & 3 & 4 \\
3 & 4 & 4 & 5
\end{array}$$

Thus, (see [4]) the composition of Evac with Rot:

$$\text{Rot}(\text{Evac}(T)) = \text{Rot} \left( \begin{array}{cccc}
1 & 2 & 3 & 4 \\
2 & 3 & 4 & 5 \\
1 & 3 & 4 & 4 & 5
\end{array} \right) = \begin{array}{cccc}
1 & 2 & 3 & 5 \\
2 & 3 & 4 & 5 \\
3 & 4 & 4 & 5
\end{array},$$

results in the SSYT $\phi_{\text{Lus}}(T)$, the value of the Lusztig involution in the crystal of SSYT applied to $T$.

Recall that an arbitrary SSYT $T$ (which can represent an arbitrary node in a crystal of SSYT) may also represent the (left-justified) highest weight node $T$ in some crystal of ptableaux. In crystals of SSYT we can factor the Lusztig involution as $\phi_{\text{Lus}} = \text{Rot} \circ \text{Evac}$.

In the next section we show that the map Evac itself factors as a product of ptableaux crystal operators. In fact, we show that, regarding some partition-shaped $T$ as a highest weight ptableau, the map Evac takes $T$ to its associated lowest weight ptableau. In the example above one can check $f_1 f_2 f_2 f_2 f_1 f_1 T = \text{Evac}(T)$. Since Evac($T$) is anti-partition shaped and is the image of $T$ under crystal operators, it is the lowest weight element in the irreducible crystal with highest weight $T$, by Theorem 6.2. It can be shown that the sequence of indices of the $f_i$’s used in this process form a “good word” in the language of string patterns (see [5]).

Thus, the evacuation map gives a combinatorial algorithm to compute lowest weight ptableaux from their associated highest weights, as well as interpreting jeu de taquin slides as crystal operators on perforated tableaux. Finding a combinatorial algorithm to compute Lusztig involutions for arbitrary ptableaux is outside the scope of this paper, but can be achieved (and requires an analysis of the Robinson-Schensted construction in the setting of ptableaux [2]).

### 8.2 Evacuation and Crystal Operators on Ptableaux

**Definition 8.2** We say $T \in \text{PTab}$ is skew-shaped if it is row-equivalent to a ptableau in which the only blanks appearing are a partition-shaped collection of blanks in the upper left corner, and an anti-partition-shaped collection of blanks in its lower right corner. We call this the skew-shaped form of $T$.

We say a blank of a skew-shaped ptableau $T$ is an inner corner of the skew-shaped form of $T$ if it appears in the partition-shaped collection of blanks in the upper left-corner, and if there are no blanks to its right or below it.

In other words, a ptableau is skew-shaped if, in its skew-shaped form, the content is skew-shaped in the traditional sense of tableaux.

Above we made use of inward jeu de taquin in the definition of the map Evac, where a blank in an outer corner is moved left or upward by jeu de taquin slides. Below we consider outward jeu de taquin, where blanks starting in an inner corner are swapped with content below or to the right of the blank.

**Lemma 8.3** Suppose $T$ is a skew-shaped ptableau. Choose an inner corner $S$ of this skew shape (that is, $S$ is a blank in the skew-shaped form of $T$ that has content directly to its right and directly below it), appearing in row $s$. We perform outward jeu de taquin on this skew-shaped form, starting at $S$, and call the resulting skew-shaped ptableau $T'$. Let $P$ denote
the outward jeu de taquin path of \( S \) from the inner corner of \( T \) in row \( s \) to an outer corner of \( T' \) appearing in some row \( t + 1 \), where \( s \leq t \). Then, up to row-equivalence, we have

\[
T' = e_t e_{t-1} \cdots e_s T,
\]

and the path of \( S \), resulting from these crystal operators, is also \( \mathcal{P} \).

**Proof** Let \( T \) be a skew-shaped ptableau, and for simplicity, denote its (fixed) skew-shape form by \( T \). Fix an inner corner of \( T \), denoted \( \square \), appearing in row \( s \) (shaded gray below). To perform outward jeu de taquin, we swap \( \square \) with content in \( T \) as long as all entries remain column strict.

We prove the claim inductively by showing that, for each step of the outward jeu de taquin path of \( \square \), if the slide moves \( \square \) from row \( k \) to row \( k + 1 \), the same (row-equivalent) ptableau is obtained by applying \( e_k \) to the ptableau determined prior to that slide.

Suppose we performed a sequence of jeu de taquin slides, stopping when \( \square \) appears in row \( i \), and is no longer able to move right without creating a column strictness violation. Call the resulting ptableau \( T' \) (where the \( \bigcirc \) denotes content):

We assume, inductively, that this ptableau is row-equivalent to \( e_{i-1} \cdots e_s T \). Because the blank \( \bigcirc \) cannot move farther right at this point, we must be unable to move the box \( \times \) over the box \( \bigcirc \), and so \( a \leq x \).

The evacuation procedure thus swaps the blank \( \bigcirc \) in row \( i \) with the content \( \bigcirc \) in row \( (i + 1) \). We show that, after we left justify \( T' \), computing \( e_{i}(T') \) will swap \( \bigcirc \) with the blank directly above it, which will show that the evacuation procedure matches the crystal operators as described in the statement of the lemma.

Since \( T \) is initially skew-shaped, we may assume \( T[i, i + 1] \), before we left justified, had the form:

\[
\begin{array}{ccc}
\cdots & \cdots & \bigcirc \ x \ \cdots & \cdots \\
\cdots & \cdots & \bigcirc \ a \ y \ \cdots
\end{array}
\]

where the boxes to the left of \( \bigcirc \) are filled.

Thus, in order to complete the left-justification of \( T[i, i + 1] \) we need only move the content to the left of the blank \( \bigcirc \) in row \( i \) (possibly) further left, but all other content in row \( i \) cannot move. Thus, the blank \( \bigcirc \) is necessarily the rightmost blank in row \( i \) over content in row \( i + 1 \), so by the definition of \( e_i \) we move the entry \( \bigcirc \) upward into row \( i \). If evacuation swaps \( \bigcirc \) down and the content \( \bigcirc \) up, then upon left justification, \( e_i(T) \) would swap \( \bigcirc \) up as well.
By duality under the Rot map, we then have:

**Corollary 8.4** Suppose a ptableau $T$ has a skew shape row-equivalent form. Choose an outer corner $S$ of this skew shape (so $S$ is a blank that has content directly to its left and directly above it), appearing in row $j + 1$, and perform inward jeu de taquin, moving $S$ from the lower outer corner of $T$ in row $j + 1$ to an inner corner of $T'$ in row $i$, where $i \leq j$. Then, up to row equivalence, we have

$$T' = f_1 f_{i-1} \cdots f_j T.$$ 

We now prove the results mentioned at the end of Section 8.1.

**Theorem 8.5**

1. Let $T \in SSYT(\nu)$ be a semistandard Young tableau of shape $\nu$, or equivalently, a partition-shaped ptableau of weight $\nu$. Then, there is a sequence of ptableaux crystal operators $f_{i_1}, \ldots, f_{i_s}$ such that

$$Evac(T) = f_{i_1}, \ldots, f_{i_s} T.$$ 

In particular, each evacuation path constructed in computing $Evac(T)$ by inward jeu de taquin may be “factored” as a product of ptableaux crystal operators.

2. If $T_{\min}$ denotes the lowest weight element in the ptableaux crystal containing the (partition-shaped) highest weight element $T$, then

$$T_{\min} = Evac(T).$$

**Proof** We identify some SSYT $T$ as a highest weight ptableau $T = T_{\max} \in \text{PTab}_n$, by Theorem 6.2. We assume, for now, that $n$ is chosen so that there are no entirely blank rows in $T_{\max}$. Thus, in its left-justified form, all blanks in $T_{\max}$ appear in the lower right corner, and this collection of blanks is anti-partition shaped (the number of blanks weakly increases as we proceed down the rows).

Applying Evac to $T_{\max}$ moves these blanks to the upper left corner. We choose to move one blank entirely through $T_{\max}$ before starting another, so that Corollary 8.4 applies. Once all blanks in the anti-partition shape of blanks below $T_{\max}$ have been moved above it, $T_{\max}$ has been transformed into the form of a lowest weight ptableau (which we denote $T_{\min}$) by Theorem 6.2. By Corollary 8.4, this ptableau is in the same crystal as $T_{\max}$.

Thus, as in the example above, the transformation from the highest weight ptableau

$$T_{\max} = \begin{array}{cccc}
1 & 1 & 2 & 2 \\
2 & 3 & 3 & 4 \\
3 & 4 & 5 & \\
\end{array},$$

to the lowest weight, $T_{\min} = \begin{array}{cccc}
1 & 2 & 3 \\
2 & 2 & 3 & 4 \\
1 & 3 & 3 & 4 \\
\end{array}$,

by evacuation is also obtained by an appropriate sequence of crystal operators $f_i$.

For the general case, we argue by example (the general principle will be clear). Assume that the highest weight ptableau $T_{\max}$ has some rows of blanks:

$$T_{\max} = \begin{array}{cccc}
1 & 1 & 2 & 2 \\
2 & 3 & 3 & 4 \\
3 & 4 & 5 & \\
\end{array}.$$
Applying evacuation in the first three rows of $T_{\text{max}}$, or applying the same appropriate set of crystal operators $f_i$, will result in the ptableau:

$$T' = \begin{array}{ccc}
1 & 2 & 3 \\
2 & 2 & 3 \\
1 & 3 & 4 & 4 & 5
\end{array}.$$  

But then 

$$(f_1)^3(f_2)^4(f_3)^6(T') = \begin{array}{ccc}
1 & 2 & 3 \\
2 & 2 & 3 \\
1 & 3 & 3 & 4 & 4 & 5
\end{array}.$$  

We now repeat the process to move blanks in the bottom two rows upward, either by evacuation or crystal operators, finally obtaining the lowest weight:

$$T_{\text{min}} = \begin{array}{ccc}
1 & 2 & 3 \\
2 & 2 & 3 \\
1 & 3 & 3 & 4 & 4 & 5
\end{array}.$$  

and the general result follows.

The second claim of the Theorem follows by the second claim of Theorem 6.2 since $\text{Evac}(T_{\text{max}})$ is anti-partition shaped and in the image of $T_{\text{max}}$ under crystal operators.

We should note that, as pointed out in [17], computing commutators for tensor products of words can be reduced, algorithmically, to computing commutators of individual factors, and that, in turn, this computation may be computed by jeu de taquin slides on tableaux formed by those factors. Further, subsequent to our results the recent work of Gerber and Lecouvey [8] has appeared and shows that jeu de taquin slides on tableaux can be related to crystal operators on the corresponding duals of those objects. However, as pointed out earlier, these results do not (quite) imply ours, as we show, in the case of ptableaux, that the jeu de taquin moves are themselves the images of crystal operators on the same object, and not its dual.

### 8.3 Commutators and Highest Weights

We now consider the problem of combinatorially determining highest weight ptableaux in tensor products of crystals, using some of the tools of the previous section. Let $B_{\mu}$ and $B_{\nu}$ be two irreducible crystals of ptableaux, with highest weight elements of weight $\mu$ and $\nu$ respectively. This tensor product decomposes:

$$B_{\mu} \otimes B_{\nu} \cong \bigoplus_{\lambda} B_{\lambda}^{\varphi_{\mu,\nu}^c},$$

where, in the the isomorphism above, $B_{\lambda}$ is a fixed irreducible crystal of highest weight $\lambda$, and the Littlewood-Richardson coefficient, $c^c_{\mu,\nu}$ counts its multiplicity in the decomposition. However, $B_{\mu}$ and $B_{\nu}$ denote specific crystals of ptableaux (corresponding to specific words in $[n]^{\otimes k}$), so, in addition to counting the multiplicity of irreducible crystals in a given isomorphism type, we should try to distinguish and compute within a particular, realized tensor product.

By Theorem 7.4 we know that a highest weight element of an irreducible constituent of the tensor product $B_{\mu} \otimes B_{\nu}$ has the form $T^{\mu,\text{max}} \otimes T$ where $T^{\mu,\text{max}}$ is the highest weight.
element of $B_\mu$, $T \in B_\nu$, and $T_{\mu \text{ max}} \otimes T$ is a partition-shaped ptableau, of some shape $\lambda$ (so $c_{\mu \nu}^{\lambda} \neq 0$).

Further, it is well known that the crystals $B_\mu \otimes B_\nu$ and $B_\nu \otimes B_\mu$ are isomorphic. A crystal isomorphism $\tau : B_\mu \otimes B_\nu \rightarrow B_\nu \otimes B_\mu$ is called a “commutator”. Given some element $T \otimes T' \in B_\mu \otimes B_\nu$, Henriques and Kamnitzer [9] compute a commutator $\tau(T \otimes T') \in B_\nu \otimes B_\mu$ by means of the Lüstig involution, $\tau(T \otimes T') = \phi_{\text{Lus}}(\phi_{\text{Lus}}(T') \otimes \phi_{\text{Lus}}(T_{\text{inv}}))$. One would hope, however, for a purely combinatorial realization of this map in terms of ptableaux. The general case has been obtained, but is outside the scope of this paper [2]. Here, however, we present a combinatorial algorithm to compute commutators of highest weight elements in tensor products, which is of interest not only as a combinatorial algorithm, but in its connections to similar algorithms appearing in other contexts.

A special case of the commutator map in the context of Littlewood-Richardson fillings (but not crystals) has been known for many years. Recall that (left-justified) highest weight ptableaux can be viewed as SSYT. In particular, classical results on Littlewood-Richardson fillings can be viewed, using ptableaux, as special tensor products resulting in highest weight ptableaux (by Theorem 7.4 above). The classical Littlewood-Richardson rule was obtained as a special case of this result, when both $B_\mu$ and $B_\nu$ are the unique crystals (of their respective weights) that correspond to reading words of SSYT (see Corollary 7.8 and the discussion following).

In this special setting, the commutator is defined as a bijection between Littlewood-Richardson filling of $\lambda/\mu$ with content $\nu$ (where $\lambda$ is the weight of the fixed, irreducible constituent of $B_\mu \otimes B_\nu$), and a corresponding Littlewood-Richardson filling of $\lambda/\nu$ of content $\mu$ (corresponding to constituents of $B_\nu \otimes B_\mu$). James and Kerber [11] give this map by defining a combinatorial algorithm that moves the content of the Littlewood-Richardson filling of content $\nu$ “through” the base of $\mu$, so that the tableau becomes one of base $\nu$, with content $\mu$.

Benkart, Sottile, and Stroomer [3] show that James’ and Kerber’s algorithm is a special case of a more general “tableau switching” algorithm.

We make use of the work of Benkart, Sottile and Stroomer (BSS) in what follows, and also describe a combinatorial algorithm on ptableaux, in the case of highest weights, that will map a given highest weight element of $B_\mu \otimes B_\nu$ to a canonical corresponding highest weight ptableau in $B_\nu \otimes B_\mu$. That is, given some $T_{\mu \text{ max}} \otimes T \in B_\mu \otimes B_\nu$, where $T_{\mu \text{ max}}$ and $T_{\nu \text{ max}} \otimes T$ are both highest weight, we define a combinatorial algorithm $\tau(T_{\mu \text{ max}} \otimes T) = T_{\nu \text{ max}} \otimes T''$ such that

1. $T'' \in B_\mu$ (the same crystal, not merely one isomorphic to $B_\mu$).
2. $T_{\nu \text{ max}} \in B_\nu$ is the highest weight element in $B_\nu$.
3. $T_{\nu \text{ max}} \otimes T''$ is highest weight, and $\text{wt}(T_{\mu \text{ max}} \otimes T) = \text{wt}(T_{\nu \text{ max}} \otimes T'')$.

The algorithm can be described as a type of tableau switching (in the sense of BSS), an evacuation procedure, or a coordinated set of crystal operators. We use all three interpretations to prove the equivalence of these algorithms, and to show they satisfy the requirements above.

We use Lemma 8.3 and its Corollary 8.4. We also need the lemma below, which will help keep track of pairs of evacuation paths.

**Lemma 8.6** Suppose $\square_1$ and $\square_2$ are outer corners of a skew shaped ptableau $T$, with $\square_1$ weakly below and strictly left of $\square_2$. Then, the evacuation path of $\square_1$ lies weakly below and strictly left of the evacuation path of $\square_2$.

 Springer
Proof  Note that, in the diagrams below, we label □₁ and □₂ as 1 and 2. Assume that at some stage of the evacuation of □₁, we have moved it as far left as it can go in its current row. Immediately before moving □₁ up, \( T \) contains

\[
\begin{array}{cccc}
  b & c \\
  a & 1 & d \\
\end{array}
\]

Because we cannot move □₁ left, we have that \( a \leq b \). After swapping □₁ and \([b]\), \( T \) has

\[
\begin{array}{cccc}
  1 & c \\
  a & b & d \\
\end{array}
\]

Now assume that at some stage in the evacuation of □₂, it has moved left as far as possible, to the column immediately to the right of the vertical path of □₁. So □₂ occupies the box that contained \( d \), or occupies a higher box in that column, with an entry \( b' < b \) immediately to its left. (In reality, □₂ might be several rows below □₁, but the argument will still hold.)

\[
\begin{array}{cccc}
  1 & c \\
  a & b' & 2 \\
\end{array}
\]

We must show that □₂ now moves up, so that its evacuation path does not cross to the left of □₁’s evacuation path. In the first case, as noted above, \( b \leq c \) and so to maintain \( T \)’s column-strictness, □₂ must move up. In the second case, because □₁ swapped with \( b' \), by column-strictness \( b' \leq c' \). Thus to maintain column-strictness, □₂ must swap with \( c' \) above it. Hence, in either case, □₂ moves vertically, and does not cross the vertical path of □₁.

□₁ started weakly below and strictly to the left of □₂, and so the evacuation path of □₁ must be weakly below and strictly left of that of □₂.

In the work of Benkart, Sottile and Stroomer (BSS) [3], what we define below as BSS perforated tableaux are called perforated tableaux. Their definition omits Properties 2 through 5 in our Definition 2.3:

**Definition 8.7** (See [3]) Let \( T \) be a tableau of numbers and blanks, with numbers from \([n]\). Then \( T \) is a BSS perforated tableau if, ignoring blanks, the entries of \( T \) are weakly increasing in rows and strictly increasing in columns.

The pttableaux defined here (Definition 2.3) are BSS perforated, but the converse does not hold. For example, the tableau

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

is BSS perforated, but not a pttableau, since the 2’s, 3’s, and 4’s do not form horizontal strips.

**Definition 8.8** Suppose we have a tableau with content of two “types,” for example, 1, 2, 3, \ldots, and 1, 2, 3, \ldots, but no blanks. If the portion of the tableau formed by each type of number separately form BSS perforated tableaux, then the entire tableau is called a BSS perforated pair.
For example, the diagram below is a BSS perforated pair:

```
1 1 1 1 2 2
2 2 3 2 4 2
2 2 3 3 3
3 3 4 4
```

Algorithm 1 “Push Down” Given: $S = T_{\mu \text{max}} \otimes T$, where $T_{\mu \text{max}}$ and $T_{\mu \text{max}} \otimes T$ are highest weight (partition shaped) ptableaux. Call the filling of $T_{\mu \text{max}}$ (resp. $T$) the $\mu$ content (resp. $\nu$ content) of $S$. At the start of the algorithm, the $\nu$ content forms a skew shape.

Let $k$ be the largest value in the $\mu$ content. The rightmost, highest $k$ in the $\mu$ content lies in an inner corner of the skew shape of the $\nu$ content.

Use outer evacuation to push the highest, rightmost $k$ through the $\nu$ content, until it emerges in an outer corner of the (still skew shaped) $\nu$ content. We work through the $k$-strip of the $\mu$ content, moving right to left, at each stage pushing a single $k$ through the $\nu$ content. At the end of this step, the $\mu$ content of $1$’s through $(k - 1)$’s lies in a partition shape above the skew shape of the $\nu$ content, and below the $\nu$ content, there is a single horizontal strip of $k$’s, which is part of the $\mu$ content.

We then repeat this pushing down process on the $(k - 1)$ strip of the $\mu$ content, starting with the rightmost, highest $k - 1$, then the $(k - 2)$ strip, etc. At the end of this process, the $\nu$ content forms a partition shape, with the $\mu$ content forming a skew shape below it.

Algorithm 2 “Push Up” In this case, we begin with the 1-strip of the $\nu$ content. Beginning with the lowest, left-most 1 in the strip, use upper evacuation to push the 1 up through the $\mu$ content, until it occupies the top left corner of the diagram. Repeat with the rest of the 1-strip, working left to right, bottom to top. Do the same for the 2 strip, etc., until the $\nu$ content forms a partition shape, with the $\mu$ content forming a skew shape below it.

We show below that both the Push Down and the Push Up algorithms result in the same resulting output, namely, a ptableau in the tensor product $B_\nu \otimes B_\mu$ of highest weight, in other words, the commutator of $S$.

Theorem 8.9 Let $B_\mu$ and $B_\nu$ denote arbitrary irreducible crystals of ptableaux of highest weights $\mu$ and $\nu$, respectively. Let $T_{\mu \text{max}} \in B_\mu$ be the highest weight ptableau in $B_\mu$, and let $T \in B_\nu$ be such that $T_{\mu \text{max}} \otimes T \in B_\mu \otimes B_\nu$ is also highest weight. Then, applying either the Push Up or the Push Down algorithm to $T_{\mu \text{max}} \otimes T$ produces the unique highest weight ptableau $T_{\nu \text{max}} \otimes T' \in B_\nu \otimes B_\mu$.

Proof The Push Down algorithm pushes $\mu$ content through the $\nu$ content by outer evacuation, keeping the $\mu$ content BSS perforated. Lemma 8.3 shows that pushing $\mu$ content down can be accomplished by performing $e_i$ crystal operators on $T$. Thus, not only is the image of $T$ a perforated tableau during the Push Down algorithm, its image lies in the same crystal $B_\nu$, and not merely one isomorphic to it. Similarly, the Push Up algorithm moves the $\nu$ content in $T$ so that it is BSS perforated, and will have the effect as applying $f_i$ crystal operators on $T_{\mu \text{max}}$, taking it to some ptableau in the same crystal $B_\mu$.

Consider the Push Down algorithm applied to $T_{\mu \text{max}} \otimes T$. By Lemma 8.6, each box in the $k$-strip of the $\mu$ content will arrive, via outer evacuation, to the right of the boxes of the $k$-strip that came before it. Thus, as we move the $k$-strip down through the $\nu$ content, it forms a horizontal strip and, in particular, the $\mu$ content formed by the Push Down algorithm forms a ptableau. From this, it is clear that at each stage of the Push Down algorithm...
(including when a box of $\mu$ content is in the interior of the $\nu$ content), the resulting tableau forms a perforated BSS pair. So, the Push Down algorithm (defined via outer evacuation) is equivalent to performing BSS tableau switching.

An analogous argument implies that the Push Up algorithm is also a special case of BSS tableau switching. However, Theorem 2.2 of [3] demonstrates that the resulting tableau obtained by BSS tableau switching is independent of the sequence of switches used. Thus, the outputs of the Push Up and the Push Down algorithms are the same.

Hence, while we start with the tensor product $T_{\mu \max} \otimes T$, with $T_{\mu \max}$ highest weight in $B_\mu$ and $T \in B_\nu$, the resulting ptableau (from either algorithm) is a tensor product $T_{\nu \max} \otimes T'$ with $T_{\nu \max}$ highest weight in $B_\nu$ and $T' \in B_\mu$. Since the action of the Push Algorithms applies crystal operators to either the $\mu$ or the $\nu$ content, and the action of both is a BSS switching, the result of either algorithm is the same. BSS switching preserves shape, so the resulting ptableau is highest weight of the same weight $\lambda = \text{wt}(T_{\mu \max} \otimes T)$.

\[9 \text{ Questions for Future Work}\]

1. **Insertion Algorithms.** There is an elegant version of “RSK”-style results, but couched in the language of ptableaux. In [2] we show that an insertion algorithm on a ptableau $T$ can produce a pair of ptableaux $(P(T), T_{\max})$ such that (1) Given any two ptableaux $T$ and $T'$, if $P(T) = P(T')$, then $T$ and $T'$ are plactically equivalent. (2) $T_{\max}$ is the highest weight element in the irreducible crystal graph containing $T$, and (3) There are simple algorithms to convert $P(T)$ to a SSYT $P$ and $T_{\max}$ to a SSYT $Q$ such that $(P, Q)$ is the RSK pair associated to the word $\omega$, with $\text{Perf}(\omega) = T$. This is natural in the sense that, for example, instead of using $Q$ as a proxy for determining if two ptableaux lie in the same irreducible crystal graph, an insertion algorithm computes their respective highest weights directly.

2. **Commutators.** The results obtained in [2] have been used to give a combinatorial algorithm to compute commutators for arbitrary ptableaux, extending the results on highest weights obtained here.

3. **Other Types.** The successes in type $A_{n-1}$ suggest looking for appropriate perforated tableau models in other types, by looking at tensor products of the standard crystals there. We have determined ptableaux models in all classical types and are investigating their usefulness in those settings.

**Funding** Not applicable

**Availability of Data and Material** Not applicable

**Code Availability** Not applicable

**Declarations**

**Conflicts of Interest/Competing Interests** Not applicable
1. Appleby, G., Whitehead, T.: Perforated Tableaux: A Combinatorial Model for Crystal Graphs in Type $A_{n−1}$. arXiv:2007.11721 (2021)
2. Appleby, G., Whitehead, T.: RSK and PTableaux (in preparation) (2022)
3. Benkart, G., Sottile, F., Stroomer, J.: Tableau switching: algorithms and applications. J. Combin. Theory Ser. A 76, 11–43 (1996)
4. Berenstein, A., Zelevinsky, A.: Canonical bases for the quantum group of type $a_r$ and piecewise-linear combinatorics. Duke Math. J. 82(3), 473–502 (1996)
5. Bump, D., Schilling, A.: Crystal Bases, Representations and Combinatorics. World Scientific, Hackensack (2017)
6. Fu, A., Lascoux, A.: Non-symmetric Cauchy kernels for the classical groups. JCTA 116, 903–917 (2009)
7. Fulton, W.: Young Tableaux. Cambridge University Press, Cambridge (1997)
8. Gerber, T., Lecouvey, C.: Duality and bicrystals on infinite binary matrices. arXiv:2009.10397 (2020)
9. Henriques, A., Kamnitzer, J.: The octahedron recurrence and $\mathfrak{gl}_n$ crystals. Adv. Math. 206(1), 211–249 (2006)
10. Heo, T., Kwon, J.: Combinatorial Howe duality of symplectic type. arXiv:2008.05093 (2020)
11. James, G., Kerber, A.: Representation Theory of the Symmetric Group. Addison-Wesley, New York (1982)
12. Kashiwara, M., Nakashima, T.: Crystal graphs for representations of the $q$-analogue of classical Lie algebras. J. Algebra 165(2), 295–345 (1994)
13. Lee, J.: Crystal structure on King tableaux and semistandard oscillating tableaux. arXiv:1910.04459 (2019)
14. Lenart, C.: On the combinatorics of crystal graphs, I. Lusztig’s Involution. Adv. Math. 211(1), 204–243 (2007)
15. Lenart, C.: On the combinatorics of crystal graphs, II. The crystal commutator. Proc. Am. Math. Soc. 136, 825–837 (2008)
16. Nakayashiki, A., Yamada, Y.: Kostka polynomials and energy functions on solvable lattice models. Sel. Math. 3, 547–599 (1997)
17. Shimozono, M.: Crystals for dummies. https://www.aimath.org/WWN/kostka/crysdumb.pdf (2005)
18. van Leeuwen, M.: Double crystals of binary and integral matrices. Electron. J. Comb. 13 (2006)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.