COMBINATORICS OF PATIENCE SORTING MONOIDS

ALAN J. CAIN, ANTÓNIO MALHEIRO, AND FÁBIO M. SILVA

Abstract. This paper makes a combinatorial study of the two monoids and the two types of tableaux that arise from the two possible generalizations of the Patience Sorting algorithm from permutations (or standard words) to words. For both types of tableaux, we present Robinson–Schensted–Knuth-type correspondences (that is, bijective correspondences between word arrays and certain pairs of semistandard tableaux of the same shape), generalizing two known correspondences: a bijective correspondence between standard words and certain pairs of standard tableaux, and an injective correspondence between words and pairs of tableaux.

We also exhibit formulas to count both the number of each type of tableaux with given evaluations (that is, containing a given number of each symbol). Observing that for any natural number n, the n-th Bell number is given by the number of standard tableaux containing n symbols, we restrict the previous formulas to standard words and extract a formula for the Bell numbers. Finally, we present a 'hook length formula' that gives the number of standard tableaux of a given shape and deduce some consequences.

Contents

Key words and phrases. Patience Sorting algorithm; Robinson–Schensted–Knuth correspondence; Bell number; hook-length formula.

The first author was supported by an Investigador FCT fellowship (IF/01622/2013/CP1161/CT0001).

For the first two authors, this work was partially supported by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the project UID/MAT/00297/2013 (Centro de Matemática e Aplicações), and the project PTDC/MHC-FIL/2583/2014.

The third author was supported by an FCT Lismath fellowship (PD/BD/52644/2014) and partially supported by the FCT project CEMAT-Ciências UID/Multi/04621/2013.

1. Introduction

Patience Sorting is a one-person card game that was created as a method to sort decks of cards. The name has its origins in the works of Mallows, who credited the game to Ross [Mal62, Mal63].

The idea of the game is to split a given shuffled deck of cards on the symbols $1, 2, \ldots, n$ into sorted subsequences called columns (piles according to [AD99, BL07]) using Mallows' Patience Sorting procedure, with the goal of finishing with as few piles as possible [AD99]. The decks of cards can therefore be seen as standard words from the symmetric group of order n (denoted $\mathfrak{S}_n$). According to this notation, following Algorithm 1.1 of [BL07], this procedure can be described in the following way

Algorithm 1.1 (Mallows' Patience Sorting procedure)**.**
Input: A standard word $\sigma = \sigma_1\sigma_2 \cdots \sigma_n \in \mathfrak{S}_n$.
Output: A set of columns $\{c_1, c_2, \ldots, c_m\}$.
Method:

- Put σ_1 on the bottom of the first column, c_1;
- for each remaining symbol σ_i, with $i \in \{2, \ldots, n\}$, if $b_1, ..., b_k$ are the symbols on the bottom of the columns $c_1, ..., c_k$ that have already been formed:
  - if $\sigma_i > b_k$, then put the symbol σ_i into the bottom of the column c_{k+1};
  - otherwise, find the leftmost symbol b_j from $b_1, \ldots, b_k$ that is greater than σ_i and put σ_i on the bottom of that column.

As noted in [BL07, AD99], there are certain similarities between this algorithm and Schensted's insertion algorithm for standard words. In fact, as noted by the authors of [BL07], this algorithm can be seen as a non-recursive version of Schensted's insertion algorithm. This perspective suggests that there would be an analogue of the Robinson correspondence (cf. [Ful97]): a bijection between standard words and pairs of standard Young tableaux of the same shape, discovered by Robinson in [Rob38] and later rediscovered by Schensted for the context of words [Sch61]. Burstein and Lankham [BL07] constructed such an analogue by considering an algorithm that simultaneously performs Mallows' Patience Sorting procedure and the recording of that procedure.

Following the terminological distinction made in [Ful97], the original Robinson correspondence (between standard words and pairs of standard Young tableaux of the same shape) has two extensions:

- the Robinson–Schensted correspondence, which is a bijection between words and pairs (P, Q), where P is a semistandard Young tableau and Q is a standard Young tableaux of the same shape; and
- the Robinson–Schensted–Knuth correspondence [Knu70], a bijection between two-rowed lexicographic arrays and pairs of semistandard Young tableaux having the same shape.

Moreover, Schensted's insertion algorithm gives rise to a combinatorial monoid, the plactic monoid [LS81], which can be described as the quotient of the free monoid over the congruence which relates words in that free monoid that yield the same semistandard Young tableau under Schensted's insertion algorithm (see [Lot02, § 6] for more details).

Regarding the combinatorial monoid part, considering a possible generalization of Algorithm 1.1 to words and using a similar method, Rey constructed the Bell monoid [Rey07]. In fact, in that same paper, Rey proposes a Robinson–Schensted-like correspondence between words and pairs of PS tableaux (Bell tableaux, in the notation of [Rey07]) (P, Q) having the same shape, P being semistandard and Q standard. However, this correspondence proves to be only an injection.

In [AD99] two possible generalizations of Patience Sorting for words were proposed, one of them coinciding with the one studied by Rey. So, following these generalizations, in [CMS17] the present authors constructed and studied two distinct monoids, the lps monoid (which coincides with the Bell monoid of [Rey07]) arising from lPS tableaux and the rps monoid arising from rPS tableaux.

Our goal in Section 3 is to provide bijective Robinson–Schensted-like and Robinson–Schensted–Knuth-like correspondences for both the lps and the rps monoids, extending the ideas of [BL07].

In sections 4 and 5 we turn to another topic, namely counting lPS and rPS tableaux. As noted in [BL07], given a totally ordered alphabet $\mathcal{B}$ and a partition of $\mathcal{B}$, there is a natural identification between the sets that compose that partition of $\mathcal{B}$ and the columns of a standard PS tableau obtained from the insertion of certain standard words over $\mathcal{B}$ under Algorithm 1.1. This identification gives rise to a one-to-one correspondence between the partitions of $\mathcal{B}$ and the standard PS tableaux obtained from applying Algorithm 1.1 to the standard words over $\mathcal{B}$. Since the n-th Bell number counts the number of distinct ways to partition a set with n distinct symbols (cf. sequence A000110 in the OEIS), the n-th Bell number also provides the number of standard PS tableaux over a totally ordered alphabet with n distinct symbols. Furthermore, as the Stirling number of second kind, $S(n, k)$, is equal to the number of distinct ways to partition n distinct elements into k distinct sets [CG96], the sum of $S(n, k)$ with k ranging between 1 and n is also equal to n-th Bell number. In Section 4 we will follow a similar strategy. More specifically, we will count the number of lPS (resp.

rPS) tableaux over words with the same number of each symbols (that is, with the same evaluation) and having the same lPS (rPS) bottom row. Then, by summing over all the possible lPS (rPS) bottom rows, we extract a formula to count the number of lPS (rPS) tableaux over words with the same evaluation. Restricting the lPS and rPS formulas to standard PS tableaux we will see that they coincide and we will deduce a formula to count the Bell numbers.

Section 5 also gives a formula connecting the PS tableaux and Bell numbers. However, we will follow an entirely different approach. Our strategy will be to provide an analogue of the 'hook length formula', which gives the number of standard Young tableaux over $\{1, \ldots, n\}$ with a given shape. This rule is due to J.S. Frame, G. de B. Robinson, and R.M. Thrall [FRT$^+$54] and has been generalized to other combinatorial objects such as shifted Young tableaux (see [Thr52], [Sag80]) and trees (see [SY89]). In this paper we provide an analogue of this result for standard PS tableaux over an arbitrary totally ordered alphabet $\mathcal{B}$. By summing the hook length formulas over all the possible shapes for standard PS tableaux over $\mathcal{B}$, we also obtain the Bell number of order $|\mathcal{B}|$. Furthermore, this rule together with the injectivity of the Robinson–Schensted-like correspondence from [Rey07] allows us to deduce an upper bound on the number of standard words that insert to a specific standard PS tableau.

2. Preliminaries and notation

In this section we introduce the notions that we shall use along the paper. For more details concerning these constructions see for instance [Lot02], [HNT05], [How95] and [CMS17].

2.1. Words and standardization.

The alphabets that we will consider on this paper will always be totally ordered.

Given an alphabet $\mathcal{B}$, $\mathcal{B}^*$ denotes the free monoid over $\mathcal{B}$. In this paper, $\mathcal{A} = \{1 < 2 < 3 < \ldots\}$ denotes the set of natural numbers viewed as an infinite totally ordered alphabet. Furthermore, for any $n \in \mathbb{N}$, the set $\mathcal{A}_n$ denotes the totally ordered subset of $\mathcal{A}$, on the symbols $1, \ldots, n$.

A *word* over an arbitrary alphabet is an element of the free monoid over that alphabet, with the symbol ε denoting the empty word. In particular, a *standard word* over a finite (resp. infinite) alphabet is a word where each symbol from the alphabet occurs exactly (at most) once. For any finite alphabet $\mathcal{B}$, denote by $\mathfrak{S}(\mathcal{B})$ the set of standard words over $\mathcal{B}$. For example, if $\mathcal{B} = \{2, 4, 5\}$, then

$$\mathfrak{S}(\mathcal{B}) = \{245, 254, 425, 452, 524, 542\}.$$

In the following paragraphs we will define several concepts that are directly related with the notion of word. So, if $\mathcal{B}$ is an arbitrary alphabet and $w = w_1 \cdots w_m \in \mathcal{B}^*$, with $w_1, \ldots, w_m \in \mathcal{B}$, then:

- the *length of* w, $|w|$, is the number of symbols from $\mathcal{B}$ that compose w, counting repetitions. If $w = \varepsilon$, then $|w| = 0$;
- a word $u \in \mathcal{B}^*$ is said to be a *subword of* w if there exists a sequence of indexes, $i_1, \ldots, i_k \in \mathbb{N}$, with $1 \leq i_1 < \ldots < i_k \leq m$, such that $u = w_{i_1} \cdots w_{i_k}$;
- for any word $u \in \mathcal{B}^*$, u is a *factor* of w if there exist words $v_1, v_2 \in \mathcal{B}^*$, such that $w = v_1 u v_2$;
- for any $a \in \mathcal{B}$, the number of occurrences of a in w, is denoted by $|w|_a$;
- the *content of* w, is the set $\operatorname{cont}(w) = \left\{a \in \mathcal{B} : |w|_a \geq 1\right\}$;
- the *evaluation of* a word $w \in \mathcal{B}^*$, denoted $\operatorname{ev}(w)$, is the $|\mathcal{B}|$-tuple of non-negative integers, indexed in increasing order by the elements of $\mathcal{B}$, whose a-th term is $|w|_a$.

Next, we define two different processes for standardizing words over $\mathcal{B}$, both allowing us to assign to any word from $\mathcal{B}^*$, a standard word of the same length over a new alphabet.

Consider the alphabet

$$\mathcal{C}(\mathcal{B}) = \{a_i : a \in \mathcal{B} \ \wedge \ i \in \mathcal{A}\}$$

totally ordered in the following way: for any $a_i, c_j \in \mathcal{C}(\mathcal{B})$

$$a_i <_{\mathcal{C}(\mathcal{B})} c_j \Leftrightarrow a <_{\mathcal{B}} c \vee (a = c \wedge i <_{\mathcal{A}} j)\,.$$

Given a word w over $\mathcal{B}$, $\operatorname{std}_{\mathrm{l}}(w)$ denotes the *left to right standardization of* w, which is the word obtained by reading w from left to right and attaching to each symbol $a \in \operatorname{cont}(w)$ an index i to the i-th occurrence of a. The *right to left standardization of* w, $\operatorname{std}_{\mathrm{r}}(w)$, is obtained in the same way, but reading the word from right to left. For example, considering the word $w = 4124321 \in \mathcal{A}_4^*$,

$$\begin{array}{lcl} w & = & 4\ 1\ 2\ 4\ 3\ 2\ 1 \\ \operatorname{std}_{\mathrm{l}}(w) & = & 4_1\ 1_1\ 2_1\ 4_2\ 3_1\ 2_2\ 1_2 \\ \operatorname{std}_{\mathrm{r}}(w) & = & 4_2\ 1_2\ 2_2\ 4_1\ 3_1\ 2_1\ 1_1 \end{array}$$

For any word w over $\mathcal{B}$, it is straightforward that both $\operatorname{std}_{\mathrm{l}}(w)$ and $\operatorname{std}_{\mathrm{r}}(w)$ are standard words over the alphabets $\operatorname{cont}(\operatorname{std}_{\mathrm{l}}(w)) \subseteq \mathcal{C}(\mathcal{B})$ and $\operatorname{cont}(\operatorname{std}_{\mathrm{r}}(w)) \subseteq \mathcal{C}(\mathcal{B})$, respectively, with the order induced from $\mathcal{C}(\mathcal{B})$.

Henceforth, any word arising from the processes of left to right or right to left standardization will be called a *standardized word*. Given any symbol $a_b \in \mathcal{C}(\mathcal{B})$, the *underlying symbol of* a_b is a and the *index of* a_b is b.

Given any standardized word $w \in \mathcal{C}(\mathcal{B})$, the *de-standardization of* w, $\operatorname{std}^{-1}(w) \in \mathcal{B}$, is the word obtained from w by erasing all of its indexes.

By the uniqueness of the writing of words, together with the fact that any symbol in any position of a given word coincides with the underlying symbol in the same position of the word obtained through any of the previous processes of standardization, we have the following

Remark 2.1. *For each word w, using processes of standardization previously described we obtain unique standardized words* $\operatorname{std}_{\mathrm{l}}(w)$ *and* $\operatorname{std}_{\mathrm{r}}(w)$. *Moreover, by applying de-standardization to any of these standardized words, we obtain the word w.*

2.2. **PS tableaux and insertion.** In this subsection we introduce the concept of pre-tableaux and recall the basic concepts regarding patience sorting (PS) tableaux over an arbitrary totally ordered alphabet $\mathcal{B}$. We will also recall the insertion on PS tableaux.

A *composition diagram* is a concatenation of a finite collection of boxes arranged in bottom-justified columns. Such a diagram is said to be a *column diagram* if all its boxes are disposed vertically.

A *pre-column* over $\mathcal{B}$ is a column diagram where each of its boxes is filled with a symbol from $\mathcal{B}$ and such that distinct boxes have distinct symbols. A *pre-tableau over* $\mathcal{B}$ is a composition diagram where all boxes are filled with symbols from $\mathcal{B}$ in such a way that distinct boxes have distinct symbols. For instance, if

$$C = \begin{array}{|c|c|c|c|} \cline{2-2} \multicolumn{1}{c|}{} & & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \\ \cline{2-4} \multicolumn{1}{c|}{} & & & \\ \hline & & & \\ \hline \end{array}, \quad \text{and} \quad P = \begin{array}{|c|c|c|} \cline{2-2} \multicolumn{1}{c|}{} & 5 & \multicolumn{1}{c}{} \\ \hline 2 & 1 & 4 \\ \hline 7 & 3 & 6 \\ \hline \end{array},$$

then C is a composition diagram and P is a pre-tableau.

An *lPS (rPS) tableau* over $\mathcal{B}$ is a composition diagram where each box is filled with a symbol from $\mathcal{B}$ and such that when reading the symbols of its columns from top to bottom they are in strictly (weakly) decreasing order and when reading the symbols in the bottom-row from left to right, they are in weakly (strictly) increasing order.

A *standard PS tableau over* $\mathcal{B}$ is an lPS tableau (or alternatively an rPS tableau) such that each symbol of $\mathcal{B}$ occurs at most once, whereas a *recording tableau* is a standard PS tableau such that if m is the number of boxes in its underlying composition diagram, then each symbol from $\mathcal{A}_m$ occurs exactly once. For instance, if

$$Q = \begin{array}{|c|c|c|} \cline{3-3} \multicolumn{1}{c}{} & \multicolumn{1}{c|}{} & 4 \\ \hline 4 & 2 & 3 \\ \hline 1 & 1 & 2 \\ \hline \end{array}, \quad R = \begin{array}{|c|c|c|} \cline{1-1} 4 & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \\ \hline 1 & 2 & 4 \\ \hline 1 & 2 & 3 \\ \hline \end{array}, \quad \text{and} \quad S = \begin{array}{|c|c|c|} \cline{3-3} \multicolumn{1}{c}{} & \multicolumn{1}{c|}{} & 7 \\ \hline 6 & 3 & 5 \\ \hline 1 & 2 & 4 \\ \hline \end{array},$$

then Q is an lPS tableau, R is an rPS tableau and S is both a standard PS and a recording tableau. The tableaux Q and R can be considered as tableaux over the alphabets $\mathcal{A}$ or $\mathcal{A}_n$ (for $n \geq 4$), whereas the tableaux P and S can be viewed as tableaux over $\mathcal{A}$ or $\mathcal{A}_n$ (for $n \geq 7$).

For any natural $n \in \mathbb{N}$, $\lambda = (\lambda_1, \ldots, \lambda_m) \in \mathbb{N}^m$ is a *composition of* n, denoted by $\lambda \vDash n$, if $\lambda_1 + \cdots + \lambda_m = n$. Given an arbitrary pre-tableau

T, such that $c_1, c_2, \ldots, c_m$ are its columns from left to right, we will often refer to this tableau using the notation $c_1 c_2 \cdots c_m$ instead of T. We make the following definitions:

- the *content of* T, $\operatorname{cont}(T)$, is the set composed by the symbols that occur in T;
- for any $i \in \{1, \ldots, m\}$, the *length of the column* c_i is the number of boxes that compose c_i and is denoted by $|c_i|$, the *length of the bottom row of* T is equal to m, and the *length of* T, $|T|$, is given by the sum of the lengths of its columns $|T| = |c_1| + \cdots + |c_m|$;
- the *shape of* T, denoted by $\operatorname{sh}(T)$ is the composition formed by the lengths of the columns of T from left to right, that is, $\operatorname{sh}(T) = \big(|c_1|, \ldots, |c_m|\big)$. Note that this notion of shape is the dual of the usual notion of shape, as given in [Lot02];
- if a belongs to the content of T and a occurs in the i-th column of T, counting columns from left to right and in the j-th box of the i-th column counting boxes from bottom to top, then the *column-row position of* a is given by the pair $(i, j) \in \mathbb{N} \times \mathbb{N}$;
- if T is a tableaux over $\mathcal{B}$, then for any $a \in \mathcal{B}$, $|T|_a$ denotes the number of symbols a occurring in T and the *evaluation of* T is the sequence (infinite or finite of length $|\mathcal{B}|$ if $\mathcal{B}$ is, respectively, infinite or finite) whose a-th term is $|T|_a$, for any $a \in \mathcal{B}$.

Considering the standard rPS tableau R given in the previous example, the content of R is given by the set $\operatorname{cont}(R) = \{1, 2, 3, 4\}$, the length of R is 7, the shape of R is given by $\operatorname{sh}(R) = (3, 2, 2)$, the symbol 3 occurs in the column-row position $(3, 1)$ of R, seen as a tableau over $\mathcal{A}$, the evaluation of R is given by the infinite sequence $(2, 2, 1, 2, 0, 0, \ldots)$, seen as a tableau over $\mathcal{A}_4$, the evaluation of R is given by the sequence $(2, 2, 1, 2)$.

Any two pre-tableaux are said to be equal if they have the same shape and if in corresponding column-row positions of the tableaux the symbols are equal. If $\mathcal{B} \subseteq \mathcal{A}_n$ for some n, and $\lambda \vDash |\mathcal{B}|$, then $\operatorname{PSTab}_{\mathcal{B}}(\lambda)$ denotes the set of standard PS tableaux with shape λ and whose content is $\mathcal{B}$. Similarly, $\operatorname{Tab}_{\mathcal{B}}(\lambda)$ denotes the set of pre-tableaux with shape λ and whose content is $\mathcal{B}$. Furthermore,

$$\operatorname{PSTab}_{\mathcal{B}} = \bigcup_{\lambda \vDash |\mathcal{B}|} \operatorname{PSTab}_{\mathcal{B}}(\lambda) \quad \text{and} \quad \operatorname{Tab}_{\mathcal{B}} = \bigcup_{\lambda \vDash |\mathcal{B}|} \operatorname{Tab}_{\mathcal{B}}(\lambda).$$

For example,

$$\operatorname{PSTab}_{\{2,4,5\}}\big((2,1)\big) = \left\{ \begin{array}{|c|c|} \cline{1-1} 4 \\ \hline 2 & 5 \\ \hline \end{array}, \begin{array}{|c|c|} \cline{1-1} 5 \\ \hline 2 & 4 \\ \hline \end{array} \right\}$$

$$\operatorname{PSTab}_{\{2,4,5\}} = \left\{ \begin{array}{|c|} \hline 5 \\ \hline 4 \\ \hline 2 \\ \hline \end{array}, \begin{array}{|c|c|} \cline{1-1} 4 \\ \hline 2 & 5 \\ \hline \end{array}, \begin{array}{|c|c|} \cline{1-1} 5 \\ \hline 2 & 4 \\ \hline \end{array}, \begin{array}{|c|c|} \cline{2-2} \multicolumn{1}{c|}{} & 5 \\ \hline 2 & 4 \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline 2 & 4 & 5 \\ \hline \end{array} \right\},$$

while

$$
\begin{aligned}
&\mathrm{Tab}_{\{2,4,5\}}((2,1))\\
&\qquad = \left\{ \begin{ytableau} 2 \\ 4 & 5 \end{ytableau}, \begin{ytableau} 2 \\ 5 & 4 \end{ytableau}, \begin{ytableau} 4 \\ 2 & 5 \end{ytableau}, \begin{ytableau} 4 \\ 5 & 2 \end{ytableau}, \begin{ytableau} 5 \\ 2 & 4 \end{ytableau}, \begin{ytableau} 5 \\ 4 & 2 \end{ytableau} \right\}
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Tab}_{\{2,4,5\}} = \Bigg\{ & \begin{ytableau} 2 \\ 4 \\ 5 \end{ytableau}, \begin{ytableau} 2 \\ 5 \\ 4 \end{ytableau}, \begin{ytableau} 4 \\ 2 \\ 5 \end{ytableau}, \begin{ytableau} 4 \\ 5 \\ 2 \end{ytableau}, \begin{ytableau} 5 \\ 2 \\ 4 \end{ytableau}, \begin{ytableau} 5 \\ 4 \\ 2 \end{ytableau}, \begin{ytableau} 2 \\ 4 & 5 \end{ytableau}, \begin{ytableau} 2 \\ 5 & 4 \end{ytableau}, \begin{ytableau} 4 \\ 2 & 5 \end{ytableau}, \begin{ytableau} 4 \\ 5 & 2 \end{ytableau},\\
& \begin{ytableau} 5 \\ 2 & 4 \end{ytableau}, \begin{ytableau} 5 \\ 4 & 2 \end{ytableau}, \begin{ytableau} \none & 2 \\ 4 & 5 \end{ytableau}, \begin{ytableau} \none & 2 \\ 5 & 4 \end{ytableau}, \begin{ytableau} \none & 4 \\ 2 & 5 \end{ytableau}, \begin{ytableau} \none & 4 \\ 5 & 2 \end{ytableau}, \begin{ytableau} \none & 5 \\ 2 & 4 \end{ytableau}, \begin{ytableau} \none & 5 \\ 4 & 2 \end{ytableau},\\
& \begin{ytableau} 2 & 4 & 5 \end{ytableau}, \begin{ytableau} 2 & 5 & 4 \end{ytableau}, \begin{ytableau} 4 & 2 & 5 \end{ytableau}, \begin{ytableau} 4 & 5 & 2 \end{ytableau}, \begin{ytableau} 5 & 2 & 4 \end{ytableau}, \begin{ytableau} 5 & 4 & 2 \end{ytableau} \Bigg\}.
\end{aligned}
$$

The following algorithm describes the insertion of an arbitrary word into a PS tableau, merging in one algorithm the algorithms 3.1 and 3.2 of [CMS17]. We will use the notation $\mathrm{P}_{\mathrm{lps}}()$, $\mathrm{P}_{\mathrm{rps}}()$ instead of the notation $\mathfrak{R}_\ell()$, $\mathfrak{R}_r()$, respectively, of [CMS17].

Algorithm 2.2 (PS insertion of a word)**.**

Input: A word w over a totally ordered alphabet.

Output: An lPS tableau $\mathrm{P}_{\mathrm{lps}}(w)$ (resp., rPS tableau $\mathrm{P}_{\mathrm{rps}}(w)$).

Method:

(1) If $w = \varepsilon$, output the empty tableaux $\emptyset$. Otherwise:

(2) $w = w_1 \cdots w_n$, with symbols $w_1, \ldots, w_n$ of the alphabet. Setting

$$\mathrm{P}_{\mathrm{lps}}(w_1) = \boxed{w_1} = \mathrm{P}_{\mathrm{rps}}(w_1),$$

then, for each remaining symbol w_j with $1 < j \leq n$, denoting by $r_1 \leq \cdots \leq r_k$ (resp., $r_1 < \cdots < r_k$) the symbols in the bottom row of the tableau $\mathrm{P}_{\mathrm{lps}}(w_1 \cdots w_{j-1})$ (resp., $\mathrm{P}_{\mathrm{rps}}(w_1 \cdots w_{j-1})$):

- if $r_k \leq w_j$ (resp., $r_k < w_j$), insert w_j in a new column to the right of r_k in $\mathrm{P}_{\mathrm{lps}}(w_1 \cdots w_{j-1})$ (resp., $\mathrm{P}_{\mathrm{rps}}(w_1 \cdots w_{j-1})$);
- otherwise, if $m = \min\big\{i \in \{1, \ldots, k\} : w_j < r_i\big\}$, (resp. $m = \min\big\{i \in \{1, \ldots, k\} : w_j \leq r_i\big\}$) construct a new empty box on top of the m-th column of $\mathrm{P}_{\mathrm{lps}}(w_1 \cdots w_{j-1})$ (resp. $\mathrm{P}_{\mathrm{rps}}(w_1 \cdots w_{j-1})$). Then bump all the symbols of the column containing r_m to the box above and insert w_j in the box which has been cleared and previously contained the symbol r_m.

Output the resulting tableau.

For brevity, we will use $\mathrm{P}()$, whenever no distinction between $\mathrm{P}_{\mathrm{lps}}()$ and $\mathrm{P}_{\mathrm{rps}}()$ is needed. Given $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$ and words u, v of the same length (possibly over different totally ordered alphabets), we say that u and v have *equivalent* x*PS insertions* if $\mathrm{sh}(\mathrm{P}_{\mathrm{xps}}(u)) = \mathrm{sh}(\mathrm{P}_{\mathrm{xps}}(v))$ and for every $i = 1, \ldots, |u|$ the i-th symbol of u and the i-th symbol of v are in the same column-row position of the respective tableaux $\mathrm{P}_{\mathrm{xps}}(u)$ and $\mathrm{P}_{\mathrm{xps}}(v)$.

The following lemma is a restatement of [CMS17, Lemma 3.4] which relates the insertion of an arbitrary word with its x-standardization.

Lemma 2.3. *For any word w over an alphabet $\mathcal{B}$ and $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$, the words w and $\mathrm{std}_{\mathrm{x}}(w)$ have equivalent* x*PS insertions.*

There are also corresponding *standardization and de-standardization* processes for PS tableaux. Considering a PS tableau R, $\mathrm{Std}_{\mathrm{l}}(R)$ denotes the lPS tableau obtained from R, by reading its entries column by column, from left to right, and on each column from top to bottom, and attaching to each symbol $a \in \mathrm{cont}(R)$ an index $i \in \mathcal{A}$ to the i-th appearance of a. Moreover, $\mathrm{Std}_{\mathrm{r}}(R)$ denotes the rPS tableau obtained from R, by reading its entries column by column, from right to left, and on each column from bottom to top, attaching to each symbol $a \in \mathrm{cont}(R)$ an index $i \in \mathcal{A}$ to the i-th appearance of a.

On the other hand, if the considered PS tableau R has symbols from $\mathcal{C}(\mathcal{B})$, for some alphabet $\mathcal{B}$, then the *de-standardization of R*, denoted by $\mathrm{Dstd}(R)$, is the tableau produced by erasing the indexes of each of its underlying symbols.

A direct consequence of the lemma is the following:

Remark 2.4. *If $R = \mathrm{P}_{\mathrm{xps}}(w)$, for some word w, then $\mathrm{P}_{\mathrm{xps}}(\mathrm{std}_{\mathrm{x}}(w)) = \mathrm{Std}_{\mathrm{x}}(R)$.*

We prove the following result.

Lemma 2.5. *For $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$, let R be a* x*PS tableau over an alphabet $\mathcal{B}$ and let $\mathrm{Std}_{\mathrm{x}}(R)$ be the corresponding standardized* x*PS tableau over the alphabet $\mathcal{C}(\mathcal{B})$. Any word that inserts to $\mathrm{Std}_{\mathrm{x}}(R)$ using Algorithm 2.2 has the form $\mathrm{std}_{\mathrm{x}}(w)$, for some word w over $\mathcal{B}$.*

Proof. Since the rPS case follows with a similar argument, we only prove the lPS case. It is clear that if z is a word over $\mathcal{C}(\mathcal{B})$ which inserts to $\mathrm{Std}_{\mathrm{l}}(R)$ using Algorithm 2.2, then there exits a word w over $\mathcal{B}$, where w is obtained from z removing the indexes of all symbols and $\mathrm{cont}(z) = \mathrm{cont}(\mathrm{std}_{\mathrm{l}}(w))$.

Suppose that there are symbols a_i, a_j with $i > j$ such that a_i occurs to the left of a_j in z. Then, using Algorithm 2.2, the symbol a_j is going to appear in $\mathrm{Std}_{\mathrm{l}}(R)$ either in the same column of a_i below it, or in a column to the right. This contradicts the process of lPS standardization of a tableaux previously described, which follows on the columns of R from left to right and top to bottom. It follows that a_j occurs in a column to the left where a_i occurs. Therefore, the only words that can insert to $\mathrm{Std}_{\mathrm{x}}(R)$ have the required form. $\square$

The following result is a restatement of [CMS17, Proposition 3.5].

Proposition 2.6. *For any word w over an alphabet $\mathcal{B}$ and* $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$, *we have*

(1) $\mathrm{P}_{\mathrm{xps}}(\mathrm{std}_{\mathrm{x}}(w)) = \mathrm{Std}_{\mathrm{xps}}(\mathrm{P}_{\mathrm{xps}}(w))$*; and*
(2) $\mathrm{Dstd}\big(\mathrm{P}_{\mathrm{xps}}\big(\mathrm{std}_{\mathrm{x}}(w)\big)\big) = \mathrm{P}_{\mathrm{xps}}(w)$.

3. A Robinson–Schensted–Knuth-like correspondence

In its original formulation, the Robinson correspondence is a bijection that maps standard words from $\mathfrak{S}(\mathcal{A}_n)$ to pairs of standard Young tableaux of the same shape with content $\mathcal{A}_n$. For each standard word σ of $\mathfrak{S}(\mathcal{A}_n)$, the first component of the pair of tableaux is obtained by applying Schensted's insertion algorithm to σ, whereas the second is the tableaux that records the position of each symbol inserted into the first tableaux. The Robinson–Schensted correspondence is the generalization of this correspondence to words. Both can be seen as particular cases of the Robinson–Schensted–Knuth correspondence, which is the bijection obtained by considering two-rowed arrays in lexicographic order as starting point and taking as images pairs of semistandard Young tableaux, such that the first tableau is obtained from the Schensted insertion applied to the second component of the two-rowed array and the second tableau is obtained from recording the steps of this insertion according to the first word of the two-rowed array (see [Ful97] for more details concerning these constructions).

In this section we introduce Robinson–Schensted–Knuth-like correspondences for both lPS and rPS tableaux. These correspondences will map two-rowed arrays over a totally ordered alphabet to pairs of PS tableaux having the same shape and avoiding certain pairs of patterns. Similarly to what happens with the original correspondence, the first element of the pair will be a PS tableau coming from the PS insertion of the second word of the two-rowed array, and the second element, the tableau which records this insertion according to the first word of the two-rowed array. To prove this result we will make use of [BL07, Theorem 3.9] in which the authors establish a Robinson-like correspondence between standard words from $\mathfrak{S}(\mathcal{A}_n)$ and pairs of standard Patience Sorting tableaux over $\mathcal{A}_n$ having the same shape and avoiding certain pairs of patterns.

3.1. A Robinson–Schensted-like correspondence.

We start by presenting a Robinson–Schensted-like correspondence. In order to do so, first we provide an algorithm which is the adaptation of [BL07, Algorithm 3.1] to words over a totally ordered alphabet.

Algorithm 3.1 (Extended PS insertion of a word)**.**

Input: A word w over an alphabet.

Output: A pair of tableaux $(\mathrm{P}_{\mathrm{lps}}(w), \mathrm{Q}_{\mathrm{lps}}(w))$ (resp. $(\mathrm{P}_{\mathrm{rps}}(w), \mathrm{Q}_{\mathrm{rps}}(w))$) of the same shape.

Method:

(1) If $w = \varepsilon$, output a pair of empty tableaux $(\emptyset, \emptyset)$. Otherwise:
(2) $w = w_1 \cdots w_n$, with symbols $w_1, \ldots, w_n$ of the alphabet. Let

$$\big(\mathrm{P}_{\mathrm{lps}}(w_1), \mathrm{Q}_{\mathrm{lps}}(w_1)\big) = \big(\boxed{w_1}, \boxed{1}\big) = \big(\mathrm{P}_{\mathrm{rps}}(w_1), \mathrm{Q}_{\mathrm{rps}}(w_1)\big),$$

and for each remaining symbol w_j with $1 < j \leq n$, denote by $r_1 \leq \cdots \leq r_k$ (resp. $r_1 < \cdots < r_k$) the symbols in the bottom row of the tableau $\mathrm{P}_{\mathrm{lps}}(w_1 \cdots w_{j-1})$ (resp. $\mathrm{P}_{\mathrm{rps}}(w_1 \cdots w_{j-1})$). Then:

- if $r_k \leq w_j$ (resp. $r_k < w_j$), simultaneously attach new boxes, one to $\mathrm{P}_{\mathrm{lps}}(w_1 \cdots w_{j-1})$ (resp. $\mathrm{P}_{\mathrm{rps}}(w_1 \cdots w_{j-1})$) and another to $\mathrm{Q}_{\mathrm{lps}}(w_1 \cdots w_{j-1})$ (resp. $\mathrm{Q}_{\mathrm{rps}}(w_1 \cdots w_{j-1})$), at the right of the bottom row, and fill the first with the symbol w_j and the second with the symbol j;
- otherwise, if $m = \min\{i \in \{1, \ldots, k\} : w_j < r_i\}$, (resp. $m = \min\{i \in \{1, \ldots, k\} : w_j \leq r_i\}$) simultaneously attach one new box on the top of each of the m-th columns of $\mathrm{P}_{\mathrm{lps}}(w_1 \cdots w_{j-1})$ (resp. $\mathrm{P}_{\mathrm{rps}}(w_1 \cdots w_{j-1})$) and $\mathrm{Q}_{\mathrm{lps}}(w_1 \cdots w_{j-1})$ (resp. $\mathrm{Q}_{\mathrm{rps}}(w_1 \cdots w_{j-1})$). Then, insert j in the new box of $\mathrm{Q}_{\mathrm{lps}}(w_1 \cdots w_{j-1})$ (resp. $\mathrm{Q}_{\mathrm{rps}}(w_1 \cdots w_{j-1})$) and in $\mathrm{P}_{\mathrm{lps}}(w_1 \cdots w_{j-1})$ (resp. $\mathrm{P}_{\mathrm{rps}}(w_1 \cdots w_{j-1})$) bump all the symbols of the column containing r_m to the box above and insert w_j in the box which has been cleared and previously contained the symbol r_m.

Output the resulting pair of tableaux.

For brevity, whenever no distinction is needed, $(\mathrm{P}(w), \mathrm{Q}(w))$ will denote one of the pairs of tableaux $(\mathrm{P}_{\mathrm{lps}}(w), \mathrm{Q}_{\mathrm{lps}}(w))$ or $(\mathrm{P}_{\mathrm{rps}}(w), \mathrm{Q}_{\mathrm{rps}}(w))$, obtained from the insertion of a word w under the previous algorithm. Note that the symbols of the PS tableau $\mathrm{P}(w)$ obtained from the insertion of w under the previous algorithm are precisely the symbols occurring in w, which is taken over an arbitrary alphabet, whereas the symbols occurring in $\mathrm{Q}(w)$ are precisely the symbols from $\mathcal{A}_{|w|}$. Moreover, $P(w)$ coincides with the PS tableau obtained from the insertion of w under Algorithm 2.2 and $\mathrm{Q}(w)$ is always a recording tableau. The difference between this algorithm and Algorithm 3.2 of [CMS17] is that we are simultaneously inserting symbols in these tableaux and recording the construction of each new box in $\mathrm{P}(w)$ using the recording tableau $\mathrm{Q}(w)$.

Guided by the example of the plactic monoid, our goal is now to show that starting with a pair of tableaux (P, Q) of the same shape, where P is a PS tableau and Q a recording tableau with content $\mathcal{A}_n$ (where n is the number of entries in P or Q), we are able to associate a unique word whose insertion under Algorithm 3.1 yields (P, Q).

First, we note that for any pair of tableaux of the form $\big(\mathrm{P}(w), \mathrm{Q}(w)\big)$, we will be able to trace back the word w, whose insertion under Algorithm 3.1 led to this pair of tableaux.

$$
\begin{aligned}
&\emptyset, \emptyset \xleftarrow[(4,1)]{} \begin{array}{|c|}\hline 4 \\ \hline\end{array}, \begin{array}{|c|}\hline 1 \\ \hline\end{array} \xleftarrow[(6,2)]{} \begin{array}{|c|c|}\hline 4 & 6 \\ \hline\end{array}, \begin{array}{|c|c|}\hline 1 & 2 \\ \hline\end{array} \xleftarrow[(2,3)]{} \begin{array}{|c|c|}\cline{1-1} 4 & \multicolumn{1}{c}{} \\ \hline 2 & 6 \\ \hline\end{array}, \begin{array}{|c|c|}\cline{1-1} 3 & \multicolumn{1}{c}{} \\ \hline 1 & 2 \\ \hline\end{array} \\
&\xleftarrow[(3,4)]{} \begin{array}{|c|c|}\hline 4 & 6 \\ \hline 2 & 3 \\ \hline\end{array}, \begin{array}{|c|c|}\hline 3 & 4 \\ \hline 1 & 2 \\ \hline\end{array} \xleftarrow[(2,5)]{} \begin{array}{|c|c|}\cline{2-2} \multicolumn{1}{c|}{} & 6 \\ \hline 4 & 3 \\ \hline 2 & 2 \\ \hline\end{array}, \begin{array}{|c|c|}\cline{2-2} \multicolumn{1}{c|}{} & 5 \\ \hline 3 & 4 \\ \hline 1 & 2 \\ \hline\end{array} \xleftarrow[(1,6)]{} \begin{array}{|c|c|}\hline 4 & 6 \\ \hline 2 & 3 \\ \hline 1 & 2 \\ \hline\end{array}, \begin{array}{|c|c|}\hline 6 & 5 \\ \hline 3 & 4 \\ \hline 1 & 2 \\ \hline\end{array} \\
&\xleftarrow[(4,7)]{} \begin{array}{|c|c|c}\cline{1-2} 4 & 6 & \\ \cline{1-2} 2 & 3 & \\ \hline 1 & 2 & \multicolumn{1}{c|}{4} \\ \hline\end{array}, \begin{array}{|c|c|c}\cline{1-2} 6 & 5 & \\ \cline{1-2} 3 & 4 & \\ \hline 1 & 2 & \multicolumn{1}{c|}{7} \\ \hline\end{array} = \mathrm{P}_{\mathrm{lps}}(w), \mathrm{Q}_{\mathrm{lps}}(w).
\end{aligned}
$$

FIGURE 1. Extended lPS insertion of the word $w = 4623214$, where the first coordinate of the pair below the arrow indicates the symbol being inserted and the second coordinate the step at which this symbol is being inserted.

The insertion of a given word $w = w_1 \cdots w_n$ under Algorithm 2.2 is done by inserting each of its symbols, from left to right in the previously obtained tableaux (starting with the empty tableaux $\emptyset$). Thus, using the notation from [CMS17, after Algorithm 3.2], it makes sense to denote the insertion of w in the following way:

$$\mathrm{P}(w) = \mathrm{P}(w_1 w_2 \cdots w_n) = \big((\emptyset \leftarrow w_1) \leftarrow w_2\big) \leftarrow \ldots \leftarrow w_n.$$

Since in the case of Algorithm 3.1, we are simultaneously making two insertions, it makes sense to extend this notation to the following:

$$\big(\mathrm{P}(w), \mathrm{Q}(w)\big) = \Big(\big((\emptyset, \emptyset) \leftarrow (w_1, 1)\big) \leftarrow (w_2, 2)\Big) \leftarrow \ldots \leftarrow (w_n, n).$$

With this notation, the steps of the extended lPS insertion of the word $w = 4623214$ are shown in Figure 1.

Note that each time we insert a symbol into the bottom row of $\mathrm{P}(w)$ using extended PS insertion, a symbol recording this insertion is simultaneously inserted on top of the corresponding column in $\mathrm{Q}(w)$. Denote by $\mathrm{Q}'(w)$ the tableau obtained by reversing the columns of $\mathrm{Q}(w)$. Reading the entries of $\mathrm{P}(w)$ according to the order determined by $\mathrm{Q}'(w)$ allows us to get the word we started with. From Figure 1, we know that if $w = 4623214$, then

$$\mathrm{P}_{\mathrm{lps}}(w) = \begin{array}{|c|c|c}\cline{1-2} 4 & 6 & \\ \cline{1-2} 2 & 3 & \\ \hline 1 & 2 & \multicolumn{1}{c|}{4} \\ \hline\end{array} \quad \text{and} \quad \mathrm{Q}'_{\mathrm{lps}}(w) = \begin{array}{|c|c|c}\cline{1-2} 1 & 2 & \\ \cline{1-2} 3 & 4 & \\ \hline 6 & 5 & \multicolumn{1}{c|}{7} \\ \hline\end{array}.$$

Figure 2 describes the process of reading $\mathrm{P}_{\mathrm{lps}}(w)$ according to $\mathrm{Q}'_{\mathrm{lps}}(w)$.

The *column reading*, denoted $\mathfrak{C}(R)$, of a tableau R is the word obtained by proceeding through the columns, from leftmost to rightmost,

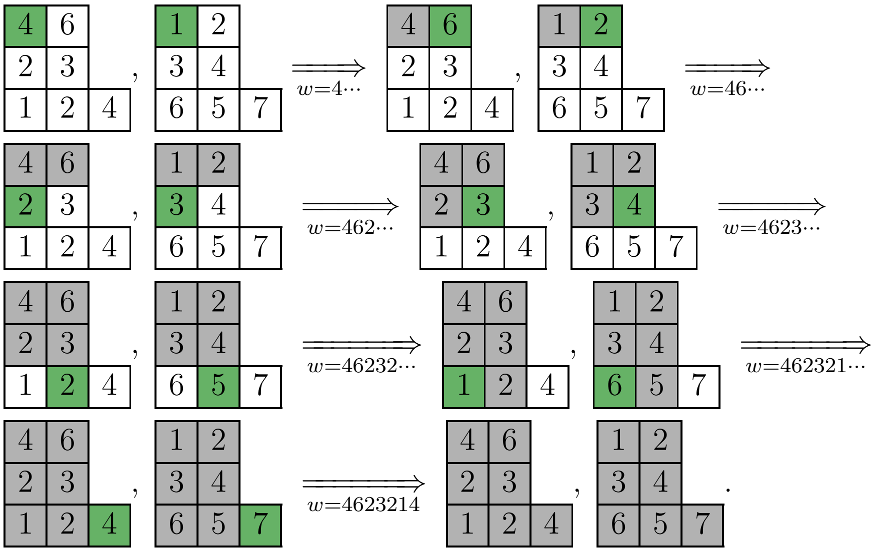

FIGURE 2. Process of obtaining $w = 4623214$ from reading $\mathrm{P}_{\mathrm{lps}}(4623214)$ according to the order given by $\mathrm{Q}'_{\mathrm{lps}}(4623214)$.

and reading each column from top to bottom. For example, the column reading of the tableau

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

is the word $v = 421\,632\,4$. Note that

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

is the recording tableau $\mathrm{Q}_{\mathrm{lps}}(v)$ of the word v.

However, it is not guaranteed that we can start from an arbitrary pair of PS tableaux (R, S) with the same shape, where R is semistandard and S is a recording tableaux, read R according to S, and obtain a word that inserts to (R, S) under Algorithm 3.1. For instance, considering the pair of lPS tableaux

$$(R, S) = \left(\begin{array}{|c|c} \hline 3 & \multicolumn{1}{c}{} \\ \hline 2 & \multicolumn{1}{c}{} \\ \hline 1 & \multicolumn{1}{c|}{1} \\ \hline \end{array}\,, \begin{array}{|c|c} \hline 4 & \multicolumn{1}{c}{} \\ \hline 3 & \multicolumn{1}{c}{} \\ \hline 1 & \multicolumn{1}{c|}{2} \\ \hline \end{array} \right)$$

then reading R according to S' leads to the word $w = 3121$. Using extended lPS insertion, this word inserts to the pair

$$\Big(\mathrm{P}_{\mathrm{lps}}(w), \mathrm{Q}_{\mathrm{lps}}(w)\Big) = \left(\begin{array}{|c|c|}\hline 3&2\\\hline 1&1\\\hline\end{array}, \begin{array}{|c|c|}\hline 2&4\\\hline 1&3\\\hline\end{array}\right) \neq \left(\begin{array}{|c|c}\cline{1-1} 3 & \\\cline{1-1} 2 & \\\hline 1&\multicolumn{1}{c|}{1}\\\hline\end{array}, \begin{array}{|c|c}\cline{1-1} 4 & \\\cline{1-1} 3 & \\\hline 1&\multicolumn{1}{c|}{2}\\\hline\end{array}\right).$$

With another simple example we can conclude that the same problem can occur when considering pairs of rPS tableaux.

Within the setting of standard words and pairs of standard PS tableaux, this problem led the authors of [BL07] to restrict the pairs of standard PS tableaux that can be considered by introducing the concept of *stable pairs set*.

In order to introduce this notion, first we need the concept of pattern avoidance, which is also defined in [BL07]. So, given standard words u and v with $|v| = m \leq n = |u|$, we say that u *contains the pattern* v, if there exists a subword $u_{i_1} \cdots u_{i_m}$ of u of length m that is order isomorphic to v. Otherwise, we say that u *avoids the pattern* v.

Note that in the above definition, the symbols $u_{i_1}, \ldots, u_{i_m}$ are not required to be contiguous. However, the definition that we will adopt henceforth is that, unless a dash is inserted in v indicating which of the symbols $u_{i_1}, \ldots, u_{i_m}$ are not required to be contiguous, they have to be contiguous.

For example

- the standard word $\sigma = 3142$ contains exactly one occurrence of a 2-31 pattern given by the subword $u = 342$;
- the standard word $\mathrm{std}_{\mathrm{l}}(2312) = 2_1 3_1 1_1 2_2$ contains exactly one occurrence of a 2-13 pattern given by the subword $u = 2_1 1_1 2_2$.

The *standard stable pairs set over an alphabet*, is the set composed by the pairs of PS tableaux (R, S) of the same shape, such that R is a standard tableaux over that alphabet, S is a recording tableau and the pair of column readings $(\mathfrak{C}(R), \mathfrak{C}(S'))$ avoids simultaneous occurrences of the pairs of patterns $(31\text{-}2, 13\text{-}2)$, $(31\text{-}2, 23\text{-}1)$ and $(32\text{-}1, 13\text{-}2)$ at the same positions of $\mathfrak{C}(R)$ and $\mathfrak{C}(S')$.

Within this setting, using Algorithm 3.1, Theorem 3.9 of [BL07] can be restated in the following way

Proposition 3.2. *There is a one-to-one correspondence between the set of standard words over an alphabet and the standard stable pairs set over that alphabet. The one-to-one correspondence is given under the mapping*

$$w \mapsto (\mathrm{P}(w), \mathrm{Q}(w)).$$

Since we aim to present a Robinson–Schensted-like correspondence, it is natural to consider the following generalization of the standard stable set previously introduced. For $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$, the xPS *stable pairs* *set over an alphabet* $\mathcal{B}$ is the set composed by the pairs of xPS tableaux (R, S) such that $\bigl(\mathrm{Std}_{\mathrm{x}}\bigl(R\bigr), S\bigr)$ is a standard stable pair over $\mathcal{C}(\mathcal{B})$.

We are now able to introduce the Robinson–Schensted-like correspondence for the words case.

Theorem 3.3. *For* $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$*, there is a one-to-one correspondence between the set of words over an alphabet and pairs of* x*PS tableaux of the* x*PS stable pairs set over that alphabet. The one-to-one correspondence is given under the mapping*

$$w \mapsto (\mathrm{P}_{\mathrm{xps}}(w), \mathrm{Q}_{\mathrm{xps}}(w)).$$

Proof. Let $\mathcal{B}$ be an alphabet and $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$. Given a word w over $\mathcal{B}$, we obtain a unique standard word $\mathrm{std}_{\mathrm{x}}(w)$ over $\mathcal{C}(\mathcal{B})$. Using Proposition 3.2, we obtain a pair $(\mathrm{P}_{\mathrm{xps}}(\mathrm{std}_{\mathrm{x}}(w)), \mathrm{Q}_{\mathrm{xps}}(\mathrm{std}_{\mathrm{x}}(w)))$ in the standard stable pairs set over $\mathcal{C}(\mathcal{B})$. By Proposition 2.6 (1), $\mathrm{P}_{\mathrm{xps}}(\mathrm{std}_{\mathrm{x}}(w)) = \mathrm{Std}_{\mathrm{x}}(\mathrm{P}_{\mathrm{xps}}(w))$. Since w and $\mathrm{std}_{\mathrm{x}}(w)$ have equivalent xPS insertions, $\mathrm{Q}_{\mathrm{xps}}(\mathrm{std}_{\mathrm{x}}(w)) = \mathrm{Q}_{\mathrm{xps}}(w)$. Thus, it follows that $(\mathrm{P}_{\mathrm{xps}}(\mathrm{std}_{\mathrm{x}}(w)), \mathrm{Q}_{\mathrm{xps}}(\mathrm{std}_{\mathrm{x}}(w))) = (\mathrm{Std}_{\mathrm{x}}(\mathrm{P}_{\mathrm{xps}}(w)), \mathrm{Q}_{\mathrm{xps}}(w))$. So, we obtain a pair of xPS tableaux $(\mathrm{P}_{\mathrm{xps}}(w), \mathrm{Q}_{\mathrm{xps}}(w))$ in the xPS stable pairs set over $\mathcal{B}$. Note that different words lead to different pairs.

Given a pair of xPS tableaux (R, S) in the xPS stable pairs set over $\mathcal{B}$, by definition $(\mathrm{Std}_{\mathrm{x}}(R), S)$ is in the standard stable pairs set over $\mathcal{C}(\mathcal{B})$. By Proposition 3.2, there exists a standard word over $\mathcal{C}(\mathcal{B})$ that inserts to $(\mathrm{Std}_{\mathrm{x}}(R), S)$. From Lemma 2.5, this word must have the form $\mathrm{std}_{\mathrm{x}}(w)$ for some word w over $\mathcal{B}$. Thus, $(\mathrm{P}_{\mathrm{xps}}(\mathrm{std}_{\mathrm{x}}(w)), \mathrm{Q}_{\mathrm{xps}}(\mathrm{std}_{\mathrm{x}}(w))) = (\mathrm{Std}_{\mathrm{x}}(R), S)$. Since by Proposition 2.6 (1), $\mathrm{P}_{\mathrm{xps}}(\mathrm{std}_{\mathrm{x}}(w)) = \mathrm{Std}_{\mathrm{x}}(\mathrm{P}_{\mathrm{xps}}(w))$, and because $\mathrm{Q}_{\mathrm{xps}}(\mathrm{std}_{\mathrm{x}}(w)) = \mathrm{Q}_{\mathrm{xps}}(w)$, it follows that $(\mathrm{Std}_{\mathrm{x}}(\mathrm{P}_{\mathrm{xps}}(w)), \mathrm{Q}_{\mathrm{xps}}(w)) = (\mathrm{Std}_{\mathrm{x}}(R), S)$. Applying Dstd() to both $\mathrm{Std}_{\mathrm{x}}(\mathrm{P}_{\mathrm{xps}}(w))$ and $\mathrm{Std}_{\mathrm{x}}(R)$, we have $(\mathrm{P}_{\mathrm{xps}}(w), \mathrm{Q}_{\mathrm{xps}}(w)) = (R, S)$. Therefore under Algorithm 3.1, the word $w = \mathrm{std}^{-1}(\mathrm{std}_{\mathrm{x}}(w))$ over $\mathcal{B}$ inserts to the pair (R, S). $\square$

3.2. **A Robinson–Schensted–Knuth-like correspondence.** It is common to present a standard word $\sigma = \sigma_1 \sigma_2 \cdots \sigma_k$ in two-line notation as

$$\sigma = \begin{pmatrix} 1 & 2 & \cdots & k \\ \sigma_1 & \sigma_2 & \cdots & \sigma_k \end{pmatrix}.$$

This notation can be extended to arbitrary words, $w = w_1 w_2 \cdots w_k$, with symbols $w_1, w_2, \ldots, w_k$, in the following way:

$$w = \begin{pmatrix} 1 & 2 & \cdots & k \\ w_1 & w_2 & \cdots & w_k \end{pmatrix}.$$

Looking to Algorithm 3.1 from this perspective, we observe that it just describes the simultaneous insertion of the words $w = w_1 w_2 \cdots w_k$ and $12 \cdots k$, where the standard word $12 \cdots k$ is inserted according to the PS insertion of w.

Hence, seeing things from the two row notation perspective, Proposition 3.2 establishes a bijection between standard words $\sigma = \left(\begin{smallmatrix} 1 & 2 & \cdots & k \\ \sigma_1 & \sigma_2 & \cdots & \sigma_k \end{smallmatrix}\right)$

and pairs of standard PS tableaux $\big(\mathrm{P}(\sigma), \mathrm{Q}(\sigma)\big)$ in the standard stable pairs set. In the same way, Theorem 3.3 establishes a bijection between words $w = \left(\begin{smallmatrix} 1 & 2 & \cdots & k \\ w_1 & w_2 & \cdots & w_k \end{smallmatrix}\right)$ and pairs of standard PS tableaux $\big(\mathrm{P}(w), \mathrm{Q}(w)\big)$ in the stable pairs set. This perspective suggests a generalization of Algorithm 3.1 to two-rowed arrays $\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)$, where u and v are words of the same length and a consequent generalization of both the Proposition 3.2 and the Theorem 3.3.

So, henceforth a *two-rowed array over an alphabet* will be an array $\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)$ where u and v are arbitrary words of the same length over that alphabet (note that word arrays are not biwords in the sense of [Lot02]). We propose the following generalization of Algorithm 3.1:

Algorithm 3.4 (Extended PS insertion of a word array)**.**

Input: A two-rowed array $w = \left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)$ over an alphabet.

Output: A pair of tableaux $(\mathrm{P}_{\mathrm{lps}}(w), \mathrm{Q}_{\mathrm{lps}}(w))$ (resp. $(\mathrm{P}_{\mathrm{rps}}(w), \mathrm{Q}_{\mathrm{rps}}(w))$) of the same shape.

Method:

(1) If $v = \varepsilon$ and $u = \varepsilon$, output a pair of empty tableaux $(\emptyset, \emptyset)$. Otherwise:

(2) $v = v_1 \cdots v_n$ and $u = u_1 \cdots u_n$, for symbols $v_1, \ldots, v_n$ and $u_1, \ldots, u_n$. Set

$$\Big(\mathrm{P}_{\mathrm{lps}}\big(\left(\begin{smallmatrix} u_1 \\ v_1 \end{smallmatrix}\right)\big), \mathrm{Q}_{\mathrm{lps}}\big(\left(\begin{smallmatrix} u_1 \\ v_1 \end{smallmatrix}\right)\big)\Big) = \big(\boxed{v_1}, \boxed{u_1}\big) = \Big(\mathrm{P}_{\mathrm{rps}}\big(\left(\begin{smallmatrix} u_1 \\ v_1 \end{smallmatrix}\right)\big), \mathrm{Q}_{\mathrm{rps}}\big(\left(\begin{smallmatrix} u_1 \\ v_1 \end{smallmatrix}\right)\big)\Big),$$

and for each remaining symbol v_j with $1 < j \leq n$, denote by $r_1 \leq \cdots \leq r_k$ (resp. $r_1 < \cdots < r_k$) the symbols in the bottom row of the tableau $\mathrm{P}_{\mathrm{lps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$ (resp. $\mathrm{P}_{\mathrm{rps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$). Then:

- if $r_k \leq v_j$ (resp. $r_k < v_j$), simultaneously attach two new boxes, one to $\mathrm{P}_{\mathrm{lps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$ (resp. $\mathrm{P}_{\mathrm{rps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$) and the other to $\mathrm{Q}_{\mathrm{lps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$ (resp. $\mathrm{Q}_{\mathrm{rps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$), at the right of the bottom row, and fill the first with the symbol v_j and the second with the symbol u_j;
- otherwise, if $m = \min\{i \in \{1, \ldots, k\} : v_j < r_i\}$, (resp. $m = \min\{i \in \{1, \ldots, k\} : v_j \leq r_i\}$) simultaneously attach one new box on top of each of the m-th columns of $\mathrm{P}_{\mathrm{lps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$ (resp. $\mathrm{P}_{\mathrm{rps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$) and $\mathrm{Q}_{\mathrm{lps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$ (resp. $\mathrm{Q}_{\mathrm{rps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$). Then, insert u_j in the new box of $\mathrm{Q}_{\mathrm{lps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$ (resp. $\mathrm{Q}_{\mathrm{rps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$) and in $\mathrm{P}_{\mathrm{lps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$ (resp. $\mathrm{P}_{\mathrm{rps}}\big(\left(\begin{smallmatrix} u_1 & \cdots & u_{j-1} \\ v_1 & \cdots & v_{j-1} \end{smallmatrix}\right)\big)$) bump all the symbols of the column containing r_m to the box above and insert v_j in the box which has been cleared and previously contained the symbol r_m.

Output the resulting pair of tableaux.

Considering the two-rowed array $w = \left(\begin{smallmatrix} 1 & 2 & 1 & 3 \\ 1 & 1 & 2 & 1 \end{smallmatrix}\right)$, the extended lPS insertion of the two-rowed array under Algorithm 3.4 is given by the pair of tableaux

$$\big(\mathrm{P_{lps}}(w), \mathrm{Q_{lps}}(w)\big) = \left(\begin{array}{|c|c|c|} \cline{3-3} \multicolumn{2}{c|}{} & 2 \\ \hline 1 & 1 & 1 \\ \hline \end{array}\,,\ \begin{array}{|c|c|c|} \cline{3-3} \multicolumn{2}{c|}{} & 3 \\ \hline 1 & 2 & 1 \\ \hline \end{array}\right)$$

The first observation that comes out from this example is that if w is an arbitrary two-rowed array, then $\mathrm{Q_{lps}}(w)$ is not necessarily an lPS tableau. In fact, considering the same two-rowed array we can conclude that $\mathrm{Q_{rps}}(w)$ is also not necessarily an rPS tableau.

However, we are interested in finding those two-rowed arrays from which we can always obtain a pair of PS tableaux.

Since for symbols $y_1 < y_2$, a two-rowed array $\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right) \in \left\{\left(\begin{smallmatrix} y_2 & y_1 \\ y_1 & y_2 \end{smallmatrix}\right), \left(\begin{smallmatrix} y_2 & y_1 \\ y_2 & y_1 \end{smallmatrix}\right), \left(\begin{smallmatrix} y_2 & y_1 \\ y_1 & y_1 \end{smallmatrix}\right)\right\}$ gives rise to a tableau $\mathrm{Q_{xps}}(w)$, for $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$, which is not an xPS tableau, we conclude that in order to obtain a pair of xPS tableau, u has to be ordered weakly increasingly. However, there are still two-rowed arrays $w = \left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)$ where u is ordered weakly increasingly, such that $\mathrm{Q_{xps}}(w)$ is not an xps tableau. For instance, for symbols $y_1 < y_2$ and two-rowed arrays $w = \left(\begin{smallmatrix} y_1 & y_1 \\ y_2 & y_1 \end{smallmatrix}\right)$ and $w' = \left(\begin{smallmatrix} y_1 & y_1 \\ y_1 & y_2 \end{smallmatrix}\right)$, we have that $\mathrm{Q_{lps}}(w)$ is not an lPS tableau and $\mathrm{Q_{rps}}(w')$ is not an rPS tableau.

This leads us to two different types of two-rowed arrays. For any two-rowed array $\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right) = \left(\begin{smallmatrix} u_1 & u_2 & \cdots & u_k \\ v_1 & v_2 & \cdots & v_k \end{smallmatrix}\right)$, consider the following conditions:

(1) $u_1 \leq u_2 \leq \cdots \leq u_k$;
(2) for $i \in \{1, 2, \ldots, k-1\}$, if $u_i = u_{i+1}$, then $v_i \leq v_{i+1}$;
(3) for $i \in \{1, 2, \ldots, k-1\}$, if $u_i = u_{i+1}$, then $v_{i+1} \leq v_i$.

The two-rowed array $\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)$ is said to be in *lexicographic order* if conditions (1) and (2) hold, and that it is in *reverse lexicographic order* if conditions (1) and (3) hold. For simplicity, we will refer the two-rowed array as an *l*-two-rowed array, in the first case, and as a *r*-two-rowed array in the second.

Lemma 3.5. *For any* $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$*, if* $w = \left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right) = \left(\begin{smallmatrix} u_1 & u_2 & \cdots & u_k \\ v_1 & v_2 & \cdots & v_k \end{smallmatrix}\right)$ *is an* x*-two-rowed array, then the tableau* $\mathrm{Q_{xps}}(w)$ *obtained from the extended* x*PS insertion of* w *under Algorithm 3.4 is an* x*PS tableau.*

Proof. Both the lPS and rPS cases follow by induction on the number n of distinct symbols from u.

For the lPS case, the induction proceeds in the following way. Case $n = 1$. In this case $w = \left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right) = \left(\begin{smallmatrix} u_1 & \cdots & u_1 \\ v_1 & \cdots & v_k \end{smallmatrix}\right)$. Since $v_1 \leq v_2 \leq \cdots \leq v_k$, by Algorithm 3.4, $\big(\mathrm{P_{lps}}(w), \mathrm{Q_{lps}}(w)\big) = \left(\boxed{v_1}\boxed{v_2}\cdots\boxed{v_k}, \boxed{u_1}\boxed{u_1}\cdots\boxed{u_1}\right)$ and thus $\mathrm{Q_{lps}}(w)$ is an lPS tableau.

Fix $n \geq 1$ and suppose by induction hypothesis that the result holds for n. Let us prove the result for $n+1$. Having $n+1$ distinct symbols, $u = x_1^{i_1} x_2^{i_2} \cdots x_{n+1}^{i_{n+1}}$, for some symbols $x_1 < x_2 < \cdots < x_{n+1}$ and indexes

$i_1, i_2, \ldots, i_{n+1} \in \mathbb{N}$. Thus

$$w = \begin{pmatrix} x_1 & \cdots & x_1 & x_2 & \cdots & x_2 & \cdots & x_{n+1} \\ v_1 & \cdots & v_{i_1} & v_{i_1+1} & \cdots & v_{i_1+i_2} & \cdots & v_k \end{pmatrix}.$$

Since $v_{i_1+\cdots+i_n+1} \leq \cdots \leq v_k$, according to Algorithm 3.4, for any $m \in \{i_1 + \cdots + i_n + 2, \ldots, k\}$, v_m is inserted in the bottom row in a column to the right of the column where v_{m-1} was inserted. Therefore each of the symbols x_{n+1} from u is inserted either on top of a column not containing x_{n+1} or as the rightmost symbol of the bottom row. Since by induction hypothesis $\mathrm{Q}_{\mathrm{lps}}\left(\left(\begin{smallmatrix} x_1^{i_1} & \cdots & x_n^{i_n} \\ v_1 & \cdots & v_{i_1+\cdots+i_n} \end{smallmatrix}\right)\right)$ is an lPS tableau, from this discussion we conclude that the columns $\mathrm{Q}_{\mathrm{lps}}(w)$ are strictly decreasing top to bottom and the bottom row is weakly increasing from left to right. Therefore it follows that $\mathrm{Q}_{\mathrm{lps}}(w)$ is an lPS tableau.

Regarding the rPS case, the induction proceeds in the following way. Case n=1. Recall that $w = \left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right) = \left(\begin{smallmatrix} u_1 & \cdots & u_1 \\ v_1 & \cdots & v_k \end{smallmatrix}\right)$. Since $v_k \leq \cdots \leq v_2 \leq v_1$, it follows that

$$\mathrm{P}_{\mathrm{rps}}(w) = \begin{array}{|c|} \hline v_k \\ \hline \vdots \\ \hline v_2 \\ \hline v_1 \\ \hline \end{array} \quad \text{and thus} \quad \mathrm{Q}_{\mathrm{rps}}(w) = \begin{array}{|c|} \hline u_1 \\ \hline \vdots \\ \hline u_1 \\ \hline u_1 \\ \hline \end{array}.$$

So, we conclude that $\mathrm{Q}_{\mathrm{rps}}(w)$ is an rPS tableau.

Fix $n \geq 1$ and suppose by induction hypothesis that the result holds for n. Since u contains $n+1$ symbols $x_1 < x_2 < \cdots < x_{n+1}$, again $u = x_1^{i_1} x_2^{i_2} \cdots x_{n+1}^{i_{n+1}}$, and

$$w = \begin{pmatrix} x_1 & \cdots & x_1 & x_2 & \cdots & x_2 & \cdots & x_{n+1} \\ v_1 & \cdots & v_{i_1} & v_{i_1+1} & \cdots & v_{i_1+i_2} & \cdots & v_k \end{pmatrix}.$$

According to Algorithm 3.4, the symbol $v_{i_1+\cdots+i_n+1}$ is inserted in the bottom row of $\mathrm{P}_{\mathrm{rps}}\left(\left(\begin{smallmatrix} x_1^{i_1} & \cdots & x_n^{i_n} \\ v_1 & \cdots & v_{i_1+\cdots+i_n} \end{smallmatrix}\right)\right)$, either as the rightmost symbol or in a column of this tableau. Since $v_k \leq \cdots \leq v_{i_1+\cdots+i_n+1}$, for any $m \in \{i_1 + \cdots + i_n + 2, \ldots, k\}$, v_m is going to be inserted in the bottom row below or to the left of the column where v_{m-1} was inserted. So, the first symbol x_{n+1} of u is either inserted to the right of the rightmost box, or on top of a column of $\mathrm{Q}_{\mathrm{rps}}\left(\left(\begin{smallmatrix} x_1^{i_1} & \cdots & x_n^{i_n} \\ v_1 & \cdots & v_{i_1+\cdots+i_n} \end{smallmatrix}\right)\right)$. The remaining symbols x_{n+1} are always inserted on top of a column.

Since by induction hypothesis $\mathrm{Q}_{\mathrm{rps}}\left(\left(\begin{smallmatrix} x_1^{i_1} & \cdots & x_n^{i_n} \\ v_1 & \cdots & v_{i_1+\cdots+i_n} \end{smallmatrix}\right)\right)$ is an rPS tableau, the previous discussion allows us to conclude that $\mathrm{Q}_{\mathrm{rps}}(w)$ is an rPS tableau. $\square$

As in the previous cases, we are interested in describing the reverse process: that is, starting from a pair of PS tableaux our goal is to obtain the unique two-rowed array that gives rise to this pair under Algorithm 3.4. The proof of Lemma 3.5 will allow us to describe it, and suggests that the lPS and rPS versions of the reverse process will be different.

The lPS *reverse insertion method* can be described as follows: let (R, S) be a pair of lPS tableaux of the same shape. Then, $|R| = |S| = k$, for some $k \in \mathbb{N}$. Let $(R, S) = (R_k, S_k)$, and for any $i \in \{1, \ldots, k-1\}$, let (R_i, S_i) be the pair of tableaux (of the same shape) obtained from removing in S_{i+1} the box containing the largest symbol that is farthest to the right, u_{i+1}, and in R_{i+1} the box containing the symbol v_{i+1} that is in the bottom of the corresponding column of R_{i+1}. From this process we obtain a two-rowed array $\binom{u}{v} = \binom{u_1\ u_2\ \cdots\ u_k}{v_1\ v_2\ \cdots\ v_k}$ and it is clear that $u_1 \leq u_2 \leq \cdots \leq u_k$.

Moreover, assuming that for each i, (R_i, S_i) is a pair of lPS tableaux, then if $u_i = u_{i+1}$ for some $i \in \{1, \ldots, k-1\}$, then u_{i+1} and u_i are in distinct columns and u_{i+1} is farthest to the right. So, if v_{i+1} is the symbol in the box removed from the bottom row of R_{i+1}, then v_{i+1} is farther to the right than the symbol v_i in the box removed from the bottom row of P_i. By the proof of the lPS case of Lemma 3.5, it follows that $v_i \leq v_{i+1}$. Therefore, the two-rowed array $\binom{u}{v}$ obtained via this method is an l-two-rowed array.

Similarly, consider the rPS *reverse insertion method* described as follows. Let (R, S) be a pair of rPS tableaux of the same shape. Then, $|R| = |S| = k$, for some $k \in \mathbb{N}$. Let $(R, S) = (R_k, S_k)$, and for any $i \in \{1, \ldots, k-1\}$, let (R_i, S_i) be the pair of tableaux obtained from removing in S_{i+1} the box containing the largest symbol that is farthest to the left on top of the respective column, u_{i+1}, and in R_{i+1} the box containing the symbol v_{i+1} that is in the bottom of the corresponding column.

Again, if $\binom{u}{v} = \binom{u_1\ u_2\ \cdots\ u_k}{v_1\ v_2\ \cdots\ v_k}$ is the two-rowed array that is obtained via this process, then it is clear that $u_1 \leq u_2 \leq \cdots \leq u_k$. Assuming that (R_i, S_i) is a pair of rPS tableaux for any $i \in \{1, \ldots, k-1\}$, if $u_i = u_{i+1}$ for some $i \in \{1, \ldots, k-1\}$, then u_{i+1} and u_i are either in distinct columns, u_{i+1} being farther to the left, or u_{i+1} and u_i are in the same column, u_{i+1} being on top of u_i. In the first case, if v_{i+1} is the symbol in the box removed from the bottom row of R_{i+1}, then v_{i+1} is farther to the left than the symbol v_i removed from the box in the bottom row of R_i. By the proof of the rPS case of Lemma 3.5, it follows that $v_{i+1} < v_i$. In the later case, if v_{i+1} is the symbol in the box removed from the bottom row of R_{i+1}, then v_{i+1} is below the symbol removed from the box in the bottom row of R_i, v_i. By the proof of the rPS case of Lemma 3.5, $v_{i+1} \leq v_i$. Therefore, the two-rowed array $\binom{u}{v}$ obtained through this method is an r-two-rowed array.

Figure 3 shows the extended lPS insertion of the l-two-rowed array $w = \binom{1\,1\,2\,3\,3\,3\,4}{3\,4\,2\,1\,1\,2\,3}$ under Algorithm 3.4, while Figure 4 describes the construction of the original l-two-rowed array $w = \binom{1\,1\,2\,3\,3\,3\,4}{3\,4\,2\,1\,1\,2\,3}$ from

$$\emptyset, \emptyset \xleftarrow[(3,1)]{} \begin{array}{|c|}\hline 3\\\hline\end{array}, \begin{array}{|c|}\hline 1\\\hline\end{array} \xleftarrow[(4,1)]{} \begin{array}{|c|c|}\hline 3&4\\\hline\end{array}, \begin{array}{|c|c|}\hline 1&1\\\hline\end{array} \xleftarrow[(2,2)]{} \begin{array}{ll} 3 & \\ 2 & 4 \end{array}, \begin{array}{ll} 2 & \\ 1 & 1 \end{array}$$

$$\xleftarrow[(1,3)]{} \begin{array}{ll} 3 & \\ 2 & \\ 1 & 4 \end{array}, \begin{array}{ll} 3 & \\ 2 & \\ 1 & 1 \end{array} \xleftarrow[(1,3)]{} \begin{array}{ll} 3 & \\ 2 & 4 \\ 1 & 1 \end{array}, \begin{array}{ll} 3 & \\ 2 & 3 \\ 1 & 1 \end{array} \xleftarrow[(2,3)]{} \begin{array}{lll} 3 & & \\ 2 & 4 & \\ 1 & 1 & 2 \end{array}, \begin{array}{lll} 3 & & \\ 2 & 3 & \\ 1 & 1 & 3 \end{array}$$

$$\xleftarrow[(3,4)]{} \begin{array}{llll} 3 & & & \\ 2 & 4 & & \\ 1 & 1 & 2 & 3 \end{array}, \begin{array}{llll} 3 & & & \\ 2 & 3 & & \\ 1 & 1 & 3 & 4 \end{array} = \mathrm{P}_{\mathrm{lps}}(w), \mathrm{Q}_{\mathrm{lps}}(w).$$

FIGURE 3. Extended lPS insertion of the l-two-rowed array $w = \binom{u}{v} = \left(\begin{smallmatrix} 1 & 1 & 2 & 3 & 3 & 3 & 4 \\ 3 & 4 & 2 & 1 & 1 & 2 & 3 \end{smallmatrix}\right)$ under Algorithm 3.4, where the pair below each arrow indicates the pair of symbols from w being inserted.

that pair of tableaux according to the lPS reverse insertion method previously described.

As in the words case, there are pairs of PS tableaux (P, Q) such that the word array, obtained from the PS reverse insertion applied to (P, Q), does not insert to (P, Q) under Algorithm 3.4. For instance, the reverse lPS insertion of the pair of lPS tableaux

$$\left(\begin{array}{ll} 3 & \\ 2 & \\ 1 & 1 \end{array}, \begin{array}{ll} 3 & \\ 2 & \\ 1 & 1 \end{array} \right)$$

leads to the two-rowed array $w = \left(\begin{smallmatrix} 1 & 1 & 2 & 3 \\ 3 & 1 & 2 & 1 \end{smallmatrix}\right)$, which under the lPS version of Algorithm 3.4 inserts to the pair

$$\big(\mathrm{P}_{\mathrm{lps}}(w), \mathrm{Q}_{\mathrm{lps}}(w)\big) = \left(\begin{array}{|c|c|}\hline 3 & 2 \\\hline 1 & 1 \\\hline\end{array}, \begin{array}{|c|c|}\hline 1 & 3 \\\hline 1 & 2 \\\hline\end{array} \right).$$

In fact, from this example we conclude that there are pairs of PS tableaux such that when applying the PS reverse insertion method we arrive to a word array that is not a PS word array. So, just as in the words case, we will have to restrict the pairs of PS tableaux that we can consider.

We will need auxiliary results in order to prove the existence of a Robinson–Schensted–Knuth-like correspondence for PS tableaux.

Lemma 3.6. *Given* $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$, *for any x-two-rowed array* $\binom{u}{v}$ *we have*

$$\mathrm{Std}_{\mathrm{x}}\Big(\mathrm{Q}_{\mathrm{xps}}\big(\tbinom{u}{v}\big)\Big) = \mathrm{Q}_{\mathrm{xps}}\Big(\tbinom{\mathrm{std}_{\mathrm{l}}(u)}{v}\Big).$$

Proof. Let $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$ and let $w = \binom{u}{v} = \left(\begin{smallmatrix} u_1 & u_2 & \cdots & u_k \\ v_1 & v_2 & \cdots & v_k \end{smallmatrix}\right)$ be an arbitrary x-two-rowed array. Then $u = x_1^{i_1} \cdots x_n^{i_n}$ for some symbols $x_1 < \cdots < x_n$

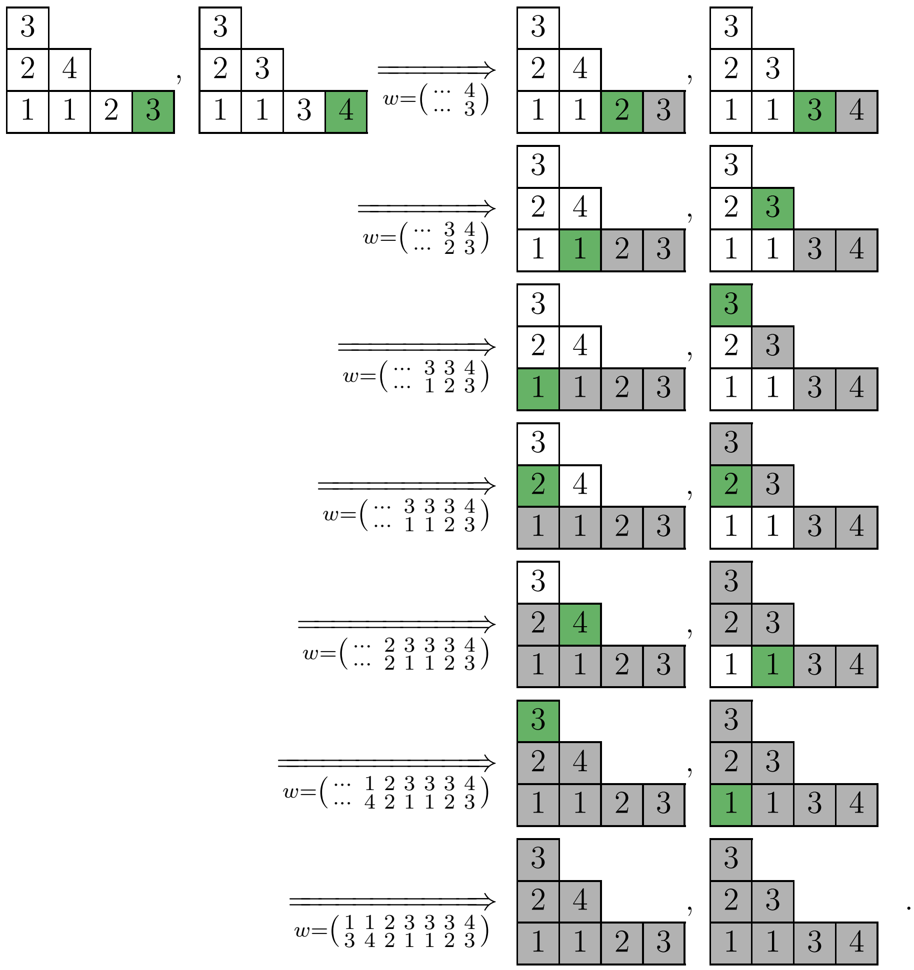

FIGURE 4. Process of obtaining the l-two-rowed array $w = \left(\begin{smallmatrix} 1 & 1 & 2 & 3 & 3 & 3 & 4 \\ 3 & 4 & 2 & 1 & 1 & 2 & 3 \end{smallmatrix}\right)$ through the lPS reverse insertion of the pair of lPS tableaux $(\mathrm{P}_{\mathrm{lps}}(w), \mathrm{Q}_{\mathrm{lps}}(w))$.

and indexes $i_1, \ldots, i_n \in \mathbb{N}$ such that $i_1 + \cdots + i_n = k$. Thus,

$$w = \begin{pmatrix} x_1 & \cdots & x_1 & x_2 & \cdots & x_2 & \cdots & x_n \\ v_1 & \cdots & v_{i_1} & v_{i_1+1} & \cdots & v_{i_1+i_2} & \cdots & v_k \end{pmatrix}.$$

Since $u = x_1^{i_1} \cdots x_n^{i_n}$, then $\mathrm{std}_{\mathrm{l}}(u) = (x_1)_1 \cdots (x_1)_{i_1} \cdots (x_n)_1 \cdots (x_n)_{i_n}$. Let $i_0 = 0$. In the lPS case (resp., rPS), since w is a l-two-rowed word, for any $j \in \{0, \ldots, n-1\}$, $m \in \{1, \ldots, i_{j+1} - 1\}$ we have

$$v_{i_1+\cdots+i_j+m} \leq v_{i_1+\cdots+i_j+m+1}$$

(resp., $\geq$). By Algorithm 3.4, it follows that for any $k \in \{1, \ldots, n\}$, if $(x_k)_p < (x_k)_q$ for some $p, q \in \{1, \ldots, i_k\}$ with $p < q$, then in $\mathrm{Q}_{\mathrm{lps}}\left(\left(\begin{smallmatrix} \mathrm{std}_{\mathrm{l}}(u) \\ v \end{smallmatrix}\right)\right)$ (resp., $\mathrm{Q}_{\mathrm{rps}}\left(\left(\begin{smallmatrix} \mathrm{std}_{\mathrm{l}}(u) \\ v \end{smallmatrix}\right)\right)$), the symbol $(x_k)_q$ is in a column to the right of the column containing the symbol $(x_k)_p$ (resp., either in the same column of $(x_k)_p$ on top of it or in a column to the left of it).

Since $\mathrm{Q}_{\mathrm{lps}}(w)$ (resp., $\mathrm{Q}_{\mathrm{rps}}(w)$) is an lPS (resp., rPS) tableau, by the process of left (resp., right) standardization of a tableau, it follows that for any $k \in \{1, \ldots, n\}$, if $(x_k)_p < (x_k)_q$ for some $p, q \in \{1, \ldots, i_k\}$ with $p < q$, then in $\mathrm{Q}_{\mathrm{lps}}\big(\binom{\mathrm{std}_{\mathrm{l}}(u)}{v}\big)$ (resp., $\mathrm{Q}_{\mathrm{rps}}\big(\binom{\mathrm{std}_{\mathrm{l}}(u)}{v}\big)$), the symbol $(x_k)_q$ is in a column to the right of the column containing the symbol $(x_k)_p$ (resp., either in the same column of $(x_k)_p$ on top of it or in a column to the left of it).

As both u and $\mathrm{std}_{\mathrm{l}}(u)$ are inserted according to the lPS (resp., rPS) insertion of v, it follows that the underlying indexes of symbols in the same position of $\mathrm{Q}_{\mathrm{lps}}(w)$ and $\mathrm{Q}_{\mathrm{lps}}\big(\binom{\mathrm{std}_{\mathrm{l}}(u)}{v}\big)$ (resp., $\mathrm{Q}_{\mathrm{rps}}(w)$ and $\mathrm{Q}_{\mathrm{rps}}\big(\binom{\mathrm{std}_{\mathrm{l}}(u)}{v}\big)$) are the same. This together with the previous considerations allow us to conclude that $\mathrm{Std}_{\mathrm{l}}(\mathrm{Q}_{\mathrm{lps}}(w)) = \mathrm{Q}_{\mathrm{lps}}\big(\binom{\mathrm{std}_{\mathrm{l}}(u)}{v}\big)$ (resp., $\mathrm{Std}_{\mathrm{r}}(\mathrm{Q}_{\mathrm{rps}}(w)) = \mathrm{Q}_{\mathrm{rps}}\big(\binom{\mathrm{std}_{\mathrm{l}}(u)}{v}\big)$). $\square$

Proposition 3.7. *For* $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$ *and any* x*-two-rowed array* $w = \binom{u}{v}$ *we have:*

(*i*) $\mathrm{Std}_{\mathrm{x}}(\mathrm{Q}_{\mathrm{xps}}(w)) = \mathrm{Q}_{\mathrm{xps}}\big(\binom{\mathrm{std}_{\mathrm{l}}(u)}{v}\big) = \mathrm{Q}_{\mathrm{xps}}\Big(\binom{\mathrm{std}_{\mathrm{l}}(u)}{\mathrm{std}_{\mathrm{x}}(v)}\Big)$;

(*ii*) $\mathrm{Std}_{\mathrm{x}}\big(\mathrm{P}_{\mathrm{xps}}(w)\big) = \mathrm{P}_{\mathrm{xps}}\Big(\binom{\mathrm{std}_{\mathrm{l}}(u)}{\mathrm{std}_{\mathrm{x}}(v)}\Big) = \mathrm{P}_{\mathrm{xps}}\Big(\binom{u}{\mathrm{std}_{\mathrm{x}}(v)}\Big)$.

Proof. The first equality of (*i*) is just Lemma 3.6. The second equality follows from the fact that the first component of the pair obtained from the extended xPS insertion of the x-two-rowed array $\binom{u}{v}$ under Algorithm 3.4 is equal to the insertion of v under Algorithm 2.2 together with the fact that v and $\mathrm{std}_{\mathrm{x}}(v)$ have equivalent xPS insertions, by Lemma 2.3.

Regarding (*ii*), note that the first component of the pair obtained from the extended xPS insertion of the x-two-rowed array $\binom{u}{v}$ under Algorithm 3.4 is equal to the insertion of v under Algorithm 2.2. Thus from Remark 2.4 it follows that $\mathrm{Std}_{\mathrm{x}}\big(\mathrm{P}_{\mathrm{xps}}(\binom{u}{v})\big) = \mathrm{P}_{\mathrm{xps}}\Big(\binom{u}{\mathrm{std}_{\mathrm{x}}(v)}\Big) = \mathrm{P}_{\mathrm{xps}}\Big(\binom{\mathrm{std}_{\mathrm{l}}(u)}{\mathrm{std}_{\mathrm{x}}(v)}\Big)$. $\square$

If we relax the definition of standard stable pairs set over an alphabet so that it allows the second component of the pair to be a standard PS tableau over that alphabet instead of just a recording tableau, then the bijection of Proposition 3.2 is given by the mapping

$$w = \binom{u}{v} \longmapsto \big(\mathrm{P}(w), \mathrm{Q}(w)\big),$$

where $\binom{u}{v}$ is a standard two-rowed array, that is, $u = u_1 \cdots u_k$ and v are standard words of the same length such that $u_1 < \cdots < u_k$ and $\big(\mathrm{P}(w), \mathrm{Q}(w)\big)$ is the insertion of w under any of the versions of Algorithm 3.4.

With this new definition of standard stable pairs set, for $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$, let the *semistandard* x*PS stable pairs set over an alphabet* $\mathcal{B}$ be the set composed by the pairs of xPS tableaux (P, Q) of the same shape over $\mathcal{B}$ such that $\big(\mathrm{Std}_{\mathrm{x}}(P), \mathrm{Std}_{\mathrm{x}}(Q)\big)$ is in the stable pairs set over $\mathcal{C}(\mathcal{B})$.

Theorem 3.8. *For* $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$*, there is a one-to-one correspondence between* x*-two-rowed arrays over an alphabet and the pairs of* x*PS tableaux from the semistandard* x*PS stable pairs set over that alphabet. The one-to-one correspondence is given under the mapping*

$$w \mapsto (\mathrm{P}_{\mathrm{xps}}(w), \mathrm{Q}_{\mathrm{xps}}(w)).$$

Proof. Let $\mathcal{B}$ be an alphabet and $\mathrm{x} \in \{\mathrm{l}, \mathrm{r}\}$. Let $\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)$ be a x-two-rowed array over $\mathcal{B}$, and thus $w = \left(\begin{smallmatrix} \mathrm{std}_{\mathrm{l}}(u) \\ \mathrm{std}_{\mathrm{x}}(v) \end{smallmatrix}\right)$ is a standard two-rowed array over $\mathcal{C}(\mathcal{B})$. Therefore, using (the two-rowed version of) Proposition 3.2, $\left(\mathrm{P}_{\mathrm{xps}}\left(\left(\begin{smallmatrix} \mathrm{std}_{\mathrm{l}}(u) \\ \mathrm{std}_{\mathrm{x}}(v) \end{smallmatrix}\right)\right), \mathrm{Q}_{\mathrm{xps}}\left(\left(\begin{smallmatrix} \mathrm{std}_{\mathrm{l}}(u) \\ \mathrm{std}_{\mathrm{x}}(v) \end{smallmatrix}\right)\right)\right)$ is a pair in the standard stable pairs set over $\mathcal{C}(\mathcal{B})$. By Proposition 3.7 (ii), it follows that $\mathrm{P}_{\mathrm{xps}}\left(\left(\begin{smallmatrix} \mathrm{std}_{\mathrm{l}}(u) \\ \mathrm{std}_{\mathrm{x}}(v) \end{smallmatrix}\right)\right) = \mathrm{Std}_{\mathrm{x}}\left(\mathrm{P}_{\mathrm{xps}}\left(\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)\right)\right)$, and by (i) of the same proposition it follows that $\mathrm{Q}_{\mathrm{xps}}\left(\left(\begin{smallmatrix} \mathrm{std}_{\mathrm{l}}(u) \\ \mathrm{std}_{\mathrm{x}}(v) \end{smallmatrix}\right)\right) = \mathrm{Std}_{\mathrm{x}}\left(\mathrm{Q}_{\mathrm{xps}}\left(\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)\right)\right)$. So, we conclude that from the two-rowed array $\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)$ we obtain a pair $\left(\mathrm{P}_{\mathrm{xps}}\left(\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)\right), \mathrm{Q}_{\mathrm{xps}}\left(\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)\right)\right)$ that is in the semistandard xPS stable pairs set over $\mathcal{B}$. Note that different two-rowed arrays lead to different pairs.

The other way around, let (R, S) be a pair of xPS tableaux in the semistandard xPS stable pairs set over $\mathcal{B}$. Then, by definition $\big(\mathrm{Std}_{\mathrm{x}}(R), \mathrm{Std}_{\mathrm{x}}(S)\big)$ is in the standard stable pairs set over $\mathcal{C}(\mathcal{B})$. So, by (the two-rowed version of) Proposition 3.2, there is a standard two-rowed array $\left(\begin{smallmatrix} u' \\ v' \end{smallmatrix}\right)$ over $\mathcal{C}(\mathcal{B})$ that maps to this pair.

The word u' is ordered strictly increasingly and so $u' = \mathrm{std}_{\mathrm{l}}(u)$, for a word u over $\mathcal{B}$, obtained from u' removing the indexes, being u also ordered weakly increasingly.

Also $v' = \mathrm{std}_{\mathrm{x}}(v)$, for some word v over $\mathcal{B}$. Indeed, the first component of the pair obtained from the extended xPS insertion of the x-two-rowed array $\left(\begin{smallmatrix} u' \\ v' \end{smallmatrix}\right)$ under Algorithm 3.4, is equal to the insertion of v' under Algorithm 2.2, so $\mathrm{P}_{\mathrm{xps}}(v') = \mathrm{Std}_{\mathrm{x}}(R)$. Thus, v' has the claimed form, by Lemma 2.5.

Hence, $\left(\mathrm{P}_{\mathrm{xps}}\left(\left(\begin{smallmatrix} \mathrm{std}_{\mathrm{l}}(u) \\ \mathrm{std}_{\mathrm{x}}(v) \end{smallmatrix}\right)\right), \mathrm{Q}_{\mathrm{xps}}\left(\left(\begin{smallmatrix} \mathrm{std}_{\mathrm{l}}(u) \\ \mathrm{std}_{\mathrm{x}}(v) \end{smallmatrix}\right)\right)\right) = \big(\mathrm{Std}_{\mathrm{x}}(R), \mathrm{Std}_{\mathrm{x}}(S)\big)$. To see that $\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)$ is a x-two-rowed array over $\mathcal{B}$, it remains to check that condition (2) holds in case $\mathrm{x} = l$, and condition (3) holds in case $\mathrm{x} = r$. In case $\mathrm{x} = l$, if condition (2) did not hold, then there would exist symbols a_i, a_j from u', with $i < j$ (and thus $a_i < a_j$), such that a_j would be inserted either in the same column of a_i on top of it, or in a column to the left of a_i. This contradicts $\mathrm{Q}_{\mathrm{xps}}\left(\left(\begin{smallmatrix}\mathrm{std_l}(u)\\ \mathrm{std_x}(v)\end{smallmatrix}\right)\right) = \mathrm{Std_x}(S)$. With a similar reasoning we deduce the case $\mathrm{x} = r$.

Finally, by Proposition 3.7, we get $\mathrm{Std_x}\left(\mathrm{P}_{\mathrm{xps}}(w)\right) = \mathrm{Std_x}(R)$ and $\mathrm{Std_x}(\mathrm{Q}_{\mathrm{xps}}(w)) = \mathrm{Std_x}(S)$, for $w = \left(\begin{smallmatrix}u\\ v\end{smallmatrix}\right)$. Thus, destandardizing the xPS tableaux we deduce that the two-rowed array w over $\mathcal{B}$ maps to (R, S). The result follows. $\square$

In the following paragraphs we illustrate Theorem 3.8 using the lPS reverse insertion. Consider the following pair of lPS tableaux

$$(R,S)=\left(\begin{array}{cccccc} & \boxed{2} & & \boxed{3} & & \\ \boxed{1}&\boxed{1}&\boxed{2}&\boxed{2}&\boxed{3}&\boxed{3}\end{array},\ \begin{array}{cccccc} & \boxed{2} & & \boxed{3} & & \\ \boxed{1}&\boxed{1}&\boxed{1}&\boxed{2}&\boxed{4}&\boxed{4}\end{array}\right)$$

in the semistandard lPS stable pairs set over $\mathcal{A}_4$.

By the previous theorem, $(R, S) = \left(\mathrm{P}_{\mathrm{lps}}(\left(\begin{smallmatrix}u\\ v\end{smallmatrix}\right)), \mathrm{Q}_{\mathrm{lps}}(\left(\begin{smallmatrix}u\\ v\end{smallmatrix}\right))\right)$ for some l-two-rowed array $\left(\begin{smallmatrix}u\\ v\end{smallmatrix}\right)$.

Applying left standardization to the pair of tableaux (R, S) we obtain

$$\left(\mathrm{Std_l}(R), \mathrm{Std_l}(S)\right)=\left(\begin{array}{cccccc} & \boxed{2_1} & & \boxed{3_1} & & \\ \boxed{1_1}&\boxed{1_2}&\boxed{2_2}&\boxed{2_3}&\boxed{3_2}&\boxed{3_3}\end{array},\ \begin{array}{cccccc} & \boxed{2_1} & & \boxed{3_1} & & \\ \boxed{1_1}&\boxed{1_2}&\boxed{1_3}&\boxed{2_2}&\boxed{4_1}&\boxed{4_2}\end{array}\right).$$

Then, applying the lPS reverse insertion to this pair, we obtain the l-two-rowed array

$$\begin{pmatrix}\mathrm{std_l}(u)\\ \mathrm{std_l}(v)\end{pmatrix} = \begin{pmatrix}1_1 & 1_2 & 1_3 & 2_1 & 2_2 & 3_1 & 4_1 & 4_2\\ 1_1 & 2_1 & 2_2 & 1_2 & 3_1 & 2_3 & 3_2 & 3_3\end{pmatrix}$$

Applying destandardization to both $\mathrm{std_l}(u)$ and $\mathrm{std_l}(v)$, we obtain

$$\begin{pmatrix}u\\ v\end{pmatrix} = \begin{pmatrix}1 & 1 & 1 & 2 & 2 & 3 & 4 & 4\\ 1 & 2 & 2 & 1 & 3 & 2 & 3 & 3\end{pmatrix},$$

which can be verified to insert to the pair (P, Q) under Algorithm 3.4.

4. Bell numbers and the number of PS tableaux

The n-th Bell number counts the number of different partitions of a set having n distinct elements (sequence A000110 in the OEIS).

Given $n \in \mathbb{N}$, as noticed in [BL07], if we represent set partitions of the totally ordered set $\mathcal{A}_n$ by:

- ordering decreasingly the sets that compose set partitions;
- ordering set partitions increasingly according to the minimum elements of their sets;

we establish a one-to-one correspondence between the partitions of the set $\mathcal{A}_n$ and the PS tableaux over standard words of $\mathfrak{S}(\mathcal{A}_n)$. So, if we denote the number of PS tableaux over standard words of $\mathfrak{S}(\mathcal{A}_n)$ by

$$P(1, \ldots, 1) = |\{\mathrm{P}(\sigma) : \sigma \in \mathfrak{S}(\mathcal{A}_n)\}|,$$

where $(1,\ldots,1)$ is a sequence with n symbols 1 and is equal to the evaluation of any standard word $\sigma \in \mathfrak{S}(\mathcal{A}_n)$, then $P(1,\ldots,1)$ is given by the n-th Bell number, B_n.

The Stirling number of second kind [CG96, Chapter 4], denoted $S(n,k)$, is the number of ways of partitioning n different elements into k distinct sets (k columns, in our setting). So, $S(n,k)$ gives the number of PS tableaux over words of $\mathfrak{S}(\mathcal{A}_n)$ that have exactly k columns. As known, the n-th Bell number is given by the sum of the Stirling numbers $S(n,k)$, where k ranges over the set $\{1,\ldots,n\}$.

The goal in the remainder of this section is to follow a similar approach for the case where words are taken over $\mathcal{A}_n^*$. Our objective is to count both the number of lPS and rPS tableaux over words of $\mathcal{A}_n^*$ with a given fixed evaluation.

4.1. **Counting PS tableaux over $\mathcal{A}_n^*$.** In this subsection we present formulas to count both the number of lPS and rPS tableaux over words of $\mathcal{A}_n$ with a given fixed evaluation $(m_1,\ldots,m_n)$, that we shall denote by $L(m_1,\ldots,m_n)$ and $R(m_1,\ldots,m_n)$, respectively. In particular, we consider the case when each symbol of $\mathcal{A}_n$ occurs exactly once. That is, we consider the case of standard words of $\mathfrak{S}(\mathcal{A}_n)$, where we count $L(1,\ldots,1)$ and $R(1,\ldots,1)$, respectively. These quantities are both equal to $P(1,\ldots,1)$ which, as we noted, is equal to the n-th Bell number, B_n.

Before we proceed, we observe that in general, $L(m_1,m_2,\ldots,m_n) \leq P(1,\ldots,1)$ and $R(m_1,m_2,\ldots,m_n) \leq P(1,\ldots,1)$, for any sequence $(1,\ldots,1)$ with $m_1+m_2+\cdots+m_n$ symbols 1.

Notice also that, the numbers $L(m_1,m_2,\ldots,m_n)$ and $R(m_1,m_2,\ldots,m_n)$ only depend on the non-zero entries of the evaluation sequence $(m_1,\ldots,m_n)$. Indeed, for any $n \in \mathbb{N}$, the number of xPS tableaux with evaluation $(m_1,\ldots,m_{k-1},0,m_{k+1},\ldots,m_{n+1})$ is given by the number of x tableaux with evaluation $(m_1,\ldots,m_{k-1},m_{k+1},\ldots,m_{n+1})$. By convenience we assume that $(m_1,\ldots,m_n) \in \mathbb{N}^n$.

It is easy to see that the length of the bottom row of an rPS tableaux with evaluation $(m_1,m_2,\ldots,m_n)$ is greater or equal than 1 and smaller or equal than n. As for the L case we have the next result.

Lemma 4.1. *The length of the bottom row of an lPS tableaux with evaluation $(m_1,m_2,\ldots,m_n)$ is greater or equal than $\max\{m_1,m_2,\ldots,m_n\}$ and less or equal than $m_1+m_2+\cdots+m_n$.*

Proof. Let R be an lPS tableau with evaluation $(m_1,m_2,\ldots,m_n)$. The m_1 symbols 1 of R will all appear in the bottom row. Also, as any column of R can contain no more than one occurrence of each symbol of $\mathcal{A}_n$, the number of columns of R is greater or equal than $\max\{m_1,\ldots,m_n\}$. The lPS insertion of the weakly increasing word with evaluation $(m_1,\ldots,m_n)$ produces a tableau with $m_1+\cdots+m_n$ columns. Thus any lPS tableau with evaluation $(m_1,\ldots,m_n)$ has at most $m_1+\cdots+m_n$ columns. $\square$

Throughout the remainder of this text, we also assume the convention that for $k > m$ the binomial coefficient $\binom{m}{k}$ is 0.

The idea is to start by presenting formulas to count both the number of lPS tableaux and the number of rPS tableaux with a given fixed evaluation and a fixed bottom row. So, given sequences $(m_1, m_2, \ldots, m_n) \in \mathbb{N}^n$ and $(j_2, j_3, \ldots, j_n) \in \mathbb{N}_0^{n-1}$, let

$$\begin{bmatrix} m_1 & m_2 & \cdots & m_n \\ 0 & j_2 & \cdots & j_n \end{bmatrix}^{\mathrm{lps}} = \binom{m_1}{m_2 - j_2} \cdots \binom{m_1 + j_2 + \cdots + j_{n-1}}{m_n - j_n}$$

Lemma 4.2. *The number of lPS tableaux having evaluation* $(m_1, \ldots, m_n) \in \mathbb{N}^n$ *whose lPS bottom row has evaluation* $(m_1, j_2, \ldots, j_n) \in \mathbb{N} \times \mathbb{N}_0^{n-1}$*, is given by the formula*

$$\begin{bmatrix} m_1 & m_2 & \cdots & m_n \\ 0 & j_2 & \cdots & j_n \end{bmatrix}^{\mathrm{lps}}.$$

Proof. These tableaux have j_2 symbols 2 in the bottom row and since any lPS column cannot contain more than one symbol from $\mathcal{A}_n$, by the order of lPS columns and lPS bottom rows, the remaining $m_2 - j_2$ symbols 2 are in the columns (in the second row) that contain the symbol 1. Thus, $m_2 - j_2$ is less or equal than m_1 and there are $\binom{m_1}{m_2 - j_2}$ possibilities to place those 2's.

More generally, such tableaux will have j_a symbols a in the bottom row, and so the remaining $m_a - j_a$ symbols a cannot be inserted in columns to the right of the columns containing symbols a. Therefore, they must be placed in columns that have symbols from $\mathcal{A}_{a-1}$. There are $m_1 + j_2 + \cdots + j_{a-1}$ such columns. Thus $m_a - j_a$ is less or equal than $m_1 + j_2 + \cdots + j_{a-1}$ and there are $\binom{m_1 + j_2 + \cdots + j_{a-1}}{m_a - j_a}$ possibilities to place those symbols. The result follows. ☐

From the previous proof we deduce that for lPS tableaux with evaluation $(m_1, \ldots, m_n)$ whose bottom row has evaluation $(m_1, j_2, \ldots, j_n)$, for any $a \in \mathcal{A}_n \setminus \{1\}$, we have $0 \leq m_a - j_a \leq m_1 + \sum_{b=2}^{a-1} j_b$.

Regarding the rPS case, for any sequences $(m_1, m_2, \ldots, m_n) \in \mathbb{N}^n$ and $(j_1, j_2, \ldots, j_n) \in \{0, 1\}^n$, let

$$\begin{bmatrix} & m_2 & \cdots & m_n \\ j_1 & j_2 & \cdots & j_n \end{bmatrix}^{\mathrm{rps}} = \binom{m_2 + j_1}{m_2 - j_2} \cdots \binom{m_n + (j_1 + j_2 + \cdots + j_{n-1})}{m_n - j_n}.$$

Lemma 4.3. *The number of rPS tableaux with evaluation* $(m_1, \ldots, m_n) \in \mathbb{N}^n$ *and whose rPS bottom row has evaluation the* 0-1 *sequence* $(1, j_2, \ldots, j_n)$*, is given by the formula*

$$\begin{bmatrix} & m_2 & \cdots & m_n \\ 0 & j_2 & \cdots & j_n \end{bmatrix}^{\mathrm{rps}}.$$

Proof. First note that the number of ways to distribute k symbols into l columns can be calculated using the "stars and bars" method. That is, the k symbols can be viewed as k stars which we want to separate into l groups by using $l-1$ bars in between. So, this number is given by the number of ways of choosing $l-1$ positions from $k+l-1$ spaces, that is, $\binom{k+l-1}{l-1} = \binom{k+l-1}{k}$.

All the symbols 1 occur in the bottom part of the first column of these tableaux. Moreover, the first column is followed by $j_2 \in \{0,1\}$ columns whose bottom symbol is 2. Considering the order of rPS columns and rPS bottom rows, the remaining $m_2 - j_2$ symbols 2 have to be distributed over their first $j_1 + j_2$ columns. By the first paragraph there are

$$\binom{m_2 - j_2 + j_1 + j_2 - 1}{m_2 - j_2} = \binom{m_2}{m_2 - j_2}$$

such possibilities.

In general, these tableaux have $j_a \in \{0,1\}$ columns whose bottom symbol is a and thus the remaining $m_a - j_a$ symbols have to be inserted over the first $j_1 + j_2 + \cdots + j_a$ columns. By the first paragraph, there are

$$\binom{m_a - j_a + (j_1 + j_2 + \cdots + j_a) - 1}{m_a - j_a} = \binom{m_a + (j_2 + \cdots + j_{a-1})}{m_a - j_a}$$

possibilities to place those symbols. The result follows. $\square$

Lemma 4.4. *The number $R(m_1, \ldots, m_n)$ of rPS tableaux with evaluation $(m_1, \ldots, m_n) \in \mathbb{N}^n$ is independent of the choice of m_1. Also, both the numbers $L(m)$ and $R(m)$ of lPS and rPS tableaux with evaluation (m) is equal to 1.*

Proof. The first part of the proposition follows from the fact that all the m_1 symbols 1 are in the bottom part of the first column and therefore $j_1 = 1$ for any $m_1 \in \mathbb{N}$. The second part of the proposition is immediate. $\square$

Proposition 4.5. *The numbers $L(m_1, \ldots, m_n)$ and $R(m_1, \ldots, m_n)$ of, respectively, lPS tableaux and rPS tableaux with evaluation $(m_1, \ldots, m_n)$, are given by*

$$L(m_1, \ldots, m_n) = \sum_{(0,\ldots,0) \leq (j_2,\ldots,j_n) \leq (m_2,\ldots,m_n)} \begin{bmatrix} m_1 & m_2 & \cdots & m_n \\ 0 & j_2 & \cdots & j_n \end{bmatrix}^{\mathrm{lps}}$$

and

$$R(m_1, \ldots, m_n) = \sum_{(0,\ldots,0) \leq (j_2,\ldots,j_n) \leq (1,\ldots,1)} \begin{bmatrix} & m_2 & \cdots & m_n \\ 0 & j_2 & \cdots & j_n \end{bmatrix}^{\mathrm{rps}}.$$

Proposition 4.6. *The numbers $L(m_1, \ldots, m_n)$ and $R(m_1, \ldots, m_n)$ of, respectively, lPS and rPS tableaux with evaluation $(m_1, \ldots, m_n)$ are recursively obtained in the following way*

$$L(m_1, m_2, \ldots, m_n) = \sum_{0 \leq j_2 \leq m_2} \binom{m_1}{m_2 - j_2} L(m_1 + j_2, m_3, \ldots, m_n).$$

and

$$
\begin{aligned}
R(m_1, m_2, \ldots, m_n) = R(m_2, m_3, \ldots, m_n) + \\
+ \sum_{(0,\ldots,0)\le(j_3,\ldots,j_n)\le(1,\ldots,1)} m_2 \begin{bmatrix} & m_3 & \cdots & m_n \\ 1 & j_3 & \cdots & j_n \end{bmatrix}^{\mathrm{rps}},
\end{aligned}
$$

where $L(m) = 1 = R(m)$ *and* $R(m, n) = \binom{n}{n} + \binom{n}{n-1} = 1 + n$, *for all* $m, n \in \mathbb{N}$.

Proof. Note that

$$
\begin{bmatrix} m_1 & m_2 & \cdots & m_n \\ 0 & j_2 & \cdots & j_n \end{bmatrix}^{\mathrm{lps}} = \binom{m_1}{m_2 - j_2} \begin{bmatrix} m_1 + j_2 & m_3 & \cdots & m_n \\ 0 & j_3 & \cdots & j_n \end{bmatrix}^{\mathrm{lps}}
$$

and for $j \in \mathbb{N}_0$

$$
\begin{bmatrix} & m_2 & \cdots & m_n \\ j & j_2 & \cdots & j_n \end{bmatrix}^{\mathrm{rps}} = \binom{m_2 + j}{m_2 - j_2} \begin{bmatrix} & m_3 & \cdots & m_n \\ j + j_2 & j_3 & \cdots & j_n \end{bmatrix}^{\mathrm{rps}}.
$$

The result now follows from Lemma 4.4 and Proposition 4.5, observing that from Proposition 4.5 we have

$$
R(m_1, m_2) = \sum_{0 \le j_2 \le 1} \begin{bmatrix} & m_2 \\ 0 & j_2 \end{bmatrix}^{\mathrm{rps}} = \binom{m_2}{m_2} + \binom{m_2}{m_2 - 1} = 1 + m_2.
$$

∎

As an example, we use the above formula to compute the number of lPS tableaux with evaluation $(2, 1, 2)$

$$
\begin{aligned}
L(2,1,2) &= \sum_{j_2=0}^{1} \binom{2}{1-j_2} L(2+j_2, 2) = \binom{2}{1} \cdot L(2,2) + \binom{2}{0} \cdot L(3,2) \\
&= 2 \cdot \left(\binom{2}{2} + \binom{2}{1} + \binom{2}{0} \right) + 1 \cdot \left(\binom{3}{2} + \binom{3}{1} + \binom{3}{0} \right) \\
&= 7 + 8 = 15
\end{aligned}
$$

Indeed, $\binom{2}{0} L(3,2)$ is obtained when $j_2 = 1$ and therefore is the number of such lPS tableaux with the symbol 2 on the bottom row:

$$
\begin{aligned}
j_3 = 2 \;&\to\; \binom{3}{0} \;\to\; \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 2 & 3 & 3 \\ \hline\end{array} \\
j_3 = 1 \;&\to\; \binom{3}{1} \;\to\; \begin{array}{cccc} \boxed{3} & & & \\ \boxed{1} & \boxed{1} & \boxed{2} & \boxed{3} \end{array},\; \begin{array}{cccc} & \boxed{3} & & \\ \boxed{1} & \boxed{1} & \boxed{2} & \boxed{3} \end{array},\; \begin{array}{cccc} & & \boxed{3} & \\ \boxed{1} & \boxed{1} & \boxed{2} & \boxed{3} \end{array} \\
j_3 = 0 \;&\to\; \binom{3}{2} \;\to\; \begin{array}{ccc} \boxed{3} & \boxed{3} & \\ \boxed{1} & \boxed{1} & \boxed{2} \end{array},\; \begin{array}{ccc} \boxed{3} & & \boxed{3} \\ \boxed{1} & \boxed{1} & \boxed{2} \end{array},\; \begin{array}{ccc} & \boxed{3} & \boxed{3} \\ \boxed{1} & \boxed{1} & \boxed{2} \end{array}
\end{aligned}
$$

The number $\binom{2}{1}L(3,2)$ counts those lPS tableaux where the symbol 2 is on top of the symbols 1 on the second row:

$$j_3 = 2 \;\rightarrow\; \binom{2}{1}\cdot\binom{2}{0} \;\rightarrow\; \begin{array}{cccc} 2 & & & \\ 1 & 1 & 3 & 3 \end{array},\; \begin{array}{cccc} & 2 & & \\ 1 & 1 & 3 & 3 \end{array}$$

$$j_3 = 1 \;\rightarrow\; \binom{2}{1}\cdot\binom{2}{1} \;\rightarrow\; \begin{array}{ccc} 3 & & \\ 2 & & \\ 1 & 1 & 3 \end{array},\; \begin{array}{ccc} 2 & 3 & \\ 1 & 1 & 3 \end{array},\; \begin{array}{ccc} 3 & 2 & \\ 1 & 1 & 3 \end{array},\; \begin{array}{ccc} & 3 & \\ & 2 & \\ 1 & 1 & 3 \end{array}$$

$$j_3 = 0 \;\rightarrow\; \binom{2}{1}\cdot\binom{2}{2} \;\rightarrow\; \begin{array}{cc} 3 & \\ 2 & 3 \\ 1 & 1 \end{array},\; \begin{array}{cc} & 3 \\ 3 & 2 \\ 1 & 1 \end{array}$$

Analogously, the number of rPS tableaux with evaluation $(2,1,2)$ is given by:

$$\begin{aligned}
R(2,1,2) &= R(1,2) + \sum_{j_3=0}^{1} 1 \begin{bmatrix} & 2 \\ 1 & j_3 \end{bmatrix}^{\mathrm{rps}} \\
&= \sum_{j_3=0}^{1} \binom{2}{2-j_3} + \left(\begin{bmatrix} & 2 \\ 1 & 0 \end{bmatrix}^{\mathrm{rps}} + \begin{bmatrix} & 2 \\ 1 & 1 \end{bmatrix}^{\mathrm{rps}} \right) \\
&= \left(\binom{2}{2} + \binom{2}{1} \right) + \left(\binom{3}{2} + \binom{3}{1} \right) = 3 + 6 = 9
\end{aligned}$$

The number $R(1,2)$ counts the number of those rPS tableaux when $j_2 = 0$, that is, whenever the symbol 2 is on top of the symbols 1 in the first column:

$$j_3 = 0 \;\rightarrow\; \binom{2}{2} \;\rightarrow\; \begin{array}{c} 3 \\ 3 \\ 2 \\ 1 \\ 1 \end{array}$$

$$j_3 = 1 \;\rightarrow\; \binom{2}{1} \;\rightarrow\; \begin{array}{cc} 3 & \\ 2 & \\ 1 & \\ 1 & 3 \end{array},\; \begin{array}{cc} 2 & \\ 1 & 3 \\ 1 & 3 \end{array}$$

The number $\sum_{j_3=0}^{1} 1 \begin{bmatrix} & 2 \\ 1 & j_3 \end{bmatrix}^{\mathrm{rps}}$ counts those rPS tableaux whenever $j_2 = 1$, that is, it counts the rPS tableaux where 2 is in the bottom row:

$$j_3 = 0 \;\rightarrow\; \binom{3}{2} \;\rightarrow\; \begin{array}{cc} 3 & \\ 3 & \\ 1 & \\ 1 & 2 \end{array},\; \begin{array}{cc} 3 & \\ 1 & 3 \\ 1 & 2 \end{array},\; \begin{array}{cc} & 3 \\ 1 & 3 \\ 1 & 2 \end{array}$$

$$j_3 = 1 \;\rightarrow\; \binom{3}{1} \;\rightarrow\; \begin{array}{ccc} 3 & & \\ 1 & & \\ 1 & 2 & 3 \end{array},\; \begin{array}{ccc} 1 & 3 & \\ 1 & 2 & 3 \end{array},\; \begin{array}{ccc} 1 & & 3 \\ 1 & 2 & 3 \end{array}$$

Applying Proposition 4.5 to the standard case, we get:

Corollary 4.7. *If* $(1,\ldots,1)$ *is a sequence with* n *symbols* 1*, the* n*-th Bell number,* B_n*, is given by both* $L(1,\ldots,1)$ *and* $R(1,\ldots,1)$*, which are equal to*

$$\sum_{0\leq p_2,\ldots,p_n\leq 1} (1+p_2)^{(1-p_3)}\cdots(1+p_2+\cdots+p_{n-1})^{(1-p_n)}.$$

Proof. Both the numbers $\begin{bmatrix} m_1 & m_2 & \cdots & m_n \\ 0 & j_2 & \cdots & j_n \end{bmatrix}^{\mathrm{lps}}$ and $\begin{bmatrix} & m_2 & \cdots & m_n \\ 0 & j_2 & \cdots & j_n \end{bmatrix}^{\mathrm{rps}}$, with $m_1 = \ldots = m_n = 1$ are given by

$$\binom{1}{1-j_2}\binom{1+j_2}{1-j_3}\binom{1+j_2+j_3}{1-j_4}\cdots\binom{1+j_2+\cdots+j_{n-1}}{1-j_n}.$$

Since each j_a is either 0 or 1, then $1-j_a$ is either 1 or 0 and thus $\binom{1+j_2+\cdots+j_a}{1-j_{a+1}} = (1+j_2+\cdots+j_a)^{(1-j_{a+1})}$, for $a \in \mathcal{A}_{n-1}\setminus\{1\}$. The result follows from Proposition 4.5. $\square$

For example, $B_4 = \sum_{0\leq p_2,p_3,p_4\leq 1} (1+p_2)^{(1-p_3)}(1+p_2+p_3)^{(1-p_4)}$. The possible triples of 0's and 1's are $(0,0,0)$, $(1,0,0)$, $(0,1,0)$, $(0,0,1)$, $(0,1,1)$, $(1,0,1)$, $(1,1,0)$ and $(1,1,1)$, which maintaining the order gives the sum $1+2\times 2+2+1+1+2+3+1 = 15$.

5. A hook length formula for standard PS tableaux

For any partition λ of a natural number, the hook-length formula is a formula that gives the number of standard Young tableaux having shape λ. Besides that, the hook-length formula also provides the dimension of the irreducible representation of the symmetric group associated to λ.

By the bijectivity of the Robinson correspondence, it follows that the hook-length formula provides the number of standard words that insert to a specific standard Young tableaux under Schensted's insertion algorithm.

In this section, our goal is to provide an analogous hook-length formula for standard Patience Sorting tableaux. As shapes of PS tableaux are compositions of natural numbers, we will work with compositions instead of partitions. As a consequence, we deduce a new formula for the Bell numbers and bounds for both factorial numbers and the number of words inserting to a specific standard Patience Sorting tableaux.

Consider the following algorithm:

Algorithm 5.1 (From pre-tableaux to standard PS tableaux)**.**

Input: A pre-tableau T from $\mathrm{Tab}_{\mathcal{B}}(\lambda)$, with $\lambda = (\lambda_1,\lambda_2,\ldots,\lambda_m)$ and $\mathcal{B} \subseteq \mathcal{A}_n$, for some $n \geq m$.

Output: A standard PS tableau $W(T) \in \mathrm{PSTab}_{\mathcal{B}}(\lambda)$.

Method:

Let $T = c_1 c_2 \cdots c_m$, for pre-columns $c_1, \ldots, c_m$ of height $\lambda_1, \ldots, \lambda_m$, respectively.

(1) Step 1: Let (j, k) denote the column-row position of the smallest symbol from T. In T, exchange the symbol in the column-row position (j, k) with the symbol in the column-row position $(1, 1)$. Then rearrange the symbols of the obtained first column in increasing order from bottom to top. Denote the obtained tableau by $c_1^{(1)} c_2^{(1)} \cdots c_m^{(1)}$;

(2) Step i $(2 \leq i \leq m)$: Let (j, k) denote the column-row position of the smallest symbol from $c_i^{(i-1)} \cdots c_m^{(i-1)}$. In $c_i^{(i-1)} \cdots c_m^{(i-1)}$, exchange the symbol in the column-row position (j, k) with the symbol in the column-row position $(i, 1)$. Then rearrange the symbols of the obtained i-th column in increasing order from bottom to top. Denote the obtained tableau by $c_i^{(i)} \cdots c_m^{(i)}$.

Output the tableau $W(T) = c_1^{(1)} c_2^{(2)} \cdots c_m^{(m)}$.

Observe that, if $\mathcal{B} \subseteq \mathcal{A}_n$ for some $n \in \mathbb{N}$ and $\lambda \vDash |\mathcal{B}|$, then applying the algorithm to an arbitrary tableau of $\mathrm{Tab}_{\mathcal{B}}(\lambda)$ always yields a tableau in $\mathrm{PSTab}_{\mathcal{B}}(\lambda)$. Indeed, if $T = c_1 c_2 \cdots c_m \in \mathrm{Tab}_{\mathcal{B}}(\lambda)$ and for any $i \in \{1, 2, \ldots, m\}$, x_i denotes the bottom symbol of the column $c_i^{(i)}$ of the tableau $W(T) = c_1^{(1)} c_2^{(2)} \cdots c_m^{(m)}$, then as the symbols from $W(T)$ are all different and x_i is the smallest symbol from $c_i^{(i-1)} \cdots c_m^{(i-1)}$, we deduce that for all $i \in \{1, 2, \ldots, m-1\}$, $x_i < x_{i+1}$. Also, as for any $i \in \{1, 2, \ldots, m\}$, $c_i^{(i)}$ is in increasing order from bottom to top, we conclude that the obtained tableau $W(T) = c_1^{(1)} c_2^{(2)} \cdots c_m^{(m)}$ belongs to $\mathrm{PSTab}_{\mathcal{B}}(\lambda)$. Furthermore, for any tableau $T \in \mathrm{PSTab}_{\mathcal{B}}(\lambda)$, we have $W(T) = T$.

For example, considering $\mathcal{B} = \{1, 2, 4, 5, 6, 7, 8, 9\} \subseteq \mathcal{A}_9$, the steps that we obtain from applying the previous algorithm to the pre-tableau

$$T = c_1 c_2 c_3 c_4 = \begin{array}{cccc} & \boxed{4} & & \\ & \boxed{5} & \boxed{1} & \boxed{7} \\ \boxed{9} & \boxed{8} & \boxed{6} & \boxed{2} \end{array} \in \mathrm{Tab}_{\mathcal{B}}\big((1,3,2,2)\big)$$

are

$$\text{Step 1: } T = \begin{array}{cccc} & \boxed{4} & & \\ & \boxed{5} & \boxed{1} & \boxed{7} \\ \boxed{9} & \boxed{8} & \boxed{6} & \boxed{2} \end{array}, \quad c_1^{(1)} c_2^{(1)} c_3^{(1)} c_4^{(1)} = \begin{array}{cccc} & \boxed{4} & & \\ & \boxed{5} & \boxed{9} & \boxed{7} \\ \boxed{1} & \boxed{8} & \boxed{6} & \boxed{2} \end{array}$$

$$\text{Step 2: } c_2^{(1)} c_3^{(1)} c_4^{(1)} = \begin{array}{ccc} \boxed{4} & & \\ \boxed{5} & \boxed{9} & \boxed{7} \\ \boxed{8} & \boxed{6} & \boxed{2} \end{array}, \quad c_2^{(2)} c_3^{(2)} c_4^{(2)} = \begin{array}{ccc} \boxed{5} & & \\ \boxed{4} & \boxed{9} & \boxed{7} \\ \boxed{2} & \boxed{6} & \boxed{8} \end{array}$$

$$\text{Step 3: } c_3^{(2)} c_4^{(2)} = \begin{array}{cc} \boxed{9} & \boxed{7} \\ \boxed{6} & \boxed{8} \end{array}, \quad c_3^{(3)} c_4^{(3)} = \begin{array}{cc} \boxed{9} & \boxed{7} \\ \boxed{6} & \boxed{8} \end{array}$$

$$\text{Step 4: } c_4^{(3)} = \begin{array}{c} \boxed{7} \\ \boxed{8} \end{array}, \quad c_4^{(4)} = \begin{array}{c} \boxed{8} \\ \boxed{7} \end{array}$$

and the algorithm outputs

$$W(T) = c_1^{(1)} c_2^{(2)} c_3^{(3)} c_4^{(4)} = \begin{array}{cccc} & \boxed{5} & & \\ & \boxed{4} & \boxed{9} & \boxed{8} \\ \boxed{1} & \boxed{2} & \boxed{6} & \boxed{7} \end{array} \in \mathrm{PSTab}_{\mathcal{B}}\left((1,3,2,2)\right).$$

These observations allow us to conclude that, for any $n \in \mathbb{N}$, $\mathcal{B} \subseteq \mathcal{A}_n$ and $\lambda \vDash |\mathcal{B}|$ the map

$$\begin{aligned} w_{\lambda,\mathcal{B}} : \mathrm{Tab}_{\mathcal{B}}(\lambda) &\to \mathrm{PSTab}_{\mathcal{B}}(\lambda) \\ T &\mapsto W(T) \end{aligned}$$

is well defined. From the fact that $W(W(T)) = W(T)$ it follows that $w_{\lambda,\mathcal{B}}$ is surjective.

Theorem 5.2. *If $\mathcal{B} \subseteq \mathcal{A}_n$, for some $n \in \mathbb{N}$, $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m) \vDash |\mathcal{B}|$, and $T \in \mathrm{PSTab}_{\mathcal{B}}(\lambda)$, then the number of tableaux in $\mathrm{Tab}_{\mathcal{B}}(\lambda)$ that map to T under $w_{\lambda,\mathcal{B}}$, $|w_{\lambda,\mathcal{B}}^{-1}(T)|$, is given by the following formula*

$$|w_{\lambda,\mathcal{B}}^{-1}(T)| = \prod_{i=0}^{m-1} \left(|\mathcal{B}| - \sum_{j=1}^{i} \lambda_j \right) \cdot \prod_{k=1}^{m} (\lambda_k - 1)!$$

Proof. The proof follows by induction on the number of columns of $T \in \mathrm{PSTab}_{\mathcal{B}}(\lambda)$, for $\lambda = (\lambda_1, \dots, \lambda_m)$.

Case $m = 1$. Considering the previous algorithm, any disposition of symbols from $\mathcal{B}$ in a column of length $\lambda_1 = |\mathcal{B}|$ will lead to the column tableau obtained by arranging these symbols in increasing order from bottom to top. There are $|\mathcal{B}|!$ possibilities for tableaux in these circumstances and

$$\begin{aligned} \prod_{i=0}^{0} \left(|\mathcal{B}| - \sum_{j=1}^{0} \lambda_j \right) \cdot \prod_{k=1}^{1} (\lambda_k - 1)! &= |\mathcal{B}| \cdot (\lambda_1 - 1)! \\ &= |\mathcal{B}| \cdot (|\mathcal{B}| - 1)! = |\mathcal{B}|! \end{aligned}$$

Fix $m > 1$ and suppose by induction hypothesis that the result is true for $m-1$. That is, suppose that for all $\mathcal{C} \subseteq \mathcal{A}$, $\lambda' = (\lambda'_1, \dots, \lambda'_{m-1})$ with $\lambda' \vDash |\mathcal{C}|$ and $T' = c'_1 c'_2 \cdots c'_{m-1} \in \mathrm{PSTab}_{\mathcal{C}}(\lambda')$,

$$|w_{\lambda',\mathcal{C}}^{-1}(T')| = \prod_{i=0}^{m-2} \left(|\mathcal{C}| - \sum_{j=1}^{i} \lambda'_j \right) \cdot \prod_{k=1}^{m-1} (\lambda'_k - 1)!$$

Consider $\mathcal{B} \subseteq \mathcal{A}_n$ for some $n \in \mathbb{N}$ and $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m) \in \mathbb{N}^m$ such that $\lambda \vDash |\mathcal{B}|$. Given a PS tableau $T = c_1 c_2 \cdots c_m \in \mathrm{PSTab}_{\mathcal{B}}(\lambda)$, considering Algorithm 5.1, the tableaux that map to T under $w_{\lambda,\mathcal{B}}$ are the tableaux $\bar{T} = \bar{c_1}\bar{c_2} \cdots \bar{c_m} \in \mathrm{Tab}_{\mathcal{B}}(\lambda)$ that contain the smallest symbol from $\mathcal{B}$ in any column-row position and such that, if $\bar{T}' = \bar{c_1}'\bar{c_2}' \cdots \bar{c_m}' \in \mathrm{Tab}_{\mathcal{B}}(\lambda)$ is the tableau obtained from exchanging the positions of the smallest symbol of $\bar{T}$ with the symbol in the column-row position $(1,1)$ of $\bar{T}$ then $\mathrm{cont}(\bar{c_1}') = \mathrm{cont}(c_1)$ and $w_{(\lambda_2,\dots,\lambda_m),\mathrm{cont}(\bar{c_2}' \cdots \bar{c_m}')}(\bar{c_2}' \cdots \bar{c_m}') =$

$c_2 \cdots c_m$. Since there are $|\mathcal{B}|$ column-row positions in $\bar{T}$ for the smallest symbol, and there are $\lambda_1 - 1$ positions in $\bar{T}'$ to be filled with $|\mathrm{cont}(c_1)| - 1 = \lambda_1 - 1$ symbols (the smallest symbol from $\mathcal{B}$ is already in the bottom left box of $\bar{T}'$), there are

$$|\mathcal{B}| \cdot (\lambda_1 - 1)! \cdot |w^{-1}_{(\lambda_2,\ldots,\lambda_m),\mathrm{cont}(c_2\cdots c_m)}(c_2 \cdots c_m)|$$

tableaux that will map to T under $w_{\lambda,\mathcal{B}}$. By the induction hypothesis,

$$\begin{aligned}
&|w^{-1}_{(\lambda_2,\ldots,\lambda_m),\mathrm{cont}(c_2\cdots c_m)}(c_2 \cdots c_m)| \\
&= \prod_{i=1}^{m-1} \left((\lambda_2 + \cdots + \lambda_m) - \sum_{j=2}^{i} \lambda_j \right) \cdot \prod_{k=2}^{m} (\lambda_k - 1)! \\
&= \prod_{i=1}^{m-1} \left((|\mathcal{B}| - \lambda_1) - \sum_{j=2}^{i} \lambda_j \right) \cdot \prod_{k=2}^{m} (\lambda_k - 1)!\ .
\end{aligned}$$

So,

$$\begin{aligned}
|w^{-1}_{\lambda,\mathcal{B}}(T)| &= |\mathcal{B}| \cdot (\lambda_1 - 1)! \cdot \prod_{i=1}^{m-1} \left((|\mathcal{B}| - \lambda_1) - \sum_{j=2}^{i} \lambda_j \right) \cdot \prod_{k=2}^{m} (\lambda_k - 1)! \\
&= \prod_{i=0}^{m-1} \left(|\mathcal{B}| - \sum_{j=1}^{i} \lambda_j \right) \cdot \prod_{k=1}^{m} (\lambda_k - 1)!\ .
\end{aligned}$$

The result follows by induction. ☐

Corollary 5.3. *If $\mathcal{B} \subseteq \mathcal{A}_n$ for some $n \in \mathbb{N}$, and $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m) \vDash |\mathcal{B}|$,*

$$|\mathrm{PSTab}_{\mathcal{B}}(\lambda)| = \frac{(|\mathcal{B}| - 1)!}{\prod_{i=1}^{m-1} \left(|\mathcal{B}| - \sum_{j=1}^{i} \lambda_j \right) \cdot \prod_{k=1}^{m} (\lambda_k - 1)!}$$

Proof. Given $\mathcal{B} \subseteq \mathcal{A}_n$ for some $n \in \mathbb{N}$ and $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m) \vDash |\mathcal{B}|$,

$$|\mathrm{Tab}_{\mathcal{B}}(\lambda)| = |\mathcal{B}|!$$

This number is also given by the sum of the cardinality of the preimages of the tableaux $T \in \mathrm{PSTab}_{\mathcal{B}}(\lambda)$ under $w_{\lambda,\mathcal{B}}$, that is,

$$|\mathrm{Tab}_{\mathcal{B}}(\lambda)| = \sum_{T \in \mathrm{PSTab}_{\mathcal{B}}(\lambda)} |w^{-1}_{\lambda,\mathcal{B}}(T)|$$

By the previous theorem, the number $|w^{-1}_{\lambda,\mathcal{B}}(T)|$ depends only on $\mathcal{B}$ and λ and not on the tableau T. Thus, using the previous theorem it follows that

$$|\mathcal{B}|! = |\mathrm{PSTab}_{\mathcal{B}}(\lambda)| \cdot \prod_{i=0}^{m-1} \left(|\mathcal{B}| - \sum_{j=1}^{i} \lambda_j \right) \cdot \prod_{k=1}^{m} (\lambda_k - 1)!$$

So,

$$\begin{aligned}|\mathrm{PSTab}_{\mathcal{B}}(\lambda)| &= \frac{|\mathcal{B}|!}{\prod_{i=0}^{m-1}\left(|\mathcal{B}| - \sum_{j=1}^{i}\lambda_j\right)\cdot\prod_{k=1}^{m}(\lambda_k-1)!}\\ &= \frac{(|\mathcal{B}|-1)!}{\prod_{i=1}^{m-1}\left(|\mathcal{B}| - \sum_{j=1}^{i}\lambda_j\right)\cdot\prod_{k=1}^{m}(\lambda_k-1)!}.\end{aligned}$$

and the result follows. □

The n-th Bell number B_n is the number of partitions of a set with n distinct elements. As noted at the beginning of Section 4, the partitions of a set $\mathcal{B} \subseteq \mathcal{A}_k$ for some $k \in \mathbb{N}$ are in one-to-one correspondence with the standard PS tableaux from the set $\mathrm{PSTab}_{\mathcal{B}}$. Thus, since

$$\mathrm{PSTab}_{\mathcal{B}} = \bigcup_{\lambda \vDash |\mathcal{B}|} \mathrm{PSTab}_{\mathcal{B}}(\lambda),$$

if $n = |\mathcal{B}|$, the n-th Bell number is also given by

$$|\mathrm{PSTab}_{\mathcal{B}}| = \sum_{\lambda \vDash n} |\mathrm{PSTab}_{\mathcal{B}}(\lambda)|$$

and therefore

Theorem 5.4. *For any $n \in \mathbb{N}$, the n-th Bell number, B_n, is given by the following formula*

$$B_n = \sum_{(\lambda_1,\dots,\lambda_m)\vDash n}\left(\frac{(n-1)!}{\prod_{i=1}^{m-1}\left(n - \sum_{j=1}^{i}\lambda_j\right)\cdot\prod_{k=1}^{m}(\lambda_k-1)!}\right).$$

For example, considering $\mathcal{B} = \{1,2,3,4\}$

$$\begin{aligned}B_4 &= |\mathrm{PSTab}_{\mathcal{B}}\left((4)\right)| + |\mathrm{PSTab}_{\mathcal{B}}\left((3,1)\right)| + |\mathrm{PSTab}_{\mathcal{B}}\left((1,3)\right)|\\ &\quad + |\mathrm{PSTab}_{\mathcal{B}}\left((2,2)\right)| + |\mathrm{PSTab}_{\mathcal{B}}\left((2,1,1)\right)| + |\mathrm{PSTab}_{\mathcal{B}}\left((1,2,1)\right)|\\ &\quad + |\mathrm{PSTab}_{\mathcal{B}}\left((1,1,2)\right)| + |\mathrm{PSTab}_{\mathcal{B}}\left((1,1,1,1)\right)|\\ &= 1+3+1+3+3+2+1+1 = 15.\end{aligned}$$

The algorithm of extended PS insertion from Section 3 (Algorithm 3.1) only defines an injection between the set of standard words $\mathfrak{S}(\mathcal{B})$, where $\mathcal{B} \subseteq \mathcal{A}_n$ for some $n \in \mathbb{N}$, and the set of pairs of standard PS tableaux of the same shape over $\mathcal{B}$, $\bigcup_{\lambda\vDash|\mathcal{B}|}\left(\mathrm{PSTab}_{\mathcal{B}}(\lambda) \times \mathrm{PSTab}_{\mathcal{B}}(\lambda)\right)$. Therefore,

Proposition 5.5. *For any $n \in \mathbb{N}$,*

$$n! \leq \sum_{(\lambda_1,\dots,\lambda_m)\vDash n}\left(\frac{(n-1)!}{\prod_{i=1}^{m-1}\left(n - \sum_{j=1}^{i}\lambda_j\right)\cdot\prod_{k=1}^{m}(\lambda_k-1)!}\right)^2.$$

Proof. Follows from the observations before the proposition together with the fact that for any $\mathcal{B} \subseteq \mathcal{A}_k$ for some $k \in \mathbb{N}$, with $|\mathcal{B}| = n$, $|\mathfrak{S}(\mathcal{B})| = n!$ and Corollary 5.3. □

Proposition 5.6. *For any $k, n \in \mathbb{N}$ and $\mathcal{B} \subseteq \mathcal{A}_k$ such that $|\mathcal{B}| = n$, if $\lambda \vDash n$ and $T \in \mathrm{PSTab}_{\mathcal{B}}(\lambda)$, then*

$$\left|\left\{\sigma \in \mathfrak{S}(\mathcal{B}) : \mathrm{P}(\sigma) = T\right\}\right| \leq \frac{(n-1)!}{\prod_{i=1}^{m-1}\left(n - \sum_{j=1}^{i} \lambda_j\right) \cdot \prod_{k=1}^{m} (\lambda_k - 1)!}.$$

Proof. It is straightforward that

$$\left|\left\{\sigma \in \mathfrak{S}(\mathcal{B}) : \mathrm{P}(\sigma) = T\right\}\right| = \left|\left\{\big(\mathrm{P}(\sigma), \mathrm{Q}(\sigma)\big) : \sigma \in \mathfrak{S}(\mathcal{B}) \wedge \mathrm{P}(\sigma) = T\right\}\right|.$$

It follows that

$$\left\{\big(\mathrm{P}(\sigma), \mathrm{Q}(\sigma)\big) : \sigma \in \mathfrak{S}(\mathcal{B}) \wedge \mathrm{P}(\sigma) = T\right\} \subseteq \{T\} \times \mathrm{PSTab}_{\mathcal{B}}(\lambda)$$

and thus we deduce the result. ☐

The inequality of the previous proposition is, in general, strict. For instance, considering the set $\mathrm{PSTab}_{\{2,4,5,6\}}\big((2,2)\big) = \left\{\begin{array}{|c|c|}\hline 4 & 6 \\ \hline 2 & 5 \\ \hline\end{array}, \begin{array}{|c|c|}\hline 5 & 6 \\ \hline 2 & 4 \\ \hline\end{array}, \begin{array}{|c|c|}\hline 6 & 5 \\ \hline 2 & 4 \\ \hline\end{array}\right\}$ and fixing $T = \begin{array}{|c|c|}\hline 5 & 6 \\ \hline 2 & 4 \\ \hline\end{array}$, then reading T according to all the possibilities for recording tableaux in $\mathrm{PSTab}_{\{2,4,5,6\}}\big((2,2)\big)$ leads to the standard words $\sigma = 5264, \tau = 5624, \upsilon = 5642$, respectively. However, according to Algorithm 2.2, only σ and τ insert to T .

References

[AD99] D. Aldous & P. Diaconis. 'Longest increasing subsequences: from patience sorting to the Baik-Deift-Johansson theorem'. *Bull. Amer. Math. Soc. (N.S.)*, 36, no. 4 (1999), pp. 413–432. DOI: 10.1090/S0273-0979-99-00796-X.

[BL07] A. Burstein & I. Lankham. 'Combinatorics of patience sorting piles'. *Sém. Lothar. Combin.*, 54A (2005/07), pp. Art. B54Ab, 19.

[CG96] J. H. Conway & R. K. Guy. *The book of numbers*. Copernicus, New York, 1996. DOI: 10.1007/978-1-4612-4072-3.

[CMS17] A. J. Cain, A. Malheiro, & F. M. Silva. 'The monoids of the patience sorting algorithm', 2017. arXiv: arXiv:1706.06884.

[FRT+54] J. S. Frame, G. d. B. Robinson, R. M. Thrall, et al. 'The hook graphs of the symmetric group'. *Canad. J. Math*, 6, no. 316 (1954), p. C324.

[Ful97] W. Fulton. *Young tableaux*, vol. 35 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 1997.

[HNT05] F. Hivert, J.-C. Novelli, & J.-Y. Thibon. 'The algebra of binary search trees'. *Theoret. Comput. Sci.*, 339, no. 1 (2005), pp. 129–165. DOI: 10.1016/j.tcs.2005.01.012.

[How95] J. M. Howie. *Fundamentals of semigroup theory*, vol. 12 of *London Mathematical Society Monographs. New Series*. The Clarendon Press, Oxford University Press, New York, 1995. Oxford Science Publications.

[Knu70] D. E. Knuth. 'Permutations, matrices, and generalized Young tableaux'. *Pacific J. Math.*, 34 (1970), pp. 709–727. URL: http://projecteuclid.org/euclid.pjm/1102971948.

[Lot02] M. Lothaire. *Algebraic combinatorics on words*, vol. 90 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2002. DOI: 10.1017/CBO9781107326019.

[LS81] A. Lascoux & M.-P. Schützenberger. 'Le monoïde plaxique'. In *Noncommutative structures in algebra and geometric combinatorics (Naples, 1978)*, vol. 109 of *Quad. "Ricerca Sci."*, pp. 129–156. CNR, Rome, 1981.

[Mal62] C. L. Mallows. 'Patience sorting'. *SIAM Review*, 4, no. 2 (1962), pp. 148–149. DOI: 10.1137/1004036.

[Mal63] C. L. Mallows. 'Patience sorting'. *SIAM Review*, 5, no. 4 (1963), pp. 375–376. DOI: 10.1137/1005107.

[Rey07] M. Rey. 'Algebraic constructions on set partitions'. In *19th International Conference on Formal Power Series and Algebraic Combinatorics*. 2007. URL: `http://www-igm.univ-mlv.fr/~fpsac/FPSAC07/SITE07/PDF-Proceedings/Talks/67.pdf`.

[Rob38] G. d. B. Robinson. 'On the Representations of the Symmetric Group'. *Amer. J. Math.*, 60, no. 3 (1938), pp. 745–760. URL: `https://doi.org/10.2307/2371609`.

[Sag80] B. Sagan. 'On selecting a random shifted young tableau'. *Journal of Algorithms*, 1, no. 3 (1980), pp. 213–234.

[Sch61] C. Schensted. 'Longest increasing and decreasing subsequences'. *Canad. J. Math.*, 13 (1961), pp. 179–191.

[SY89] B. E. Sagan & Y. N. Yeh. 'Probabilistic algorithms for trees'. *Fibonacci Quart*, 27, no. 3 (1989), pp. 201–208.

[Thr52] R. M. Thrall. 'A combinatorial problem'. *Michigan Math. J.*, 1 (1952), pp. 81–88. URL: `http://projecteuclid.org/euclid.mmj/1028989731`.

Centro de Matemática e Aplicações, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
E-mail address: `a.cain@fct.unl.pt`

Centro de Matemática e Aplicações and Departamento de Matemática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
E-mail address: `ajm@fct.unl.pt`

Departamento de Matemática and CEMAT-CIÊNCIAS, Faculdade de Ciências, Universidade de Lisboa, Lisboa 1749-016, Portugal.
E-mail address: `femsilva@fc.ul.pt`