Orthosymplectic Cauchy Identities

Aalekh Patel, Harsh Patel and Anna Stokke

Abstract. We give bijective proofs of orthosymplectic analogues of the Cauchy identity and dual Cauchy identity for orthosymplectic Schur functions. To do so, we present two insertion algorithms; these are orthosymplectic versions of Berele’s symplectic insertion algorithms, which were used by Sundaram to give bijective proofs of Cauchy identities for symplectic Schur functions.

1. Introduction

The Robinson–Schensted–Knuth (RSK) correspondence, which gives a bijection between generalized permutations and pairs of semistandard tableaux \((P, Q)\) of the same shape, yields a direct proof of the Cauchy identity for Schur functions [5]. Using an insertion algorithm for symplectic tableaux developed by Berele [2], Sundaram proved a Cauchy identity for the characters of irreducible representations of symplectic groups in [13].

Benkart et al. worked with representations of orthosymplectic Lie superalgebras spo\((2m, n)\) in [1] where they introduced spo-tableaux as hybrids of symplectic tableaux and semistandard tableaux to give combinatorial descriptions of characters of spo\((2m, n)\)-representations. They also gave algebraic proofs of various identities for orthosymplectic characters, including Jacobi–Trudi-type formulae and an analogue of the Cauchy identity.

In [12] an orthosymplectic Jacobi–Trudi identity is proved combinatorially using lattice path arguments. In [11], the third author proved a Pieri rule, which describes the integer coefficients of the orthosymplectic characters that appear in the expansion of the product of two orthosymplectic characters, one of which is associated to a one-row tableau. In the current paper, we give bijective proofs of orthosymplectic analogues of the Cauchy identity and dual Cauchy identity.

This research was supported by NSERC Grant RGPIN-2018-05877.
After establishing preliminaries in Sect. 2, we recall the jeu de taquin algorithm for spo-tableaux from [1] and then define an spo-insertion algorithm in Sect. 3. An spo-insertion algorithm is also given in [1] and used in [11]; the one we describe here is slightly different, but both algorithms are hybrids of regular RSK-insertion and the Berele insertion algorithm [2,13]. In Sect. 4, we introduce the spo-correspondence, which yields a bijection between a set \( \mathcal{A} \) of two-line arrays defined in Sect. 2 and triples that consist of an spo-tableau, a semistandard tableau of the same shape and Burge two-line arrays. This spo-correspondence is the main tool that we use to establish an orthosymplectic Cauchy identity. In Sect. 5 we give a dual spo-correspondence, and use it to give a direct proof of an orthosymplectic analogue of the dual Cauchy identity.

2. Preliminaries

A partition of a positive integer \( N \) is a \( k \)-tuple of positive integers \( \lambda = (\lambda_1, \ldots, \lambda_k) \) with \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k \) and \( |\lambda| = \sum_{i=1}^{k} \lambda_i = N \). The Young diagram of shape \( \lambda \) contains \( N \) boxes in \( k \) left-justified rows with \( \lambda_i \) boxes in the \( i \)th row. The length of the Young diagram of shape \( \lambda \) is the number of rows in \( \lambda \), denoted \( \ell(\lambda) \). The conjugate or transpose of \( \lambda \) is the partition \( \lambda^t = (\lambda_1^t, \lambda_2^t, \ldots, \lambda_s^t) \) where \( \lambda_i^t \) is the number of boxes in the \( i \)th column of the Young diagram of shape \( \lambda \). A partition is said to be even if the number of boxes in each row is even.

A \( \lambda \)-tableau is obtained by filling the boxes of the Young diagram of shape \( \lambda \) with entries from a set of positive integers \( \{1, 2, \ldots, n\} \). A \( \lambda \)-tableau is semistandard if the entries in the rows are weakly increasing from left to right and the entries in the columns are strictly increasing from top to bottom. Given a \( \lambda \)-tableau \( T \), we will denote \( \text{sh}(T) = \lambda \).

The weight of a \( \lambda \)-tableau \( T \) is a monomial in the variables \( X = \{x_1, \ldots, x_n\} \) defined as \( \text{wt}(T) = \prod_{i=1}^{n} x_i^{\alpha_i} \) where \( \alpha_i \) denotes the number of appearances of the entry \( i \) in \( T \). The Schur function corresponding to \( \lambda \) is defined as \( s_{\lambda}(X) = \sum_{T} \text{wt}(T) \), where the sum runs over the semistandard \( \lambda \)-tableaux with entries in \( \{1, \ldots, n\} \).

Given partitions \( \lambda \) and \( \mu \), we say that \( \mu \subseteq \lambda \) if \( \mu_i \leq \lambda_i \), for \( i \geq 1 \). A skew diagram of shape \( \lambda/\mu \) is obtained by removing the Young diagram of shape \( \mu \) from the top-left corner of the Young diagram \( \lambda \). A skew tableau is obtained by filling a skew diagram with the entries from a set \( \{1, 2, \ldots, n\} \). A semistandard skew tableau is a skew tableau in which the entries in the rows weakly increase from left to right and the entries in the columns strictly increase from top to bottom. The weight of a skew tableau and skew Schur function are defined similar to the above.

Let \( B_0 = \{1, \overline{1}, 2, \overline{2}, \ldots, m, \overline{m}\} \) with the ordering \( 1 < \overline{1} < 2 < \overline{2} < \cdots < m < \overline{m} \) and suppose that \( \lambda \) is a partition with \( \ell(\lambda) \leq m \). A semistandard symplectic \( \lambda \)-tableau has entries in \( B_0 \), is semistandard with respect to the above ordering, and satisfies the symplectic condition, which requires that the entries in the \( i \)th row of \( T \) are greater than or equal to \( i \), for each \( 1 \leq i \leq m \).
If an entry $j$ appears in row $i$ where $j < i$, we refer to this as a symplectic violation.

The weight of a symplectic $\lambda$-tableau $T$ is a monomial in the entries $X = \{x_1, x_1^{-1}, \ldots, x_m, x_m^{-1}\}$ and is given by $\text{wt}(T) = \prod_{i=1}^{m} x_i^{a_i - a_i^\ast}$ where $a_i$ (respectively $a_i^\ast$) is equal to the number of entries equal to $i$ (respectively $i^\ast$) in $T$. The symplectic Schur function is defined as

$$
\text{sp}_\lambda(X) = \sum \text{wt}(T),
$$

where the sum runs over the semistandard symplectic $\lambda$-tableaux $T$ with entries in $X$.

Let $B_1 = \{1^\circ, 2^\circ, \ldots, n^\circ\}$ and order $B_0 \cup B_1$ as follows:

$$
1 < 1^\circ < 2 < 2^\circ < \cdots < m < m^\circ < 1^\circ < 2^\circ < \cdots < n^\circ.
$$

An spo$(2m, n)$-tableau $T$ of shape $\lambda$ (or an spo-tableau) is a filling of the Young diagram of shape $\lambda$ with entries from $B_0 \cup B_1$ such that the portion $S$ of $T$ that contains entries only from $B_0$ is a semistandard symplectic tableau. Furthermore, the skew tableau formed by removing $S$ from $T$, which contains only entries from $B_1$, is strictly increasing across the rows from left to right and weakly increasing down columns from top to bottom.

Let $T = \{t_1, \ldots, t_n\}$ and $Z = \overline{X} \cup T$. The orthosymplectic character corresponding to $\lambda$ is given by

$$
\text{spo}_\lambda(Z) = \sum_{\mu \subseteq \lambda} \text{sp}_\mu(\overline{X}) s_{\lambda^t/\mu^t}(T),
$$

where $\text{sp}_\mu(\overline{X})$ is a symplectic Schur function in $\overline{X}$ and $s_{\lambda^t/\mu^t}(Y)$ is a skew Schur polynomial in $T$.

To define an orthosymplectic character in terms of spo-tableaux, define the weight of an spo-tableau $T$ by replacing each entry $i \in \{1, \ldots, m\}$ in $T$ with $x_i$, each $i^\ast \in \{1, \ldots, m\}$ in $T$ with $x_i^{-1}$ and each $i^\circ \in \{1^\circ, \ldots, n^\circ\}$ in $T$ with $t_i$ and let $\text{wt}(T)$ be the product of these variables. We have $\text{spo}_\lambda(Z) = \sum T \text{wt}(T),$ where the sum is over the spo-tableaux of shape $\lambda$ with entries from $Z$ [1, Theorem 5.1].

**Example 2.1.** Let $\lambda = (2)$, $m = 1$ and $n = 2$. Then $\overline{X} = \{x_1, x_1^{-1}\}$, $T = \{t_1, t_2\}$. The spo-tableaux are

$$
\begin{array}{|c|}
\hline
1 & 1^\circ \\
\hline
\end{array},
\begin{array}{|c|}
\hline
1 & T \\
\hline
\end{array},
\begin{array}{|c|}
\hline
1 & 1^\circ \\
\hline
\end{array},
\begin{array}{|c|}
\hline
1 & 2^\circ \\
\hline
\end{array},
\begin{array}{|c|}
\hline
T & T \\
\hline
\end{array},
\begin{array}{|c|}
\hline
T & 1^\circ \\
\hline
\end{array},
\begin{array}{|c|}
\hline
T & 2^\circ \\
\hline
\end{array},
\begin{array}{|c|}
\hline
1^\circ & 2^\circ \\
\hline
\end{array}
$$

and

$$
\text{spo}_\lambda(Z) = x_1^2 + 1 + x_1 t_1 + x_1 t_2 + x_1^{-2} + x_1^{-1} t_1 + x_1^{-1} t_2 + t_1 t_2.
$$

Let $q$ be a positive integer and consider the set of two-line arrays

$$
\pi = \begin{pmatrix} a_1 & a_2 & \ldots & a_t \\ b_1 & b_2 & \ldots & b_t \end{pmatrix}
$$

such that $a_i \in \{1, \ldots, q\}$ and $b_i \in B_0 \cup B_1$. Let $A$ denote the set of such arrays that satisfy the following properties:
(1) \( a_i \leq a_{i+1} \) and if \( a_i = a_{i+1} \) then \( b_i \leq b_{i+1} \);
(2) any column \( (i)_j \) with \( j \in B_1 \) occurs at most once in \( \pi \).

Define \( \mathcal{A}^* \) to be the set of arrays that satisfy:
(1) \( a_i \leq a_{i+1} \) and if \( a_i = a_{i+1} \) then \( b_i \geq b_{i+1} \);
(2) any column \( (i)_j \) with \( j \in B_0 \) occurs at most once in \( \pi \).

A Burge two-line array \( \pi \) has \( a_i \in \{1, \ldots, q\}, b_i \in \{1, \ldots, q - 1\} \) and satisfies
(1) \( a_i \leq a_{i+1} \) and if \( a_i = a_{i+1} \) then \( b_i \leq b_{i+1} \);
(2) \( a_i > b_i \) for \( 1 \leq i \leq t \).

A dual Burge two-line array \( \pi \) has both \( a_i, b_i \in \{1, \ldots, q\} \) and satisfies
(1) \( a_i \leq a_{i+1} \) and if \( a_i = a_{i+1} \) then \( b_i \geq b_{i+1} \);
(2) \( a_i \geq b_i \) for \( 1 \leq i \leq t \).

Example 2.2. Let
\[
\pi_1 = \begin{pmatrix}
1 & 1 & 2 & 2 & 3 & 3 \\
2 & 3 & 3 & 3 & 3 & 3
\end{pmatrix}, \quad \pi_2 = \begin{pmatrix}
1 & 1 & 2 & 2 & 2 & 2 \\
3 & 2 & 2 & 2 & 2 & 2
\end{pmatrix}.
\]

Then \( \pi_1 \in \mathcal{A}, \pi_1 \notin \mathcal{A}^* \), \( \pi_2 \in \mathcal{A}^*, \pi_2 \notin \mathcal{A} \).

3. Orthosymplectic Jeu de taquin and Insertion

A punctured spo-tableau is an spo-tableau with an entry removed from one of its boxes. The jeu de taquin algorithm for spo-tableaux moves an empty box in the first column of a punctured spo-tableau through the tableau using forward slides. A forward slide is defined as follows:

\[
\begin{array}{c}
\begin{array}{c}
\text{if } x < y \text{ or } x = y \text{ and } x, y \in B_1 \\
\begin{array}{c}
\begin{array}{c}
\text{if } y < x \text{ or } x = y \text{ and } x, y \in B_0.
\end{array}
\end{array}
\end{array}
\end{array}
\]

The result of performing a forward slide on a punctured spo-tableau is a punctured spo-tableau and forward slides can be continued, beginning with an empty box in the first column, until the empty box lands in an outer corner [1, Lemmas 5.2 & 5.3], which means there are no boxes directly below it or to its immediate right. The procedure is also reversible and the slides that reverse the procedure are called reverse slides.

Ordinary Robinson–Schensted–Knuth (RSK) row insertion for semistandard \( \lambda \)-tableaux gives an algorithm for inserting a letter into a semistandard tableau and returns a new semistandard tableau with one more box (see [4,8–10]). The procedure is as follows for a semistandard \( \lambda \)-tableau \( T \) and a positive integer \( x \): if \( x \) is greater than or equal to all entries in the first row, add \( x \) in a new box at the end of the first row, which yields a new semistandard tableau \( T \leftarrow x \). Otherwise, remove the left-most entry in the first row that is strictly larger than \( x \), place \( x \) in that box, and bump the displaced entry into the second row by repeating the procedure with the displaced positive integer in the second row so that it is placed either at the end of the second row or replaces an
entry in the second row. Continue this process for successive rows until a new box is created at the end of a row or at the bottom of the tableau, forming a new row. The final step gives a new semistandard tableau \( S = T \leftarrow x \) and the added box belongs to an outer corner of the Young diagram. The row insertion algorithm can be reversed; starting with a box in an outside corner, the entry \( x \) in that box replaces \( x' \) in the row above it, where \( x' \) is the rightmost entry in the row with \( x' < x \) and the procedure continues until an entry is bumped out of the first row.

Ordinary RSK-column insertion is similar except that \( x \) is added to the end of the first column if it is larger than or equal to all entries in the column, and otherwise replaces the smallest entry \( y \) that is strictly larger than it in the first column and bumps \( y \) into the second column. In this case the process continues until a new box is created at the end of a column or until a new column is formed with one entry, with the new box necessarily appearing in an outer corner.

We will also use dual row insertion of a letter \( x \) into a tableau \( T \) where the transpose \( T^t \) is semistandard. In this case, \( x \) is appended to the end of the first row of \( T \) if \( x \) is larger than or equal to every entry in the row. Otherwise, \( x \) replaces the least entry \( x' \) in the first row that satisfies \( x' \geq x \). The entry \( x' \) is then bumped into the second row of the tableau, and the procedure repeats until a new box is created at the end of a row or at the bottom of the tableau. This results in a new tableau \( S = x \rightarrow T \) such that \( S^t \) is semistandard.

We now define an insertion algorithm for spo-tableaux. Our algorithm is slightly different than the insertion algorithm in [1,11], where column insertion is applied to elements from \( B_1 \). The algorithm we describe here instead uses dual row insertion for elements from \( B_0 \), which ensures that the second tableau in the triple we define in Sect. 4 is semistandard. To insert an entry \( x \in B_0 \cup B_1 \) into an spo-tableau \( T \), begin by considering the first row.

1. If \( x \in B_0 \) and \( x \) is larger than or equal to every entry in the first row or if \( x \in B_1 \) and \( x \) is strictly larger than every entry in the first row, append \( x \) to the end of the first row, giving \( T \leftarrow x \).
2. Otherwise, if \( x \in B_1 \), find the leftmost entry \( x' \) in the first row with \( x' \geq x \) and replace this entry with \( x' \), bumping \( x' \). If \( x \in B_0 \) find the leftmost entry \( x' \) in the first row with \( x' > x \), replace \( x' \) with \( x \), bumping \( x \) from the first row.
3. Insert \( x' \) into the second row by repeating the procedure and continue for successive rows unless a symplectic violation occurs.
4. If no symplectic violation is introduced at any point in the process, the procedure ends when a new box is created at the end of a row or at the bottom of the tableau. The resulting tableau is \( T \leftarrow x \).
5. On the other hand, if a symplectic violation is introduced at some point in the procedure, then for some \( i \), an \( i \) is bumped out of row \( i \) and into row \( i+1 \) by an \( i \). The Berele insertion algorithm is applied at this point: find the least \( i \) for which a violation occurs and, at this stage, instead of replacing the \( i \) with an \( i \), remove the \( i \) from the tableau. Apply jeu de
taquin forward slides to the empty box until it lands in an outer corner, giving an spo-tableau $T \leftarrow x$ that has one less box than $T$.

**Lemma 3.1.** Let $T$ be an spo-tableau and let $b \in B_0 \cup B_1$. Then $T \leftarrow b$ is an spo-tableau.

**Proof.** The design of the algorithm preserves the row conditions. Suppose an entry $y$ bumps an entry $x$ from a given row and $x \in B_0$; if there is an entry $z$ below $x$ then $z > x$ so insertion of $x$ into the subsequent row either places $x$ in the same column or in a column to the left, which necessarily has entries above it that are strictly less than $x$ since $y < x$. On the other hand, if $x \in B_1$ and there is an entry $z$ below it, then $x \leq z$ and insertion of $x$ into the subsequent row places $x$ in the same column or to the left, with entries above it weakly less than $x$.

If a symplectic violation occurs at some point in the insertion process, then $T \leftarrow b$ is an spo-tableau since the output of the jeu de taquin process is an spo-tableau. □

**Example 3.2.** This example illustrates the insertion process when the entry $\bar{1}$ is inserted into the spo-tableau $T$.

$T = \begin{array}{ccc}
1 & 2 & 1^o \\
2 & 3 & 2^o \\
4 & 4 & 2^o \\
5 & & \\
\end{array} \leftarrow \begin{array}{ccc}
1 & 1 & 1^o \\
2 & 2 & 3^o \\
4 & 4 & 2^o \\
5 & & \\
\end{array} \leftarrow 2$

4. Establishing the Orthosymplectic Cauchy Identity

In this section we give a bijection, which we call the spo-correspondence, between the set $\mathcal{A}$ of two-line arrays defined in Sect. 2 and the set of triples $(\tilde{P}_\lambda, P_\lambda, L)$, where $\tilde{P}_\lambda$ is an spo-tableau, $P_\lambda$ a semistandard $\lambda$-tableau and $L$ is a Burge two-line array. Applying the bijection between Burge two-line arrays and semistandard tableaux $Q_\beta$, with even length columns as in [13] then gives a bijection between $\mathcal{A}$ and triples $(\tilde{P}_\lambda, P_\lambda, Q_\beta)$, which establishes an orthosymplectic analogue of the Cauchy identity.

Given a two-line array $\pi \in \mathcal{A}$ with $s$ columns, we perform the steps described below to obtain a triple $(P_\lambda, P_\lambda, L)$.

1. Set $\tilde{P}_0 = P_0 = L_0 = \phi$.
2. Working from left to right, and beginning with the first column, consider the $k^{th}$ column in $\pi$, with top entry $i$ and bottom entry $j$. To obtain $(\tilde{P}_k, P_k, L_k)$ from $(\tilde{P}_{k-1}, P_{k-1}, L_{k-1})$, apply the spo-insertion algorithm to get $\tilde{P}_k = \tilde{P}_{k-1} \leftarrow j$. This operation either adds or deletes exactly one box in $\tilde{P}_{k-1}$.
   a) If a box is added by the insertion of $j$, form $P_k$ by adding a new box to $P_{k-1}$ containing $i$ at the same position where a new box was added to $\tilde{P}_{k-1}$ to get $\tilde{P}_k$. This creates $P_k$ with the same shape as that of $\tilde{P}_k$. In this case, let $L_k = L_{k-1}$.
(b) If $\tilde{P}_k$ has one less box than $\tilde{P}_{k-1}$, the extra box in $\tilde{P}_{k-1}$ occurs in an outer corner. Locate the position of the extra box in $\tilde{P}_{k-1}$ and perform reverse column bumping on the element that belongs to the extra box to get $P_k$, which has the same shape as $\tilde{P}_k$. This will bump an entry $a$ from the first column. Add the column $\binom{a}{i}$ on the right of $L_{k-1}$ to obtain $L_k$.

(3) Let $\tilde{P} = \tilde{P}_s$, $P = P_s$ and $L = L_s$, obtaining the triple $(\tilde{P}_\lambda, P_\lambda, L)$.

**Example 4.1.** Let

$$\pi = \left( \begin{array}{ccccc}
2 & 3 & 4 & 4 & 4 \\
1 & 2 & 1 & 1 & 2
\end{array} \right)$$

The tableaux and two-line arrays obtained at each step of the algorithm are listed below.

$\tilde{P}_1 = \begin{array}{|c|}
\hline
\text{T} \\
\hline
\end{array}$ \hspace{1cm} $P_1 = \begin{array}{|c|}
\hline
2 \\
\hline
\end{array}$ \hspace{1cm} $L_1 = \phi$

$\tilde{P}_2 = \begin{array}{|c|}
\hline
\text{T} & 1^\circ \\
\hline
\end{array}$ \hspace{1cm} $P_2 = \begin{array}{|c|}
\hline
2 & 2 \\
\hline
\end{array}$ \hspace{1cm} $L_2 = \phi$

$\tilde{P}_3 = \begin{array}{|c|}
\hline
\text{T} & 1^\circ & 2^\circ \\
\hline
\end{array}$ \hspace{1cm} $P_3 = \begin{array}{|c|}
\hline
2 & 2 & 2 \\
\hline
\end{array}$ \hspace{1cm} $L_3 = \phi$

$\tilde{P}_4 = \begin{array}{|c|}
\hline
1^\circ & 2^\circ \\
\hline
\end{array}$ \hspace{1cm} $P_4 = \begin{array}{|c|}
\hline
2 & 2 \\
\hline
\end{array}$ \hspace{1cm} $L_4 = \left( \begin{array}{c}
3 \\
2
\end{array} \right)$

$\tilde{P}_5 = \begin{array}{|c|}
\hline
\text{T} & 2^\circ \\
\hline
1^\circ \\
\hline
\end{array}$ \hspace{1cm} $P_5 = \begin{array}{|c|}
\hline
2 & 2 \\
\hline
\end{array}$ \hspace{1cm} $L_5 = \left( \begin{array}{c}
3 \\
2
\end{array} \right)$

$\tilde{P}_6 = \begin{array}{|c|}
\hline
1^\circ & 2^\circ \\
\hline
\end{array}$ \hspace{1cm} $P_6 = \begin{array}{|c|}
\hline
2 & 2 \\
\hline
\end{array}$ \hspace{1cm} $L_6 = \left( \begin{array}{c}
3 & 4 \\
2 & 3
\end{array} \right)$

$\tilde{P}_7 = \begin{array}{|c|}
\hline
1^\circ & 2^\circ \\
\hline
1^\circ \\
\hline
\end{array}$ \hspace{1cm} $P_7 = \begin{array}{|c|}
\hline
2 & 2 \\
\hline
\end{array}$ \hspace{1cm} $L_7 = \left( \begin{array}{c}
3 & 4 \\
2 & 3
\end{array} \right)$

$\tilde{P}_8 = \begin{array}{|c|}
\hline
1 & 1^\circ & 2^\circ \\
\hline
1^\circ & 2^\circ \\
\hline
\end{array}$ \hspace{1cm} $P_8 = \begin{array}{|c|}
\hline
2 & 2 \\
\hline
\end{array}$ \hspace{1cm} $L_8 = \left( \begin{array}{c}
3 & 4 \\
2 & 3
\end{array} \right)$

$\tilde{P}_9 = \begin{array}{|c|}
\hline
1 & 1^\circ & 2^\circ \\
\hline
1^\circ & 2^\circ \\
\hline
\end{array}$ \hspace{1cm} $P_9 = \begin{array}{|c|}
\hline
2 & 2 & 4 \\
\hline
\end{array}$ \hspace{1cm} $L_9 = \left( \begin{array}{c}
3 & 4 \\
2 & 3
\end{array} \right)$

The image of $\pi$ is $(\tilde{P}_9, P_9, L_9)$.

**Lemma 4.2.** Suppose that $T$ is an spo-tableau and that $x, x' \in B_0 \cup B_1$ with $x \leq x'$ if $x, x' \in B_0$ and $x < x'$ otherwise. Suppose that $T \leftarrow x$ has one more box $B$ than $T$ and that $(T \leftarrow x) \leftarrow x'$ has one more box $B'$ than $T \leftarrow x$. Then $B'$ is strictly right of $B$.

**Proof.** Since each insertion adds a box to the starting tableau, the spo-insertion process is similar in this case to regular row insertion, except that an entry $y \in B_1$ may bump and replace an identical entry $y$ from a row during the process. Since $x < x'$ if either element belongs to $B_1$, the proof involves a slight adjustment of the one for regular insertion (see, for example [4]). If $x$ bumps an entry $y$ from the first row and $x'$ subsequently bumps $y'$ from the
first row then, since $x \leq x'$, $y'$ must belong to a box that is strictly right of $y$ in the first row of $T$. Furthermore, if $y, y' \in B_0$ then $y \leq y'$ and otherwise $y < y'$ so the same reasoning applies to subsequent rows, which inductively gives the result.

The following two lemmas are needed to establish Theorem 4.5 and are orthosymplectic versions of [13, Lemmas 3.3 and 3.2]. We omit the proofs since they are essentially the same as those given in [13], adjusted slightly to accommodate jeu de taquin slides that involve entries from $B_1$.

**Lemma 4.3.** Let $T$ be an spo-tableau and let $x, x' \in B_0$ with $x \leq x'$. Suppose that spo-insertion of $x$ into $T$ causes a cancellation and spo-insertion of $x'$ into $T \leftarrow x$ also causes a cancellation. Then the jeu de taquin path of the empty box created by inserting $x$ ends in the same row or higher than the jeu de taquin path of the empty box created by insertion of $x'$ into $T \leftarrow x$.

**Lemma 4.4.** Suppose that $T$ is an spo-tableau and let $x, x' \in B_0$ with $x \leq x'$. If spo-insertion of $x'$ into $T \leftarrow x$ causes a cancellation then so did the prior spo-insertion of $x$ into $T$.

**Theorem 4.5.** The spo-correspondence yields a triple $(\tilde{P}_\lambda, P_k, L)$, where $\tilde{P}_\lambda$ is an spo-tableau, $P_k$ is semistandard and $L$ is a Burge two-line array.

**Proof.** Lemma 3.1 guarantees that $\tilde{P}_\lambda$ is an spo-tableau. If $P_{k-1}$ is semistandard and $P_k$ has one less box than $P_{k-1}$, then $P_k$ is also semistandard since reverse column insertion preserves semistandard properties. If a box is added to $P_{k-1}$ to form $P_k$, then it is added at an outer corner, since the corresponding box added to $\tilde{P}_{k-1}$ to form $\tilde{P}_k$ is at the same outer corner. Since the entries in the top row of $\pi$ are weakly increasing, the rows and columns of the associated $P_k$ are weakly increasing. Further, whenever $a_i$ and $a_{i+1}$ in the top row of $\pi$ are equal, the corresponding bottom entries $b_i$ and $b_{i+1}$ satisfy $b_i \leq b_{i+1}$ and $b_i \neq b_{i+1}$ if $b_i, b_{i+1} \in B_1$, so it follows from Lemma 4.2 that the columns of $P_k$ are strictly increasing.

The proof that $L$ is a two-line Burge array is the same as in the symplectic case, which is covered in [13]. Suppose that $L = \begin{pmatrix} i_1 & i_2 & \cdots & i_r \\ j_1 & j_2 & \cdots & j_r \end{pmatrix}$. The entries in the top row of $L$ are weakly increasing by construction. Any entry $i_k$ in the top row of $L$ records the top entry in $\pi$ that corresponds to a cancellation when the entry below it in $\pi$ is inserted into some $\tilde{P}_t$. The entry $j_k$ belongs to $P_t$ so it appears to the left of $i_k$ (so to the left of column $t$) in the top row of $\pi$ so $j_k \leq i_k$. If $i_k = j_k = a$ then there is at least one additional entry $a$ in the top row of $\pi$ left of column $t$ and, since $j_k = a$, $a$ bumps out of $P_t$. By Lemma 4.4 all entries in the bottom row of $\pi$ to the left of column $t$ for which there is an $a$ in the top row must cause a cancellation so there cannot be any entries equal to $a$ in $P_t$ so $j_k < i_k$.

Finally, suppose that $i_k = i_{k+1} = a$ and let $b$ and $b'$ denote the entries below $i_k$ and $i_{k+1}$ in $\pi$ respectively. Both entries cause cancellations and by Lemma 4.3, the jeu de taquin path for $b'$ is in a weakly lower row than that of $b$. Then the corresponding column bumping path in $P_{t+1}$, which bumps $j_{k+1}$
out of the first column begins in a row weakly lower than the column bumping path in \( P \), which bumps \( j_k \) out of the first column. By a property of column insertion [9], \( j_k \leq j_{k+1} \).

To show that the spo-correspondence is invertible, we describe an algorithm below that maps a triple \((\tilde{P}_\lambda, P_\lambda, L)\), where \( \tilde{P}_\lambda \) is an spo-tableau, \( P_\lambda \) a semistandard tableau and \( L \) a Burge two-line array to a two-line array \( \pi \in A \). Given a two-line array \( \pi \), the spo-correspondence produces a triple \((\tilde{P}_\lambda, P_\lambda, L)\) where the first row of \( L \) records entries from the top row of \( \pi \) that correspond to cancellations, while the entries in \( P_k \) are entries from the top row of \( \pi \) that correspond to additions at each step. If an entry appears in both \( P_k \) and in \( L_k \), Lemma 4.4 tells us that the cancellations occurred prior to the additions. Lemma 4.2 ensures that the order in which elements from the bottom row of \( \pi \) were inserted into \( \tilde{P}_k \) can be determined from \( P_k \). It thus follows that the map we now describe is the inverse of the above map.

1. Set \( \tilde{P}_s = \tilde{P}_\lambda, P_s = P_\lambda, L_s = L \) and \( \pi_s = \phi \), where \( n \) is the total number of entries in \( P_\lambda \) and \( L \) combined.
2. Beginning with \( k = s \), to obtain \((\tilde{P}_{k-1}, P_{k-1}, L_{k-1})\) and \( \pi_{k-1} \) from \((\tilde{P}_k, P_k, L_k)\) and \( \pi_k \), find the largest entry \( i \) among all the entries in \( P_k \) and top row of \( L_k \).
   (a) If \( i \) belongs to \( P_k \), then select the rightmost entry \( i \) in \( P_k \) and delete the box containing \( i \) to get \( P_{k-1} \). Since \( \tilde{P}_k \) and \( P_k \) are of the same shape, \( \tilde{P}_k \) and \( P_{k-1} \) differ by exactly one box. Perform reverse spo-insertion on the entry in the extra box in \( \tilde{P}_k \) to obtain \( \tilde{P}_{k-1} \) and suppose this bumps \( j \) from the first row. Prepend the column \((i, j)\) to \( \pi_k \) to get \( \pi_{k-1} \) and let \( L_{k-1} = L_k \).
   (b) If \( i \) does not belong to \( P_k \) (so only belongs to \( L_k \)), then select the right-most column \((i, a)\) in \( L_k \) and delete this column from \( L_k \) to get \( L_{k-1} \). Then column insert \( a \) into \( P_k \) to get \( P_{k-1} \). This creates a new box and since \( \tilde{P}_k \) and \( P_k \) have the same shape, \( \tilde{P}_k \) and \( P_{k-1} \) differ by exactly one box. Add an empty box to \( \tilde{P}_k \) in this location and perform reverse jeu de taquin to slide the empty box to the highest possible row \( r \) in the first column. Replace the hole with the row number \( r \), and replace the rightmost \( r \) in that row with \( \pi \). Note that there is at least one \( r \) in the row \( r \), since we placed an entry \( r \) in the empty box. Bump the entry \( r \) into the row above it and perform reverse row bumping until some entry \( j \) gets bumped out of the first row to obtain \( \tilde{P}_{k-1} \). Prepend column \((i, j)\) to \( \pi_k \) to obtain \( \pi_{k-1} \).
3. Let \( \pi = \pi_0 \).

**Example 4.6.** We will apply the inverse algorithm to the triple obtained in Example 4.1. Let

\[
(\tilde{P}_\lambda, P_\lambda, L) = \left( \begin{array}{ccc}
1 & 1^\circ & 2^\circ \\
1^\circ & 2^\circ \\
\end{array} \right), \quad \left( \begin{array}{ccc}
2 & 2 & 4 \\
2 & 4 \\
4 & 4 \\
\end{array} \right), \quad \left( \begin{array}{c}
3 \\
4 \\
2 \\
3 \\
\end{array} \right)
\]

We obtain the following tableaux and two-line arrays.

\[
\begin{align*}
\tilde{P}_9 &= \begin{array}{ccc}
1 & 1^\circ & 2^\circ \\
1^\circ & 2^\circ & \\
\end{array} \\
\tilde{P}_8 &= \begin{array}{ccc}
1 & 1^\circ & \\
1^\circ & 2^\circ & \\
\end{array} \\
\tilde{P}_7 &= \begin{array}{ccc}
1 & 2^\circ & \\
1^\circ & \\
\end{array} \\
\tilde{P}_6 &= \begin{array}{ccc}
1^\circ & 2^\circ & \\
& \\
\end{array} \\
\tilde{P}_5 &= \begin{array}{ccc}
1^\circ & \\
& 2^\circ \\
\end{array} \\
\tilde{P}_4 &= \begin{array}{ccc}
1^\circ & 2^\circ & \\
& 2^\circ & \\
\end{array} \\
\tilde{P}_3 &= \begin{array}{ccc}
1 & 1^\circ & 2^\circ \\
& 2 & 2 & 2 \\
\end{array} \\
\tilde{P}_2 &= \begin{array}{ccc}
1^\circ & 1^\circ & \\
& 2 & 2 \\
\end{array} \\
\tilde{P}_1 &= \begin{array}{ccc}
1^\circ & 1^\circ & \\
& 2 \\
\end{array} \\
\tilde{P}_0 &= \begin{array}{ccc}
1^\circ & 1^\circ & 2^\circ \\
& 2 & 2 & 2 \\
\end{array}
\end{align*}
\]

\[
P_9 = \begin{array}{ccc}
2 & 2 & 4 \\
4 & 4 & \\
\end{array} \\
P_8 = \begin{array}{ccc}
2 & 2 & \\
4 & 4 & \\
\end{array} \\
P_7 = \begin{array}{ccc}
2 & 2 & \\
4 & \\
\end{array} \\
P_6 = \begin{array}{ccc}
2 & 2 & \\
& 3 \\
\end{array} \\
P_5 = \begin{array}{ccc}
2 & 2 & \\
& 3 \\
\end{array} \\
P_4 = \begin{array}{ccc}
2 & 2 & \\
& 3 \\
\end{array} \\
P_3 = \begin{array}{ccc}
2 & 2 & 2 \\
& 3 & 3 & 4 & 4 & 4 & 4 \\
\end{array} \\
P_2 = \begin{array}{ccc}
2 & 2 & 2 \\
& 3 & 3 & 4 & 4 & 4 & 4 \\
\end{array} \\
P_1 = \begin{array}{ccc}
2 & 2 & 2 \\
& 3 & 3 & 4 & 4 & 4 & 4 \\
\end{array} \\
P_0 = \begin{array}{ccc}
2 & 2 & 2 \\
& 3 & 3 & 4 & 4 & 4 & 4 \\
\end{array}
\]

\[
L_9 = \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix} \quad \pi_9 = \phi
\]

\[
L_8 = \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix} \quad \pi_8 = \begin{pmatrix} 4 \\ 2^\circ \end{pmatrix}
\]

\[
L_7 = \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix} \quad \pi_7 = \begin{pmatrix} 4 & 4 \\ 1^\circ & 2^\circ \end{pmatrix}
\]

\[
L_6 = \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix} \quad \pi_6 = \begin{pmatrix} 4 & 4 & 4 \\ 1 & 1 & 1^\circ & 2^\circ \end{pmatrix}
\]

\[
L_5 = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \quad \pi_5 = \begin{pmatrix} 4 & 4 & 4 \\ 1^\circ & 1 & 1^\circ & 2^\circ \end{pmatrix}
\]

\[
L_4 = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \quad \pi_4 = \begin{pmatrix} 3 & 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1^\circ & 2^\circ \end{pmatrix}
\]

\[
L_3 = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \quad \pi_3 = \begin{pmatrix} 3 & 3 & 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1^\circ & 2^\circ \end{pmatrix}
\]

\[
L_2 = \begin{pmatrix} 2 \\ 2 \\ 3 & 3 & 4 & 4 & 4 & 4 \\ 1^\circ & 1^\circ & 2^\circ & 2^\circ & 2^\circ & 2^\circ & 2^\circ \end{pmatrix}
\]

\[
L_1 = \begin{pmatrix} 2 \\ 2 \\ 3 & 3 & 4 & 4 & 4 & 4 \\ 1^\circ & 2^\circ & 2^\circ & 1^\circ & 1 & 1^\circ & 2^\circ \end{pmatrix}
\]

\[
L_0 = \begin{pmatrix} 2 \\ 2 \\ 3 & 3 & 4 & 4 & 4 & 4 \\ 1^\circ & 1^\circ & 2^\circ & 1^\circ & 1 & 1^\circ & 2^\circ \end{pmatrix}
\]

The final step is to map Burge two-line arrays to semistandard tableaux \(Q_\beta\), where \(\beta^t\) is even, which is achieved using a correspondence due to Burge [3]; this correspondence is also used in [13].

1. Set \(Q_0 = \phi\).
2. Iterate over the columns of \(L\) from left to right, starting with the first column. For the \(k\)th column in \(L\) with \(i\) and \(j\) as top and bottom entries respectively, to obtain \(Q_k\) from \(Q_{k-1}\) perform the following steps:
   1. Insert \(j\) into \(Q_{k-1}\) using ordinary row insertion to get an intermediate tableau \(T = Q_{k-1} \leftarrow j\), which has one more box than \(Q_{k-1}\).
   2. Add a new box containing \(i\) to \(T\) directly below the new box that was created by the row insertion process to obtain \(Q_k\).
3. The image of \(L\) is \(Q_\beta = Q_s\), which is a tableau with even length columns.

**Example 4.7.** Let

\[
L = \begin{pmatrix} 4 & 4 & 5 & 8 & 9 \\ 1 & 3 & 2 & 5 & 4 \end{pmatrix}
\]

We apply the algorithm to obtain \(Q_5\), which is a tableau with even length columns.
Orthosymplectic Cauchy Identities

The inverse algorithm maps a semistandard tableau $Q_\beta$ of shape $\beta$ with even length columns and $2s$ boxes to a Burge two-line array $L$ via the procedure described below.

1. Start by setting $L_s = \phi$ and $Q_s = Q_\beta$.
2. To obtain $L_{k-1}$ and $Q_{k-1}$ from $L_k$ and $Q_k$, and starting with $k = n$, find the rightmost box in $Q_k$ that contains its largest entry $i$ and delete it. Perform reverse row bumping on the entry immediately above the removed box to bump an entry $j$ out of the first row. Let the remaining tableau be $Q_{k-1}$, which has two fewer boxes than $Q_k$ and add the column $\begin{pmatrix} i \\ j \end{pmatrix}$ in $L_k$ to obtain $L_{k-1}$.
3. Continue to obtain $L = L_0$.

Example 4.8. We perform the inverse algorithm on $Q_3 = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 6 \end{bmatrix}$.

$Q_2 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$, $Q_3 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$, $Q_4 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$, $Q_5 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$

The inverse algorithm maps a semistandard tableau $Q_\beta$ of shape $\beta$ with even length columns and $2s$ boxes to a Burge two-line array $L$ via the procedure described below.

(1) Start by setting $L_s = \phi$ and $Q_s = Q_\beta$.
(2) To obtain $L_{k-1}$ and $Q_{k-1}$ from $L_k$ and $Q_k$, and starting with $k = n$, find the rightmost box in $Q_k$ that contains its largest entry $i$ and delete it. Perform reverse row bumping on the entry immediately above the removed box to bump an entry $j$ out of the first row. Let the remaining tableau be $Q_{k-1}$, which has two fewer boxes than $Q_k$ and add the column $\begin{pmatrix} i \\ j \end{pmatrix}$ in $L_k$ to obtain $L_{k-1}$.
(3) Continue to obtain $L = L_0$.

Example 4.8. We perform the inverse algorithm on $Q_3 = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$. Given a two-line array $\pi \in A$, define the weight of $\pi$ as

$$\text{wt}(\pi) = \prod_{1 \leq i \leq q} \prod_{1 \leq j \leq m} \prod_{1 \leq k \leq n} y_i^{\alpha_i} x_j^{\beta_j-\beta^j} t_k^{\gamma_k},$$

where $\alpha_i$ denotes the number of entries $i$ in the top row of $\pi$, $\beta_j$ and $\beta^j$ the number of entries $j$ and $\bar{j}$ respectively in the bottom row of $\pi$ and $\gamma_k$ the number of entries equal to $k^\circ \in B_1$ in the bottom row of $\pi$. For example, the array $\pi$ in Example 4.1, has $\text{wt}(\pi) = y_2^3 y_3^4 x_1 t_1 t_2^2$.

If a column $\begin{pmatrix} i \\ j \end{pmatrix}$ appears $k$ times in $\pi$, where $j \in B_0$, it contributes either $y_i^k x_i^j$ or $y_i^k x_i^{-j}$ to $\text{wt}(\pi)$ and columns $\begin{pmatrix} i \\ j \end{pmatrix}$ where $j \in B_1$ can occur only once in two-line arrays $\pi \in A$. With this in mind, the spo-correspondence between
two-line arrays and triples \((\tilde{P}_\lambda, P_\lambda, Q_\beta)\) yields the following identity:

\[
\sum_\lambda \spo_\lambda(Z)s_\lambda(Y) \sum_{\beta \text{even}} s_\beta(Y) \\
= \prod_{1 \leq i \leq n} (1 + t_i y_j) \prod_{1 \leq j \leq m} (1 - x_i y_j)^{-1} (1 - x_i^{-1} y_j)^{-1}. \\
\]

The left-hand side of the equation enumerates the triples \((\tilde{P}_\lambda, P_\lambda, Q_\beta)\) while the right-hand side enumerates the two-line arrays in \(A\). The Littlewood identity (see [6], [10, Exercise 7.28] or [7, I.5.]) states that

\[
\prod_{1 \leq i < j \leq q} (1 - y_i y_j)^{-1} = \sum_{\beta \text{even}} s_\beta(Y),
\]

so rearranging gives an orthosymplectic analogue of the Cauchy identity [1, Theorem 4.24 (c)].

**Theorem 4.9.** We have

\[
\sum_\lambda \spo_\lambda(Z)s_\lambda(Y) \\
= \prod_{1 \leq i \leq n} (1 + t_i y_j) \prod_{1 \leq j \leq m} (1 - x_i y_j)^{-1} (1 - x_i^{-1} y_j)^{-1} \prod_{1 \leq i < j \leq q} (1 - y_i y_j).
\]

Specializing to \(x_i = t_i = 1\) yields the following Corollary.

**Corollary 4.10.** We have

\[
\sum_\lambda f_{\spo}^\lambda s_\lambda(Y) = \prod_{1 \leq j \leq q} (1 + y_j)^n (1 - y_j)^{-2m} \prod_{1 \leq i < j \leq q} (1 - y_i y_j),
\]

where \(f_{\spo}^\lambda\) denotes the number of \(\spo(2m, n)\)-tableaux of shape \(\lambda\).

A sequence of partitions \(\Lambda = (\emptyset = \lambda^0, \lambda^1, \ldots, \lambda^k = \lambda)\) is an **up-down tableau** of shape \(\lambda\) and length \(k\) if each \(\lambda^i\) has one more box or one less box than \(\lambda^{i-1}\) for \(i = 1, \ldots, k\). An **up-down \((m, n)\)-tableau** is an up-down tableau \(\Lambda\) such that \(\lambda^i_{m+1} \leq n\) for each \(1 \leq i \leq k\). An \(m\)-symplectic up-down tableau \(\Lambda\) is an up-down \((m, n)\)-tableau with \(n = 0\).

The spo-insertion algorithm we defined in Sect. 3 can be adjusted to give a bijection between the set of words of length \(k\) in the letters \(B_0 \cup B_1\) and pairs \((T, \Lambda)\) of spo-tableaux \(T\) of shape \(\lambda\), where \(\lambda \vdash k - 2r\) for some \(r\) with \(0 \leq r \leq [k/2]\), and up-down \((m, n)\)-tableaux \(\Lambda\) of shape \(\lambda\) and length \(k\). The bijection given in [1] also yields Corollary 4.12.

Given a word \(w = w_1 \ldots w_k\), where \(w_i \in B_0 \cup B_1\), let \(P_0 = \emptyset\) and \(\lambda^0 = \emptyset\). For \(1 \leq i \leq k\), let \(P_i = P_{i-1} \leftarrow w_i\) and \(\lambda^i\) be the shape of \(T_i\). Then the word \(w\) is mapped to \((T, \Lambda)\) where \(T = P_k\) and \(\Lambda = (\lambda^0, \lambda^1, \ldots, \lambda^k)\). We give an example of our bijection for completeness.

**Example 4.11.** Let \(w = 1^0 \overline{1} 2^0 \overline{2} 2^0 1^2 \overline{2} 1\).

\[
P_0 = \emptyset\quad \lambda^0 = \emptyset;\quad P_1 = 1^0\quad \lambda^1 = \begin{array}{|c|c|}
\end{array}
\]
The word $w$ maps to $(P_8, \Lambda)$, where $\Lambda = (\lambda^0, \lambda^1, \ldots, \lambda^8)$.

**Corollary 4.12** [1, Corollary 5.6]. Given positive integers $k, m$ and $n$, we have

$$(2m + n)^k = \sum_{r=0}^{\lfloor k/2 \rfloor} \sum_{\lambda \vdash k-2r} f^\lambda_{spo}f^\lambda_{ud},$$

where $f^\lambda_{ud}$ is the number of up-down $(m, n)$-tableaux of shape $\lambda$ and length $k$.

There is a bijection between two-line arrays $\pi \in A$ with $k$ columns and the set of $q \times (2m+n)$ matrices $M_\pi$ with integer entries greater than or equal to zero, where the rightmost $q \times n$ portion of $M_\pi$ contains entries from $\{0, 1\}$ and the sum of the entries in $M_\pi$ is equal to $k$. A two-line array $\pi \in A$ is associated with a matrix $M_\pi$, where the rows of $M_\pi$ are labelled by the entries in $\{1, \ldots, q\}$ and the columns by the entries in $B_0 \cup B_1$ in ascending order where the $ij$-th entry in $M_\pi$ is equal to the number of times the column $(i_j)$ appears in $\pi$. The process can clearly be reversed. Using this bijection to enumerate the two-line arrays $\pi \in A$ with $k$ columns, coupled with our spo-correspondence, yields the following.

**Corollary 4.13.** Given a positive integer $k$,

$$\sum_{r=0}^{k} \sum_{\mu} \binom{nq}{r} \left(\frac{2mq}{\ell(\mu)}\right)^{\ell(\mu)!} = \sum_{|\lambda|=|\mu|=k} \sum_{\text{cont}(\mu, i)} f^\lambda_{spo} d^\lambda_{even} d^\beta_{even},$$

where $d^\lambda$ is the number of semistandard tableaux of shape $\lambda$ with entries in $\{1, \ldots, q\}$, $d^\beta_{even}$ is the number of semistandard even tableaux of shape $\beta$, $\mu = (\mu_1, \mu_2, \ldots, \mu_t)$ and $\text{cont}(\mu, i)$ denotes the number parts $\mu_j$ such that $\mu_j = i$.

**Proof.** Given a two-line array with $k$ columns, if its associated matrix has $r$ entries equal to one in the rightmost $q \times n$ portion of the matrix then the entries in the $q \times 2m$ portion of the matrix must sum to $k-r$ so correspond
to a partition $\mu$ of $k - r$. For each such partition $\mu$ of $k - r$ there are

$$\frac{2mq}{(2mq - \ell(\mu))! \prod_{i=1}^{k-r} \text{cont}(\mu, i)} \prod_{i=1}^{k-r} \text{cont}(\mu, i)$$

such matrices.

\[
\frac{2mq}{(2mq - \ell(\mu))!} \prod_{i=1}^{k-r} \text{cont}(\mu, i) = \frac{(2mq)^{\ell(\mu)} \ell(\mu)!}{\prod_{i=1}^{k-r} \text{cont}(\mu, i)}
\]

\[\square\]

5. Establishing the Dual Orthosymplectic Cauchy Identity

In this section we give a bijective proof of the dual orthosymplectic Cauchy identity [1, Theorem 4.24].

**Theorem 5.1.** We have

$$\sum_{\lambda} \text{spo}_\lambda(Z)s_{\lambda'}(Y)$$

$$= \prod_{1 \leq i \leq m \atop 1 \leq j \leq q} (1 + x_i y_j)(1 + x_i^{-1} y_j) \prod_{1 \leq i \leq n \atop 1 \leq j \leq q} (1 - t_i y_j)^{-1} \prod_{1 \leq i \leq j \leq q} (1 - y_i y_j).$$

Using the dual Burge correspondence [13, Theorem 5.4], which states that

$$\prod_{1 \leq i \leq j \leq q} (1 - y_i y_j)^{-1} = \sum_{\beta \text{ even}} s_{\beta}(Y),$$

and rearranging the equation in Theorem 5.1, we have

$$\sum_{\lambda} \text{spo}_\lambda(Z)s_{\lambda'}(Y) \sum_{\beta \text{ even}} s_{\beta}(Y)$$

$$= \prod_{1 \leq i \leq m \atop 1 \leq j \leq q} (1 + x_i y_j)(1 + x_i^{-1} y_j) \prod_{1 \leq i \leq n \atop 1 \leq j \leq q} (1 - t_i y_j)^{-1}.$$

The left-hand side enumerates triples $(\tilde{P}_\lambda, P_{\lambda'}, P_{\beta})$, where $\tilde{P}_\lambda$ is an spo-tableau of shape $\lambda$, $P_{\lambda'}$ is a semistandard tableau of shape $\lambda'$ and $P_{\beta}$ is a semistandard tableau with even-length rows. The right-hand side counts two-line arrays $\pi \in \mathcal{A}^*$. To prove the identity, we will define a bijection between the two sets, which we will call the dual spo-correspondence. We begin by defining a bijection from the set $\mathcal{A}^*$ to the set of triples $(\tilde{P}_\lambda, P_{\lambda'}, L)$, where $L$ is a two-line dual Burge array. Given $\pi \in \mathcal{A}^*$, perform the following steps:

1. Let $\tilde{P}_0 = P_0 = L_0 = \emptyset$.
2. For $k \geq 1$ and starting from the leftmost column of $\pi$, consider the $k$th column $(i_j)$ in $\pi$. Insert $j$ into $\tilde{P}_{k-1}$ via spo-insertion to obtain $\tilde{P}_{k} = \tilde{P}_{k-1} \leftarrow j$. Then $\tilde{P}_k$ has either one more box or one less box than $\tilde{P}_{k-1}$.
   a. If $\tilde{P}_k$ has one more box than $\tilde{P}_{k-1}$, add a new box with entry $i$ to $P_{k-1}$ at the position where the extra box appears in $\tilde{P}_k$ to form $P_k$, which has the same shape as $\tilde{P}_k$ and let $L_k = L_{k-1}$. 

(1) Let $\tilde{P}_0 = P_0 = L_0 = \emptyset$.
(2) For $k \geq 1$ and starting from the leftmost column of $\pi$, consider the $k$th column $(i_j)$ in $\pi$. Insert $j$ into $\tilde{P}_{k-1}$ via spo-insertion to obtain $\tilde{P}_{k} = \tilde{P}_{k-1} \leftarrow j$. Then $\tilde{P}_k$ has either one more box or one less box than $\tilde{P}_{k-1}$.
   a. If $\tilde{P}_k$ has one more box than $\tilde{P}_{k-1}$, add a new box with entry $i$ to $P_{k-1}$ at the position where the extra box appears in $\tilde{P}_k$ to form $P_k$, which has the same shape as $\tilde{P}_k$ and let $L_k = L_{k-1}$. 

(1) Let $\tilde{P}_0 = P_0 = L_0 = \emptyset$.
(2) For $k \geq 1$ and starting from the leftmost column of $\pi$, consider the $k$th column $(i_j)$ in $\pi$. Insert $j$ into $\tilde{P}_{k-1}$ via spo-insertion to obtain $\tilde{P}_{k} = \tilde{P}_{k-1} \leftarrow j$. Then $\tilde{P}_k$ has either one more box or one less box than $\tilde{P}_{k-1}$.
   a. If $\tilde{P}_k$ has one more box than $\tilde{P}_{k-1}$, add a new box with entry $i$ to $P_{k-1}$ at the position where the extra box appears in $\tilde{P}_k$ to form $P_k$, which has the same shape as $\tilde{P}_k$ and let $L_k = L_{k-1}$.
(b) If $\tilde{P}_k$ has one less box than $P_{k-1}$, consider the location of this extra box in $P_{k-1}$ and perform reverse dual row insertion: bump the entry $x$ in that box into the row above it by replacing the largest entry $x'$ in that row that is less than or equal to $x$. Bump $x'$ into the row above it using the same procedure and continue until an entry $a$ is bumped out of the first row. Let $P_k$ be the resulting tableau, which has the same shape as $\tilde{P}_k$ and add the column $\binom{a}{d}$ on the right of $L_{k-1}$ to obtain $L_k$.

(3) Let $\tilde{P}_\lambda = \tilde{P}_s$ and $P_{\lambda^t} = (P_s)^t$. The array $L_s$ satisfies the property that whenever entries $i_k = i_{k+1}$ in its top row the corresponding bottom entries $b_k$ and $b_{k+1}$ satisfy $b_k \leq b_{k+1}$ (see proof of Theorem 5.6). Sort the bottom row of $L_n$ so that whenever there are two entries $i_k = i_{k+1}$ in the top row, the corresponding entries $b_k$ and $b_{k+1}$ in the bottom row satisfy $b_k \geq b_{k+1}$ and let $L$ be the resulting array.

**Example 5.2.**

Let $\pi = \left( \begin{array}{cccccccccccc}
1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 4 & 4 & 4 & 4 \\
5 & 3 & 1 & 3^o & 5 & 4 & 1^o & 1^o & 2 & 2 & 4 & 3 & 3 & 2 & 1
\end{array} \right)$

We omit the first several steps.

\[
\begin{array}{|c|c|c|c|}
\hline
\tilde{P}_9 &=& \begin{array}{ccc}
\overset{\circ}{2} & \overset{\circ}{4} & 1^o \\
3 & 5 & 1^o \\
\frac{3}{5} & 3^o \\
\frac{5}{5}
\end{array} & P_9 &=& \begin{array}{ccc}
1 & 2 & 3 \\
1 & 2 & 3 \\
1 & 2 \\
3
\end{array} & L_9 &=& \emptyset \\
\hline
\tilde{P}_{10} &=& \begin{array}{ccc}
\overset{\circ}{2} & \overset{\circ}{4} & 1^o \\
\overset{\circ}{5} & \overset{\circ}{5} & 1^o \\
\frac{5}{5} & 3^o \\
\frac{5}{5}
\end{array} & P_{10} &=& \begin{array}{ccc}
1 & 2 & 3 \\
1 & 2 & 3 \\
1 & 2 \\
1 & 3
\end{array} & L_{10} &=& \left( \begin{array}{c}
3 \\
2
\end{array} \right) \\
\hline
\tilde{P}_{11} &=& \begin{array}{ccc}
\overset{\circ}{1} & \overset{\circ}{4} & 1^o \\
5 & 1^o \\
\frac{5}{5} & 3^o \\
\frac{5}{5}
\end{array} & P_{11} &=& \begin{array}{ccc}
1 & 2 & 3 \\
1 & 2 \\
1 & 3
\end{array} & L_{11} &=& \left( \begin{array}{cc}
3 & 3 \\
2 & 3
\end{array} \right) \\
\hline
\tilde{P}_{12} &=& \begin{array}{ccc}
\overset{\circ}{1} & \overset{\circ}{4} & 1^o \\
\overset{\circ}{4} & 1^o \\
\overset{\circ}{5} & 3^o \\
\overset{\circ}{5}
\end{array} & P_{12} &=& \begin{array}{ccc}
1 & 2 & 3 \\
1 & 2 \\
1 & 3 \\
4
\end{array} & L_{12} &=& \left( \begin{array}{cc}
3 & 3 \\
2 & 3
\end{array} \right) \\
\hline
\tilde{P}_{13} &=& \begin{array}{ccc}
\overset{\circ}{1} & \overset{\circ}{3} & 1^o \\
\overset{\circ}{4} & 1^o \\
\overset{\circ}{4} & 3^o \\
\overset{\circ}{5}
\end{array} & P_{13} &=& \begin{array}{ccc}
1 & 2 & 3 \\
1 & 2 \\
1 & 3 \\
4 \\
4
\end{array} & L_{13} &=& \left( \begin{array}{cc}
3 & 3 \\
2 & 3
\end{array} \right)
\end{array}
\]
\[ \tilde{P}_{14} = \begin{array}{ccc} 1 & 3 & 1^\circ \\ 3 & 1^\circ & 4 \\ 4 & 3^\circ & \end{array}, \quad P_{14} = \begin{array}{ccc} 1 & 2 & 3 \\ 1 & 3 & 1 \\ 1 & 4 & 4 \end{array}, \quad L_{14} = \begin{pmatrix} 3 & 3 & 4 \\ 2 & 3 & 2 \end{pmatrix} \]

\[ \tilde{P}_{15} = \begin{array}{ccc} 1 & 2 & 1^\circ \\ 3 & 1^\circ & 3 \\ 3 & 3^\circ & \end{array}, \quad P_{15} = \begin{array}{ccc} 1 & 2 & 3 \\ 1 & 4 & 1 \\ 1 & 4 & 4 \end{array}, \quad L_{15} = \begin{pmatrix} 3 & 3 & 4 & 4 \\ 2 & 3 & 2 & 3 \end{pmatrix} \]

\[ \tilde{P}_{16} = \begin{array}{ccc} 2 & 1^\circ & \end{array}, \quad P_{16} = \begin{array}{ccc} 1 & 2 & \\ 1 & 4 & \\ 1 & 4 & \end{array}, \quad L_{16} = \begin{pmatrix} 3 & 3 & 4 & 4 & 4 \\ 2 & 3 & 2 & 3 & 3 \end{pmatrix} \]

The image of \( \pi \) is \( (\tilde{P}_{16}, P_{16}^{t}, L) \), where \( L = \begin{pmatrix} 3 & 3 & 4 & 4 & 4 \\ 3 & 2 & 3 & 3 & 2 \end{pmatrix} \).

To map a dual Burge array to a semistandard tableau \( Q_{\beta} \), where \( \beta^{t} \) has rows of even length, we use the dual Burge correspondence [3]. This is similar to the Burge correspondence described earlier, except that instead of performing regular row insertion in step 2(a), we perform dual row insertion. The new box from step 2(b) is added directly to the right of the newly created box and not directly below it like previously.

**Example 5.3.** Consider the dual Burge array \( L = \begin{pmatrix} 3 & 3 & 4 & 4 & 4 \\ 3 & 2 & 3 & 3 & 2 \end{pmatrix} \). The dual Burge correspondence yields the tableau \( S^{t} \), where \( S^{t} = \begin{array}{cccc} 2 & 2 & 3 & 3 \\ 3 & 3 & 3 & 4 \\ 3 & 4 & 4 & 4 \end{array} \).

The following is a description of the inverse dual algorithm that maps a triple \( (\tilde{P}_{\lambda}, P_{\lambda}^{t}, L) \) to a two-line array \( \pi \in A^{*} \), where \( \tilde{P}_{\lambda} \) is an spo-tableau of shape \( \lambda \), \( P_{\lambda}^{t} \) is a semistandard tableau of shape \( \lambda^{t} \) and \( L \) is a dual Burge two-line array.

1. Let \( \tilde{P}_{s} = \tilde{P}_{\lambda} \) and \( P_{s} = (P_{\lambda}^{t})^{t} \). Rewrite \( L_{n} \) so that when entries \( i_{k} \) and \( i_{k+1} \) in its top row satisfy \( i_{k} = i_{k+1} \), the corresponding bottom entries satisfy \( b_{k} \leq b_{k+1} \) and let this be \( L_{s} \). Let \( \pi_{s} = \emptyset \), where \( s \) is the total number of entries in \( P_{\lambda}^{t} \) and \( L \).
2. To obtain \( (\tilde{P}_{k-1}, P_{k-1}^{t}, L_{k-1}) \) and \( \pi_{k-1} \) from \( (\tilde{P}_{k}, P_{k}^{t}, L_{k}) \) and \( \pi_{k} \), find the largest entry \( i \) among all the entries in \( P_{k} \) and the top row of \( L_{k} \).
   a. If \( i \) belongs to \( L_{k} \), select the rightmost column \( \binom{i}{a} \) with a maximal and delete this column from \( L_{k} \) to obtain \( L_{k-1} \). Use dual row insertion to insert \( a \) into \( P_{k} \) to obtain \( P_{k-1} \), which has one more box than \( P_{k} \). The tableau \( P_{k-1} \) has one more box than \( \tilde{P}_{k} \); add an empty box to \( \tilde{P}_{k} \) in this spot and perform reverse jeu de taquin to slide the empty box into the first column and into the highest possible row \( r \), without creating a symplectic violation. Place an \( r \) in the empty box and then replace the right-most \( r \) in that row with \( \tau \). (Note
that there is at least one \( r \) in the row \( r \), since we placed an \( r \) in the empty box.) Then bump an entry \( r \) into the row above and perform reverse row bumping until some entry \( j \) is bumped out of the first row to give \( P_{k-1}. \) Prepend column \((\frac{1}{j})\) to \( \pi_k \) to obtain \( \pi_{k-1}. \)

(b) If \( i \) does not belong to \( L_k \) then select the lowest entry \( i \) in \( P_k \) and delete that box to get \( P_{k-1}. \) Since \( \hat{P}_k \) and \( P_k \) are of the same shape, \( \hat{P}_k \) and \( P_{k-1} \) differ by exactly one box. Locate the extra box in this location in \( \hat{P}_k \) and perform reverse spo-insertion, starting with the entry in this box, to obtain \( \hat{P}_{k-1}. \) This bumps out some entry \( j \) from the first row. Prepend column \((\frac{1}{j})\) to \( \pi_k \) to get \( \pi_{k-1} \) and let \( L_{k-1} = L_k. \)

(3) Set \( \pi = \pi_0. \)

**Lemma 5.4.** Let \( T \) be an spo-tableau and \( x, x' \in B_0 \cup B_1. \) Suppose \( x' \geq x \) if \( x, x' \in B_1 \) and \( x' > x \) otherwise. If \( T \leftarrow x' \) has one more box \( B' \) than \( T \) and \( (T \leftarrow x') \leftarrow x \) has one more box \( B \) than \( T \leftarrow x' \) then \( B \) lies strictly below \( B'. \)

**Proof.** Suppose that \( x' \) bumps \( y' \) from row one of \( T \) and that \( x \) subsequently bumps \( y \) from row one. Then \( y \) either belongs to the same box as \( y' \) or to a box to its left. The two bumped entries \( y \) and \( y' \) can only be equal if \( y, y' \in B_0 \) and otherwise \( y' > y \) so the argument can also be applied to all rows below the first. The bumping route for \( x' \) ends with the insertion of an entry \( z' \) in the box \( B' \) and the bumping route for \( x \) then either bumps \( z' \) into the next row or bumps an entry to the left of \( B' \) into the next row, so the added box \( B \) lies strictly below \( B'. \)

**Lemma 5.5.** Suppose that \( x, x' \in B_0 \) with \( x' > x \). If spo-insertion of \( x' \) into \( T \) causes a cancellation then insertion of \( x \) into \( T \leftarrow x' \) also causes a cancellation. Furthermore the jeu de taquin path of the empty box caused by inserting \( x' \) ends in a lower row than that of the empty box caused by inserting \( x. \)

**Proof.** Since insertion of \( x' \) causes a cancellation, it causes some \( i \geq x' \) in the first column of \( T \) to bump an \( \overline{7} \) in row \( i \) of \( T \) into row \( i+1 \). Thus each row \( r \) of the first column of \( T \leftarrow x' \) with \( r \leq i-1 \) contains either \( r \) or \( \overline{7}. \) The bumping path of \( x \) in the symplectic part of the tableau lies weakly left and strictly above the bumping path of \( x' \) (see [4]) so insertion of \( x \) causes an entry to bump out of one of the first \( i-1 \) rows of the first column of \( T \leftarrow x' \) into a subsequent row, which causes a cancellation in the first column.

Suppose that the empty box caused by inserting \( x' \) enters row \( i+1 \) at column \( m \) and let \( a_m \) denote the entry in row \( i \) of column \( m \) of \( T \) and \( b_m \) the entry directly above it. Then this box slides \( a_m \) into column \( m-1 \) of row \( i \) and, if the subsequent box formed by inserting \( x \) enters row \( i \), it must slide to at least column \( m \) or a column to the right of column \( m \) before entering row \( i+1 \) since \( a_m \geq b_m \) with possible equality if \( a_m, b_m \in B_1. \) The same argument applies to subsequent rows and the result now follows.

**Theorem 5.6.** The triple \((\hat{P}_\lambda, P^t_\lambda, L)\) consists of an spo-tableau \( \hat{P}_\lambda, \) a semistandard tableau \( P^t_\lambda \) and a dual Burge array \( L. \)
Proof. Since the entries in the top row of \( \pi \in A^* \) are weakly increasing and boxes added at any step to \( P_{k-1} \) to obtain \( P_k \) are in outer corners, the entries in the rows and columns of any \( P_k \) in the algorithm are weakly increasing. For entries \( a_i = a_{i+1} \) in the top row of \( \pi \in A^* \), we have \( b_i \geq b_{i+1} \) and Lemma 5.4 guarantees that a box added by inserting \( b_i \) into \( \tilde{P}_{i-1} \) is strictly above a box added by inserting \( b_{i+1} \) into \( \tilde{P}_i \), so the rows of any \( P_k \) obtained in the algorithm are strictly increasing, and thus \( P_k \) is semistandard.

If \( L_n = \left( i_1 \ i_2 \cdots \ i_r \ b' \ b \right) \), the justification that \( i_k \leq i_{k+1} \) and that \( i_k \geq j_k \) is the same as in the proof of Theorem 4.5. If \( i_k = i_{k+1} \), then the entries \( b' \) and \( b \) below \( i_k \) and \( i_{k+1} \) in \( \pi \) respectively satisfy \( b' > b \) and both cause cancellations. By Lemma 5.5 the taquin path of the empty box created by inserting \( b' \) ends in a lower row than the taquin path of the empty box created by subsequently inserting \( b \). It follows that the corresponding reverse row bumping path in \( P_k \), which bumps \( j_k \) out of the first row, begins in a lower row than the bumping path that bumps \( j_{k+1} \) out of the first row of \( P_{k+1} \) and, thus, \( j_k \leq j_{k+1} \). After rearranging, the array \( L \) is a dual Burge array. \( \Box \)

Acknowledgements

The authors wish to thank the anonymous referees who carefully read the paper and provided useful suggestions.

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

References

[1] G. Benkart, C. Shader, and A. Ram, Tensor product representations for orthosymplectic Lie superalgebras, J. Pure Appl. Algebra 130 (1998), 1–48.

[2] A. Berele, A Schensted-type correspondence for the symplectic group, J. Combin. Theory Ser. A 43 (1986), 320–328.

[3] W.H. Burge, Four correspondences between graphs and generalized Young tableaux, J. Combin. Theory Ser. A 17 (1974), 12–30.

[4] W. Fulton, Young tableaux, London Mathematical Society Student Texts, Cambridge University Press, Cambridge, 1997.

[5] D.E. Knuth, Permutations, matrices and generalized Young tableaux, Pacific J. Math. 34 (1970), 709–727.

[6] D. Littlewood and A.R. Richardson, Group characters and algebra, Philos. Trans. R. Soc. Lond. Ser. A 233 (1934), 99–141.

[7] I. G. Macdonald, Symmetric functions and Hall polynomials, second edition, Oxford University Press, 1995.
[8] B. E. Sagan, *The symmetric group: Representations, combinatorial algorithms, and symmetric functions*, Second edition, Graduate Texts in Mathematics, Springer-Verlag, 2001.

[9] C. Schensted, *Longest increasing and decreasing subsequences*, Canad. J. Math. **13** (1961), 179–191.

[10] R.P. Stanley, *Enumerative Combinatorics*, Vol. 2, Cambridge University Press, New York, 316–380, 1999.

[11] A. Stokke, *An orthosymplectic Pieri rule*, Electron. J. Comb. **25**(3) (2018).

[12] A. Stokke and T. Visentin, *Lattice path constructions for orthosymplectic determinantal formulas*, European J. Combin. **58** (2016), 38–51.

[13] S. Sundaram, *The Cauchy identity for Sp(2n)*, J. Combin. Theory Ser. A **53** (1990), 209–238.

Aalekh Patel and Harsh Patel  
Winnipeg  
Manitoba  
Canada

Anna Stokke  
Department of Mathematics and Statistics  
University of Winnipeg  
Winnipeg MB R3B 2E9  
Canada  
e-mail: a.stokke@uwinnipeg.ca

Communicated by Nathan Williams  
Received: 6 May 2021.  
Accepted: 28 December 2021.