



HAL
open science

String of columns rewriting and confluence of the jeu de taquin

Nohra Hage, Philippe Malbos

► **To cite this version:**

Nohra Hage, Philippe Malbos. String of columns rewriting and confluence of the jeu de taquin. 2020. ⟨hal-03161577v2⟩

HAL Id: hal-03161577

<https://hal.science/hal-03161577v2>

Preprint submitted on 7 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

STRING OF COLUMNS REWRITING AND CONFLUENCE OF THE JEU DE TAQUIN

NOHRA HAGE PHILIPPE MALBOS

Abstract – Schützenberger’s jeu de taquin is an algorithm on the structure of tableaux, which transforms a skew tableau into a Young one by local transformation rules on the columns of the tableaux. This algorithm defines an equivalence relation on tableaux compatible with the plactic congruence, and gives a proof of the Littlewood–Richardson rule on Schur polynomials. In this article, we introduce the notion of string of columns rewriting system as mechanism of transformations of glued sequences of columns. We describe the execution of the jeu de taquin algorithm as rewriting paths of a string of columns rewriting. We deduce algebraic properties on the plactic congruence and we relate the jeu de taquin to insertion algorithms on tableaux.

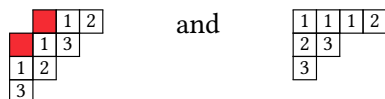
Keywords – Jeu de taquin, string (of columns) rewriting, Young tableaux, plactic monoids.

M.S.C. 2010 – Primary: 20M05. **Secondary:** 05E99, 68Q42, 20M35.

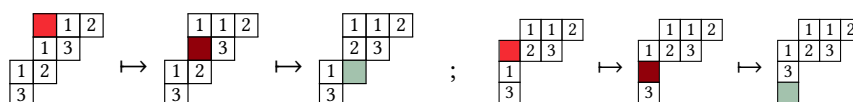
1. INTRODUCTION

Schützenberger introduced the *jeu de taquin* as an algorithm on the structure of Young tableaux to prove the *Littlewood–Richardson rule* on the multiplicity of a Schur polynomial in a product of Schur polynomials, namely the multiplicity of an irreducible representation of the general Lie algebra in a tensor product of two irreducible representations, [22]. The jeu de taquin has later found many applications in algebraic combinatorics and probabilistic combinatorics [4, 20, 25], and many similar algorithms were also introduced on other structures of tableaux, [9, 15, 16, 20, 23, 24].

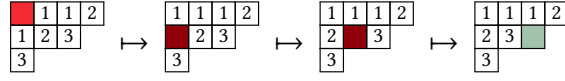
A *Young tableau* is a collection of boxes in left-justified rows filled with elements of the totally ordered alphabet $[n] := \{1 < \dots < n\}$, where the entries weakly increase along each row and strictly increase down each column. A *skew tableau* is obtained by eliminating boxes from the rows of a Young tableau starting from top to bottom and from left to right. The eliminated boxes located above and to the left of two non-empty boxes are called *inner corners* of the skew tableau. We read tableaux column-wise from left to right and from bottom to top: the following tableaux



are respectively skew tableau and Young tableau whose readings are 3121312 and 3213112, and where the empty red boxes denote the inner corners. The jeu taquin consists in applying successively *forward sliding operations* on a skew tableau that move an inner corner into an outer position by keeping the rows weakly increasing and the columns strictly increasing, until no more inner corners remain in the initial skew tableau, as follows



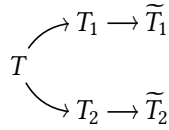
1. Introduction



Schützenberger proved remarkable properties of the jeu de taquin on skew tableaux, [22]. He proved that the *rectification* of a skew tableau by the jeu de taquin is a Young tableau whose reading is equivalent to the reading of the initial skew tableau with respect the *plactic congruence* relation generated by the following *Knuth relations*, [13]:

$$zxy = xzy, \text{ for } 1 \leq x \leq y < z \leq n, \text{ and } yzx = yxz, \text{ for } 1 \leq x < y \leq z \leq n.$$

This congruence defines the *plactic monoid of type A*, [14], which emerged from the works of Schensted [21] and Knuth [13] on the combinatorial study of Young tableaux. Plactic monoids have found several applications in algebraic combinatorics, representation theory, probabilistic combinatorics and rewriting theory, [2–4, 8, 10, 17]. Schützenberger proved that the resulting Young tableau does not depend on the order in which we choose inner corners in the forward slidings. This is the *confluence property* of the jeu de taquin. His proof follows the *cross-section property* of Young tableaux with respect the plactic congruence, proved by Knuth in [13], namely two words on $[n]$ are plactic congruent if and only if they lead to the same Young tableau after applying Schensted’s insertion algorithm, [21]. Explicitly, if there are two sequences of sliding operations that transform a tableau T into two different tableaux T_1 and T_2 , then we continue applying sliding operations until we reach normal forms \tilde{T}_1 and \tilde{T}_2 , that is tableaux without inner corners:

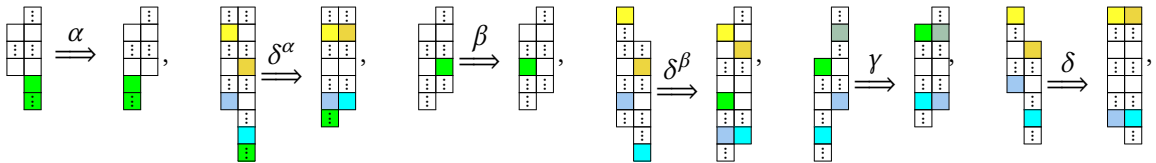


Since \tilde{T}_1 and \tilde{T}_2 are two Young tableaux such that their readings are plactic congruent, following the cross-section property, we deduce that $\tilde{T}_1 = \tilde{T}_2$.

In this article, we introduce a machinery to prove by using a rewriting approach the confluence of the jeu de taquin. We define the sliding operations as rewriting rules on *strings of columns*, that is strings composed by glued sequences of columns, with a gluing map that describes the relative positions of columns. This combinatorial structure generalizes many structures of tableaux such as skew tableaux, [22], Young tableaux of type A, [26], Young tableaux of type B, C, D and G_2 , [12], quasi-ribbon tableaux, [19], and patience sorting structures [1]. In Subsection 2.2 we define a *string of columns rewriting system* as a binary relation on the set of strings of columns, whose *rules* are applied with respect to right and left positions. That is, a set of rules of the following form

$$|_{p_s} u |_{q_s} \Rightarrow |_{p_t} v |_{q_t},$$

where u, v are strings of columns and p_s, q_s, p_t, q_t are positions in \mathbb{Z} . In Subsection 3.2 we present the jeu de taquin by the rewriting system \mathcal{FS}_n , whose rules are of the following form:



see Subsection 3.2.1 for detailed positions of the columns. For instance, the rectification of the above skew tableau is computed with the following reductions:

$$\begin{array}{c} \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 1 & 3 \\ \hline 3 & \end{array} \end{array} \xrightarrow{\gamma} \begin{array}{c} \begin{array}{|c|c|c|} \hline 1 & 1 & 2 \\ \hline 2 & 3 & \end{array} \\ \begin{array}{|c|} \hline 3 \\ \hline \end{array} \end{array} \xrightarrow{\gamma} \begin{array}{c} \begin{array}{|c|c|c|} \hline 1 & 1 & 2 \\ \hline 1 & 2 & 3 \\ \hline 3 & & \end{array} \end{array} \xrightarrow{\beta} \begin{array}{c} \begin{array}{|c|c|c|c|} \hline 1 & 1 & 1 & 2 \\ \hline 2 & & 3 & \end{array} \\ \begin{array}{|c|} \hline 3 \\ \hline \end{array} \end{array} \xrightarrow{\alpha} \begin{array}{c} \begin{array}{|c|c|c|c|} \hline 1 & 1 & 1 & 2 \\ \hline 2 & 3 & & \end{array} \\ \begin{array}{|c|} \hline 3 \\ \hline \end{array} \end{array}$$

The main result of this article, Theorem 3.2.3, states that

- i) *The rewriting system \mathcal{FS}_n is confluent and terminating.*
- ii) *The normal forms with respect to \mathcal{FS}_n are Young tableaux.*
- iii) *Left and right Schensted's insertion algorithms coincide with the leftmost and rightmost normalization strategies of \mathcal{FS}_n .*
- iv) *The rewriting system \mathcal{FS}_n computes the cross-section property.*

The second result of this article, Theorem 3.3.1, proves the compatibility of the rewriting system \mathcal{FS}_n with respect the plactic congruence. Finally, from Theorems 3.2.3 and 3.3.1 we recover the cross-section property of Young tableaux with respect the plactic congruence and the commutation of right and left Schensted's insertion algorithms. In particular, we prove that the rectification map defines a surjective morphism of monoids between the sets of diagonal skew tableaux and the set of Young tableaux equipped with the corresponding insertion products.

Notations. We will consider the totally ordered set $[n] := \{1 < \dots < n\}$, for $n \in \mathbb{Z}_{>0}$, as ground alphabet. We denote by $[n]^*$ the free monoid of words over $[n]$, whose empty word is denoted by λ . We will denote by $|w|$ the length of a word w over $[n]$. We will denote by $<_{lex}$ the lexicographic order on $[n]^*$ induced by the order on $[n]$, and by \leq_{lex} and \leq_{revlex} the lexicographic and the reverse lexicographic order respectively on tuples of natural numbers.

2. STRING OF COLUMNS REWRITING

This section deals with two-dimensional strings defined by gluing columns by introducing the notion of *string of columns* as a generalization of the structure of Young tableaux. We define in Subsection 2.2 the notion of *string of columns rewriting system* as a binary relation on the set of string of columns, whose rules are applied with respect to right and left positions and we show rewriting properties on these rewriting systems.

2.1. Strings of columns

2.1.1. Columns. A *column* (over $[n]$) is a decreasing string $c^k \dots c^1$ over $[n]$, i.e., with $c^{i+1} > c^i$, for $1 \leq i < k$. It is represented by a collection of boxes in left-justified rows, filled with elements of $[n]$, whose each row contains only one box, and the entries strictly increase down:

$$c = \begin{array}{|c|} \hline c^1 \\ \hline c^2 \\ \hline \vdots \\ \hline c^k \\ \hline \end{array}$$

2. String of columns rewriting

where $1 \leq c^1 < \dots < c^k \leq n$, and is also denoted by $(c^1; \dots; c^k)$. Denote $|c| := k$ the *length* of c . We denote by Col_n the set of columns over $[n]$. A column of length 0 is the *empty column* denoted by λ_c .

2.1.2. String of columns. Two columns c_1, c_2 in Col_n can be *glued at position p* in \mathbb{Z} as follows:

$$c_1|_p c_2 = \begin{array}{c} \boxed{c_2^1} \\ \vdots \\ \boxed{c_2^i} \\ \boxed{c_1^1} \boxed{c_2^{i+1}} \\ \boxed{c_1^2} \vdots \\ \vdots \vdots \\ \boxed{c_1^p} \boxed{c_2^{i+p}} \\ \vdots \\ \boxed{c_1^k} \end{array}$$

For $1 \leq j \leq p$, we say that (c_1^j, c_2^{i+j}) is a *full row of length 2* in $c_1|_p c_2$. A pair (\square, c_2^j) , for $1 \leq j \leq i$, and a pair (c_1^j, \square) , for $p+1 \leq j \leq k$, is called a *row of length 2*, where \square denotes the *empty box*. A *string of columns* is a sequence of glued columns:

$$w = c_1|_{p_1} c_2|_{p_2} \dots c_m|_{p_m} c_{m+1}. \quad (2.1.3)$$

The sequence (p_1, p_2, \dots, p_m) in \mathbb{Z}^m is called the *gluing sequence* of w . Gluing sequences can be defined in a consistent way by considering a *gluing map* $g : \text{Col}_n \times \text{Col}_n \rightarrow \mathbb{Z}$ that associates to columns c and c' , a gluing position $g(c, c')$. Given a gluing map g , we define the set of strings of columns constructed with respect to g as the set of string of columns of the form (2.1.3), where for any $1 \leq i \leq m$, $p_i = g(c_i, c_{i+1})$. The *total length* of w is the tuple $tl(w) = (|c_1|, \dots, |c_{m+1}|) \in \mathbb{N}^{m+1}$. We will denote by $\|w\|$ the number of columns of w .

For $3 \leq k \leq m+1$, a *connected row of length k* is a sequence $(c_{i_1}^{j_1}, \dots, c_{i_k}^{j_k})$ such that $(c_{i_l}^{j_l}, c_{i_{l+1}}^{j_{l+1}})$ is a full row of length 2 in $c_{i_l}|_{p_l} c_{i_{l+1}}$, for $1 \leq l \leq k$. A connected row $(c_{i_1}^{j_1}, \dots, c_{i_k}^{j_k})$ is *increasing* if $c_{i_1}^{j_1} \leq \dots \leq c_{i_k}^{j_k}$. We call a *row (over $[n]$)* a string of columns whose gluing sequence is constant equal to 1 and columns are of length 1. A string of columns is *row connected* (resp. *row increasing*) if all its rows are connected (resp. increasing).

2.1.4. Monoids of string of columns. We will denote by Scol_n the set of strings of columns over $[n]$ and by Scol_n^{\leq} the set of row connected and row increasing string of columns over $[n]$.

Given a fixed gluing map g , we define a concatenation operation with respect to g by the map $\cdot|_g : \text{Scol}_n \times \text{Scol}_n \rightarrow \text{Scol}_n$, by setting

$$(c_1|_{p_1} \dots |_{p_m} c_{m+1}) |_g (c'_1|_{q_1} \dots |_{q_n} c'_{n+1}) = c_1|_{p_1} \dots |_{p_m} c_{m+1} |_{g(c_{m+1}, c'_1)} c'_1|_{q_1} \dots |_{q_n} c'_{n+1}.$$

The operation $|_g$ is associative and unitary, where the identity is the *empty string of columns* denoted by λ_c . We denote by Scol_n^g the set of string of columns in Scol_n whose gluing sequence is given by the gluing map g . In other words, Scol_n^g is the free monoid on Col_n with respect the product $|_g$.

2.1.5. The four corner readings. The *south-west reading* is the map $R_{SW} : \text{Scol}_n \rightarrow [n]^*$ that reads a string of columns, column-wise from left to right and from bottom to top. There are three other corner readings R_{NW} , R_{NE} , R_{SW} and R_{SE} defined in a similar way and that read a string of columns, column by column, with respect right or left and top or bottom directions.

Define the map $\text{Fac} : [n]^* \rightarrow [n]^*$ sending a word w into the factorization $w = w_1 \dots w_k$, where each w_i , for $i = 1, \dots, k$, is a maximal strictly decreasing sequence, that is, the R_{SW} -reading of a column in Col_n . For a fixed gluing map $g \in \mathbb{Z}^n$, consider the map

$$[\]_g : [n]^* \rightarrow \text{Scol}_n \quad (2.1.6)$$

that transforms each word w in $[n]^*$ into a string of columns (c_1, \dots, c_k) where each column c_i is filled by the elements of w_i in $\text{Fac}(w)$ from bottom to top, for $i = 1, \dots, k$, with respect the gluing map g .

2.1.7. Properties of strings of columns. A row connected string of columns w as in (2.1.3) is called

- i) *left-justified* (resp. *right-justified*) if $|c_i| \geq |c_{i+1}|$ and $|c_{i+1}| \leq p_i \leq |c_i|$ (resp. $|c_{i+1}| \geq |c_i|$ and $|c_i| \leq p_i \leq |c_{i+1}|$), for all $1 \leq i \leq m$.
- ii) *top-justified*, (resp. *bottom-justified*) if $p_i = |c_{i+1}|$ (resp. $p_i = |c_i|$), for all $1 \leq i \leq m$.
- iii) *decreasing* (resp. *increasing*) if its gluing sequence is decreasing (resp. increasing).

2.1.8. Example: skew tableaux. A *skew tableau* with $m+1$ columns is a string of columns $c_1|_{p_1} \dots |_{p_m} c_{m+1}$ in Scol_n^{\leq} , whose gluing sequence satisfies $p_k \leq |c_{k+1}|$, for all $1 \leq k \leq m$. A *diagonal skew tableau* is a skew tableau $c_1|_{p_1} \dots |_{p_m} c_{m+1}$ whose gluing sequence satisfies $p_k = 1$, for all $1 \leq k \leq m$. We denote by s the gluing map for diagonal skew tableaux. We will denote by Sk_n (resp. dSk_n) the set of skew (resp. diagonal skew) tableaux over $[n]$. Any string u over $[n]$ is the R_{SW} -reading of a unique diagonal skew tableau, thus the map R_{SW} defines a bijection from dSk_n to $[n]^*$. An *inner corner* in a skew tableau w is an empty box located above and to the left of two non-empty boxes. An *outer corner* in w is an empty box located to the end of a row or at the bottom of a column.

We define the *top* (resp. *bottom*) *concatenation* of an element x in a column $c = (c^1; \dots; c^k)$ as the skew tableau defined by

$$c \leftarrow_a x = \begin{cases} c|_1 x & \text{if } x \geq c^1, \\ (x; c^1; \dots; c^k) & \text{else.} \end{cases} \quad (\text{resp. } x \rightsquigarrow_a c = \begin{cases} (c^1; \dots; c^k; x) & \text{if } x > c^k, \\ x|_1 c & \text{else.} \end{cases}).$$

We extend these concatenations into insertion maps on skew tableaux, defined for $x \in [n]$ and $w = c_1|_{p_1} \dots |_{p_m} c_m$ in Sk_n by setting

$$w \leftarrow_{I_r^a} x = c_1|_{p_1} \dots |_{p_m} (c_m \leftarrow_a x), \quad (\text{resp. } x \rightsquigarrow_{I_r^a} w = (x \rightsquigarrow_a c_1)|_{p_1} \dots |_{p_m} c_m). \quad (2.1.9)$$

For any word $w = x_1 \dots x_k$, denote by $C_{d\text{Sk}_n^r}(w)$ (resp. $C_{d\text{Sk}_n^c}(w)$) the diagonal skew tableau obtained from w by inserting its letters iteratively from left to right (resp. right to left) using the right (resp. left) insertion starting from the empty tableau:

$$C_{d\text{Sk}_n^r}(w) := (\emptyset \leftarrow_{I_r^a} w) = ((\dots (\emptyset \leftarrow_{I_r^a} x_1) \leftarrow_{I_r^a} \dots) \leftarrow_{I_r^a} x_k), \\ (\text{resp. } C_{d\text{Sk}_n^c}(w) := (w \rightsquigarrow_{I_r^a} \emptyset) = (x_1 \rightsquigarrow_{I_r^a} (\dots \rightsquigarrow_{I_r^a} (x_k \rightsquigarrow_{I_r^a} \emptyset) \dots))).$$

2. String of columns rewriting

Define now an internal product $\star_{I_r^a}$ (resp. $\star_{I_l^a}$) on dSk_n by setting

$$t \star_{I_r^a} t' := (t \leftarrow_{I_r^a} R_{SW}(t')), \quad (\text{resp. } t \star_{I_l^a} t' := (R_{SW}(t') \rightsquigarrow_{I_l^a} t)) \quad (2.1.10)$$

for all t, t' in dSk_n . By definition the relations $t \star_{I_r^a} \emptyset = t$ (resp. $t \star_{I_l^a} \emptyset = t$) and $\emptyset \star_{I_r^a} t = t$ (resp. $\emptyset \star_{I_l^a} t = t$) hold, showing that the product $\star_{I_r^a}$ (resp. $\star_{I_l^a}$) is unitary with respect to \emptyset .

The top (resp. bottom) concatenation on a diagonal skew tableau w acts only on the last (resp. first) column of w and do not change the others columns. As a consequence, for all $x, y \in [n]$, we have the following *commutation property*:

$$y \rightsquigarrow_{I_r^a} (w \leftarrow_{I_r^a} x) = (y \rightsquigarrow_{I_r^a} w) \leftarrow_{I_r^a} x. \quad (2.1.11)$$

Hence, we deduce that the insertion products $\star_{I_r^a}$ and $\star_{I_l^a}$ are associative.

2.1.12. Example: Young tableaux. A *Young tableau* with $m + 1$ columns is a string of columns $c_1 |_{p_1} \dots |_{p_m} c_{m+1}$ in Scol_n^{\leq} whose gluing sequence is decreasing and satisfies

$$p_k = |c_{k+1}| \quad \text{for all } 1 \leq k \leq m. \quad (2.1.13)$$

We denote by \mathcal{Y} the gluing map for Young tableaux, and by Yt_n the set of Young tableaux over $[n]$.

Given a row r (resp. a column c), we denote by $\text{ROWINSERT}(r, x)$ (resp. $\text{COLUMNINSERT}(c, x)$) the procedure that inserts an element x in a row r (resp. column c) and returns a pair (r', y) (resp. (c', y)) made of the resulting row r' (resp. column c') and the bumping element y that can be empty, as follows. If x is bigger or equal (resp. strictly bigger) than all the elements of r (resp. c), then r' (resp. c') is obtained by adding x to the end (resp. the bottom) of r (resp. c) and y is empty. Otherwise, let y be the smallest element of r (resp. c) such that $x < y$ (resp. $x \leq y$), then r' (resp. c') is obtained from r (resp. c) by replacing y by x . The *right* (resp. *left*) *insertion algorithm* computes a tableau $(t \leftarrow_{S_r} x)$ (resp. $(x \rightsquigarrow_{S_l} t)$) as follows,[21]:

RIGHTINSERTYT(t, x)

Input: A Young tableau t and x in $[n]$.

Output: The Young tableau $(t \leftarrow_{S_r} x)$.

$y := x; t' := \emptyset;$

while $y \neq \lambda$ **do**

$r := t[1];$

$t := t/r;$

$(r', y) := \text{ROWINSERT}(r, y)$

$t' := (t'; r')$

end

return $(t'; t)$

Algorithm 1: Schensted's right algorithm

LEFTINSERTYT(t, x)

Input: A Young tableau t and x in $[n]$.

Output: The Young tableau $(x \rightsquigarrow_{S_l} t)$.

$y := x; t' := \emptyset;$

while $y \neq \lambda$ **do**

$c := t_{[1]};$

$t := t/c;$

$(c', y) := \text{COLUMNINSERT}(c, y)$

$t' := [t'; c']$

end

return $[t'; t]$

Algorithm 2: Schensted's left algorithm

where $t[i]$ (resp. $t_{[i]}$) denotes the i -th row (resp. column) of the tableau t , and $t/t[1]$ (resp. $t_{/t_{[1]}}$) the Young tableau obtained from t by removing its first row (resp. column), and where $(t; t')$ (resp. $[t; t']$) denotes the Young tableau obtained by concatenating t over (resp. to the right of) a Young tableau t' when the concatenation defines a Young tableau.

For any word $w = x_1 \dots x_k$, denote by $C_{Y_n^r}(w)$ (resp. $C_{Y_n^c}(w)$) the Young tableau obtained from w by inserting its letters iteratively from left to right (resp. right to left) using the right (resp. left) insertion starting from the empty tableau:

$$\begin{aligned} C_{Y_n^r}(w) &:= (\emptyset \leftarrow_{S_r} w) = ((\dots (\emptyset \leftarrow_{S_r} x_1) \leftarrow_{S_r} \dots) \leftarrow_{S_r} x_k), \\ (\text{resp. } C_{Y_n^c}(w) &:= (w \rightsquigarrow_{S_l} \emptyset) = (x_1 \rightsquigarrow_{S_l} (\dots \rightsquigarrow_{S_l} (x_k \rightsquigarrow_{S_l} \emptyset) \dots))). \end{aligned}$$

Define now an internal product \star_{S_r} (resp. \star_{S_l}) on Yt_n by setting

$$t \star_{S_r} t' := (t \leftarrow_{S_r} R_{SW}(t')), \quad (\text{resp. } t \star_{S_l} t' := (R_{SW}(t') \rightsquigarrow_{S_l} t)) \quad (2.1.14)$$

for all t, t' in Yt_n . By definition the relations $t \star_{S_r} \emptyset = t$ (resp. $t \star_{S_l} \emptyset = t$) and $\emptyset \star_{S_r} t = t$ (resp. $\emptyset \star_{S_l} t = t$) hold, showing that the product \star_{S_r} (resp. \star_{S_l}) is unitary with respect to \emptyset .

Let c be a column of length p , the *Schützenberger involution* of c , denoted by c^* , is the column of length $n - p$ obtained by taking the complement of the elements of c . This involution is extended to string of columns by setting $(c_1 \dots |c_r)^* = c_r^* | \dots | c_1^*$, for all c_1, \dots, c_r in Col_n . If $c_1 | y \dots | y c_r$ is a Young tableau, then $(c_1 | y \dots | y c_r)^* = c_r^* | y \dots | y c_1^*$ is also a Young tableau. Moreover, the following equality $(c_1 \star_{S_r} \dots \star_{S_r} c_r)^* = (c_r^* \star_{S_r} \dots \star_{S_r} c_1^*)$ holds, for all c_1, \dots, c_r in Col_n . In particular, for three columns c_i, c_j and c_k in Col_n , we have $(c_i \star_{S_r} c_j \star_{S_r} c_k)^* = (c_k^* \star_{S_r} c_j^* \star_{S_r} c_i^*)$, see [10, Remark 3.2.7].

2.2. String of columns rewriting

2.2.1. Rewriting steps. Let define $\text{Scol}_n^{\mathcal{P}} = \mathbb{Z} \times \text{Scol}_n \times \mathbb{Z}$, whose elements are triples (p, u, q) where p, q are positions in \mathbb{Z} and u is a string of columns, that we will denote by $|_p u|_q$. We define a *string of columns rewriting system*, called *rewriting system* for short in the sequel, as a binary relation on $\text{Scol}_n^{\mathcal{P}}$. That is, a set of *rules* of the form

$$\alpha : |_{p_s} u|_{q_s} \Rightarrow |_{p_t} v|_{q_t}, \quad (2.2.2)$$

where u, v are strings of columns in Scol_n and p_s, q_s, p_t, q_t are positions in \mathbb{Z} . The pair (p_s, q_s) (resp. (p_t, q_t)) is called the *source* (resp. *target*) *positions*, and u (resp. v) is called the *string of columns source* (resp. *target*) of the rule α , denoted by $s(\alpha)$ (resp. $t(\alpha)$).

A string of columns w is said to be *reducible with respect to* α , if there is a factorization $w = w_1 |_{p_s} s(\alpha) |_{q_s} w_2$ in Scol_n . In that case, w *reduces into* $w' = w_1 |_{p_t} t(\alpha) |_{q_t} w_2$. Such a *reduction* is denoted by $w_1 |_{p_s} \alpha |_{q_s} w_2$, or α if there is no possible confusion. Given a rewriting system \mathcal{R} , the set of all reductions defines a binary relation on Scol_n , called the \mathcal{R} -*rewrite relation* that we will denote by $\Rightarrow_{\mathcal{R}}$, or \Rightarrow if there is no possible confusion. The elements of $\Rightarrow_{\mathcal{R}}$ are called \mathcal{R} -*rewriting steps*, and have the form

$$|_{p_1} w_1 |_{p_s} s(\alpha) |_{q_s} w_2 |_{p_2} \Rightarrow |_{p_1} w_1 |_{p_t} t(\alpha) |_{q_t} w_2 |_{p_2}, \quad (2.2.3)$$

for all α in \mathcal{R} and w_1, w_2 in Scol_n . In (2.2.3) the data $|_{p_1} w_1 |_{p_s} - |_{q_s} w_2 |_{p_2}$ is called the *context* of the rule α . If we denote by C this context, the reduction (2.2.3) can be also denoted by $C[\alpha]$.

We denote by $\Rightarrow_{\mathcal{R}}^*$ the reflexive and transitive closure of the relation $\Rightarrow_{\mathcal{R}}$, whose elements are called \mathcal{R} -*rewriting paths*.

2. String of columns rewriting

2.2.4. Rewriting properties. A rewriting system \mathcal{R} is *terminating* if there is no infinite \mathcal{R} -rewriting path. A *local branching* (resp. *branching*) of \mathcal{R} is a pair (φ, ψ) of \mathcal{R} -rewriting steps (resp. \mathcal{R} -rewriting paths) having the same source as depicted in the following reduction diagram:

$$\begin{array}{ccc} & \varphi & \rightarrow |_{p_1} w_1 |_{q_1} \\ & \searrow & \\ |_{p_0} w |_{q_0} & & \\ & \searrow & \\ & \psi & \rightarrow |_{p_2} w_2 |_{q_2} \end{array}$$

Such a branching is *confluent* if there exist \mathcal{R} -rewriting paths φ' and ψ' with a common target as follows:

$$\begin{array}{ccccc} & \varphi & \rightarrow & |_{p_1} w_1 |_{q_1} & \xrightarrow{\varphi'} & |_{p'} w' |_{q'} \\ & \searrow & & & & \\ |_{p_0} w |_{q_0} & & & & & \\ & \searrow & & & & \\ & \psi & \rightarrow & |_{p_2} w_2 |_{q_2} & \xrightarrow{\psi'} & |_{p'} w' |_{q'} \end{array} \quad (2.2.5)$$

We say that \mathcal{R} is *locally confluent* (resp. *confluent*) if any local branching (resp. branching) of \mathcal{R} is confluent, and that \mathcal{R} is *convergent* if it is confluent and terminating. A string of columns w is in *normal form with respect to \mathcal{R}* , if there is no rule that reduces w . When \mathcal{R} is convergent, any string of columns w has a unique normal form, denoted by $\text{Nf}(w, \mathcal{R})$.

2.2.6. Critical branching. A local branching of the form (φ, ψ) is called *aspherical*. A local branching (φ, ψ) is called *orthogonal* if the source of φ does not *overlap* with the source of ψ , that is the source of the branching is of the form $|_{p_0} w_1 |_{p_1} s(\varphi) |_{q_1} w_2 |_{p_2} s(\psi) |_{q_2} w_3 |_{p_3}$, with w_1, w_2, w_3 in Scol_n . A local branching that is neither aspherical nor orthogonal is called *overlapping*. There are three shapes of overlapping branchings (φ, ψ) , where $s(\varphi) = |_{p_0} w_1 |_{p_s} s(\alpha) |_{q_s} w_2 |_{p_2}$, and $s(\psi) = |_{p'_0} w'_1 |_{p'_s} s(\beta) |_{q'_s} w'_2 |_{p'_2}$, with $\alpha, \beta \in \mathcal{R}$, described by the following situations:

- i) (*position overlapping*) $q_s = q'_s$,
- ii) (*string overlapping*) $s(\alpha) = |_{p_s} u_\alpha |_{p'_s} v_\alpha |_{q_s}$ and $s(\beta) = |_{p'_s} v_\beta |_{q_s} u_\beta |_{q'_s}$,
- iii) (*inclusion*) $s(\beta) = |_{p'_s} u_\beta |_{p_s} s(\alpha) |_{q_s} v_\beta |_{q'_s}$.

An overlapping branching that is minimal for the relation \sqsubseteq on branchings generated by

$$\begin{array}{ccc} & \varphi & \rightarrow |_{p_1} w_1 |_{q_1} \\ & \searrow & \\ |_{p_0} w |_{q_0} & & \\ & \searrow & \\ & \psi & \rightarrow |_{p_2} w_2 |_{q_2} \end{array} \sqsubseteq \begin{array}{ccc} & |_{p'} u |_{p} \varphi |_{q} v |_{q'} & \rightarrow |_{p'} u |_{p_1} w_1 |_{q_1} v |_{q'} \\ & \searrow & \\ |_{p'} u |_{p} w |_{q} v |_{q'} & & \\ & \searrow & \\ & |_{p'} u |_{p} \psi |_{q} v |_{q'} & \rightarrow |_{p'} u |_{p_2} w_2 |_{q_2} v |_{q'} \end{array}$$

for any branching (φ, ψ) and context $u|_p - |_q v$ of reductions φ, ψ , is called a *critical branching*.

2.2.7. Lemma. A rewriting system \mathcal{R} is *locally confluent* if and only if all its critical branchings are *confluent*. Moreover, if \mathcal{R} is *terminating* with all its critical branchings are *confluent*, then it is *confluent*.

Proof. The first statement is the critical branching lemma. Suppose that all the critical branchings of \mathcal{R} are confluent and prove that any local branching of \mathcal{R} is confluent. By definition, every aspherical branching is trivially confluent, and every orthogonal local branching is confluent. Consider an overlapping but not minimal local branching (φ, ψ) , there exist factorizations $\varphi = C[\varphi']$ and $\psi = C[\psi']$, where (φ', ψ') is a critical branching of \mathcal{R} . By hypothesis, this branching is confluent, and there are reductions paths $\varphi'' : t(\varphi') \rightarrow w$ and $\psi'' : t(\psi') \rightarrow w$ that reduce targets of φ' and ψ' to the same string of columns w . It follows that the reductions paths $C[\varphi'']$ and $C[\psi'']$ make the branching (φ, ψ) confluent.

The second statement is an immediate consequence of Newman's lemma, [18], that proves that any locally confluent terminating rewriting system is confluent. \square

2.2.8. Normalization strategies. A *reduction strategy* for a rewriting system \mathcal{R} specifies a way to apply the rules in a deterministic way. When \mathcal{R} is normalizing, a *normalization strategy* is a mapping σ of every string of columns $|_p w|_q$ to a rewriting path $\sigma|_p w|_q$ with source $|_p w|_q$ and target a chosen normal form of $|_p w|_q$ with respect to \mathcal{R} . For a reduced rewriting system, we distinguish the leftmost reduction strategy and the rightmost one, according to the way we apply first the rewriting rule that reduces the leftmost or the rightmost string of columns. They are defined as follows. For every string of columns $|_p w|_q$, the set of rewriting steps with source $|_p w|_q$ can be ordered from left to right by setting $\varphi < \psi$, for rewriting steps $\varphi = |_p w_1|_{p_s} \alpha |_{q_s} w_2|_q$ and $\psi = |_p w'_1|_{p_s} \beta |_{q_s} w'_2|_q$ such that $\|w_1\| < \|w'_1\|$. If \mathcal{R} is finite, then the order $<$ is total and the set of rewriting steps of source $|_p w|_q$ is finite. Hence this set contains a smallest element $\sigma|_p w|_q$ and a greatest element $\eta|_p w|_q$, respectively called the *leftmost* and the *rightmost rewriting steps on* $|_p w|_q$. If, moreover, the rewriting system terminates, the iteration of σ (resp. η) yields a normalization strategy for \mathcal{R} called the *leftmost* (resp. *rightmost*) *normalization strategy of* \mathcal{R} :

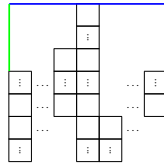
$$\rho_{\mathcal{R}}^{\top}(|_p w|_q) = \sigma|_p w|_q | \rho_{\mathcal{R}}^{\top}(t(\sigma|_p w|_q)) \quad (\text{resp. } \rho_{\mathcal{R}}^{\perp}(|_p w|_q) = \eta|_p w|_q | \rho_{\mathcal{R}}^{\perp}(t(\eta|_p w|_q))).$$

The *leftmost* (resp. *rightmost*) *rewriting path* on a string of columns $|_p w|_q$ is the rewriting path obtained by applying the leftmost (resp. rightmost) normalization strategy $\rho_{\mathcal{R}}^{\top}$ (resp. $\rho_{\mathcal{R}}^{\perp}$). We refer the reader to [6] and [7] for more details on rewriting normalization strategies.

2.2.9. Top-left sliding order. A way to prove termination of a string of columns rewriting system \mathcal{R} is to consider a map $f : \text{Scol}_n \rightarrow (X, <)$, where $(X, <)$ is a well-ordered set satisfying, for all $w, w' \in \text{Scol}_n$,

$$w \Rightarrow_{\mathcal{R}} w' \quad \text{implies} \quad f(w') < f(w).$$

We will use the following termination order. Let $w = u_1|_{p_1} \dots |_{p_{m-1}} u_m$ be in Scol_n . Denote by h_{u_k} the number of empty boxes between the top box of the column u_k and the *top position* of w , shown by the blue line in the following picture



Define the *top deviation* of w as the sequence $d^{\top}(w) = (h_{u_1}, \dots, h_{u_m}) \in \mathbb{N}^m$. Denote by \leq_{lex} the total order on Scol_n defined by $w \leq_{lex} w'$ if and only if

$$tl(w) <_{revlex} tl(w') \quad \text{or} \quad (tl(w) = tl(w') \quad \text{and} \quad d^{\top}(w) <_{lex} d^{\top}(w')).$$

3. Convergence of the jeu de taquin

In order to prove the termination of top-left sliding operations presented in 3.2.1, we define the total order \ll_{tl} on Scol_n by setting, for w, w' in Scol_n , $w \ll_{tl} w'$ if and only if

$$\|w\| < \|w'\| \text{ or } (\|w\| = \|w'\| \text{ and } w \leq_{lex} w').$$

3. CONVERGENCE OF THE JEU DE TAQUIN

In this section, we study the confluence of the jeu de taquin through a rewriting system defined by column slidings. We show that this rewriting system is convergent and we present the jeu de taquin as a surjective map from the set of diagonal skew tableaux to the set of Young tableaux using insertion. We recover properties relating the jeu de taquin to the plactic congruence and insertion algorithms on the structure of Young tableaux.

3.1. Jeu de taquin

3.1.1. Plactic monoids. Recall that the *plactic monoid (of type A) of rank n*, introduced in [14], and denoted by \mathbf{P}_n , is generated on $[n]$ and submitted to the following *Knuth relations*, [13]:

$$zxy = xzy, \text{ for } 1 \leq x \leq y < z \leq n, \text{ and } yzx = yxz, \text{ for } 1 \leq x < y \leq z \leq n. \quad (3.1.2)$$

Knuth in [13] described the congruence $\approx_{\mathbf{P}_n}$ generated by these relations in terms of Young tableaux and proved the cross-section property for the monoid \mathbf{P}_n .

3.1.3. Forward sliding, [22]. A *forward sliding* is a sequence of the following slidings:

$$\begin{array}{|c|} \hline y \\ \hline x \\ \hline \end{array} \leftrightarrow \begin{array}{|c|c|} \hline x & y \\ \hline & \\ \hline \end{array} \text{ for } x \leq y, \quad \begin{array}{|c|} \hline x \\ \hline y \\ \hline \end{array} \leftrightarrow \begin{array}{|c|c|} \hline x & \\ \hline & y \\ \hline \end{array} \text{ for } x < y, \quad \begin{array}{|c|} \hline \\ \hline x \\ \hline \end{array} \leftrightarrow \begin{array}{|c|c|} \hline & x \\ \hline & \\ \hline \end{array}, \quad \begin{array}{|c|} \hline x \\ \hline \\ \hline \end{array} \leftrightarrow \begin{array}{|c|c|} \hline x & \\ \hline & \\ \hline \end{array}$$

starting from a skew tableau and one of its inner corners, and moving the empty box until it becomes an outer corner. The *jeu de taquin* on a skew tableau w consists in applying successively the forward slide algorithm starting from w until we get a string of columns without inner corners denoted $\pi_{tq}(w)$, which is shown to be a Young tableau. In this way, the jeu de taquin defines a map

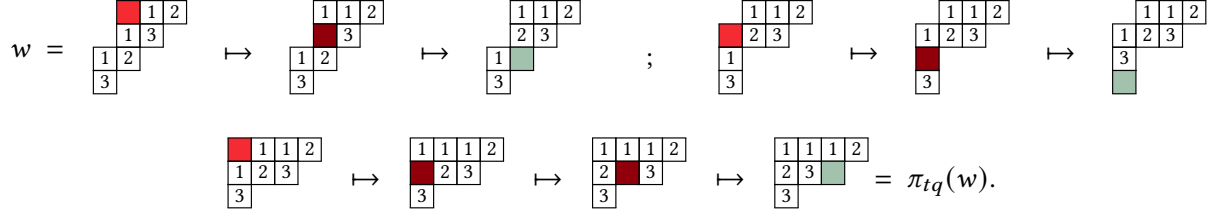
$$\pi_{tq} : \text{Sk}_n \rightarrow \text{Yt}_n,$$

also called the *rectification* of skew tableaux. Schützenberger proved in [22] many properties of the jeu de taquin. These properties are also presented by Fulton in [4], as follows. For any $w \in \text{Sk}_n$, the following conditions hold

- i)** [4, Proposition 2]. $R_{SW}(w) \approx_{\mathbf{P}_n} R_{SW}(\pi_{tq}(w))$,
- ii)** [4, Corollary 1]. The rectification $\pi_{tq}(w)$ is the unique Young tableau satisfying **i)**,
- iii)** [4, Claim 2]. The map π_{tq} does not depend on the order in which the inner corners are chosen.

Note that condition **ii)** is a consequence of the cross-section property for \mathbf{P}_n proved in [13] and condition **i)**. Moreover, condition **iii)** is a consequence of conditions **i)** and **ii)**. In the rest of this section, we show that conditions **i)** and **ii)** are direct consequence of a confluence property of a rewriting system that computes the map π_{tq} , and without supposing the cross-section property for \mathbf{P}_n which will be also consequence of this confluence property.

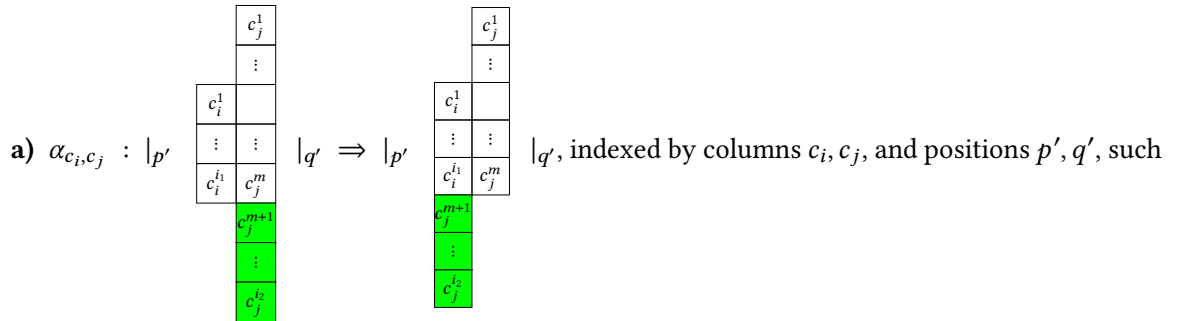
3.1.4. Example. The jeu de taquin on the following skew tableau w starting with the inner corner \blacksquare applies three occurrences of forward sliding, where \blacksquare denotes the empty box, and \blacksquare the outer corner:



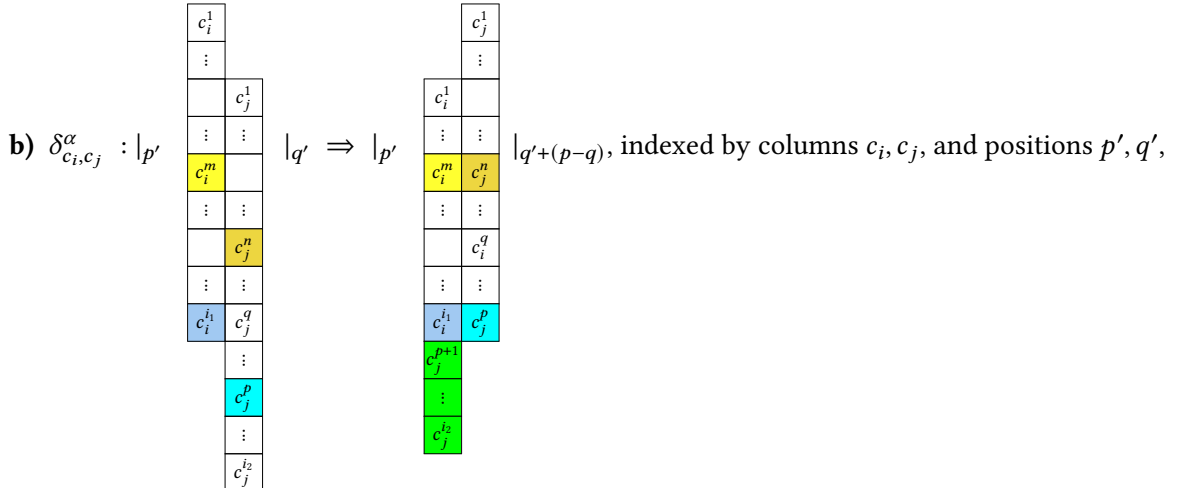
3.2. Jeu de taquin as rewriting

3.2.1. Rules of the jeu de taquin. The jeu de taquin map π_{tq} is described by the union of rewriting systems $\mathcal{FS}_n = \mathcal{LS}_n \cup \mathcal{IS}_n \cup \mathcal{TS}_n$ whose sets of rules are defined as follows.

i) \mathcal{LS}_n the set of *left-sliding* rules that move sub-columns to the left in the following two situations:



that $(c_i^{i_1}, c_j^m)$ is a row, $i_1 \leq m < i_2$ and $c_i |_{(i_1+i_2-m)} c_j \in \text{Scol}_n^{\leq}$.



such that $c_i |_{(i_1+i_2-q)} c_j \notin \text{Scol}_n^{\leq}$, and m is maximal such that $c_i |_{(i_1+i_2-p)} c_j \in \text{Scol}_n^{\leq}$ and $i_1 \leq p$.

ii) \mathcal{IS}_n is the set of *insert-sliding* rules performing insertion in the following two situations:

3. Convergence of the jeu de taquin

a) $\beta_{c_i, c_j} : |_{p'} \Rightarrow |_{p'}$ $|_{q'} \Rightarrow |_{q'}$, indexed by columns c_i, c_j , and positions p', q' , such

	c_j^1
	\vdots
c_i^1	
\vdots	\vdots
c_i^{l-1}	c_j^m
c_i^l	c_j^{m+1}
\vdots	\vdots
c_i^k	$c_j^{i_2}$
\vdots	
$c_i^{i_1}$	

	c_j^1
	\vdots
c_i^1	
\vdots	\vdots
c_i^{l-1}	c_j^{m-1}
c_i^l	c_j^{m+1}
\vdots	\vdots
c_i^l	c_j^{m+2}
\vdots	\vdots
c_i^{k-1}	$c_j^{i_2}$
\vdots	
$c_i^{i_1}$	

that $c_i|_k c_j \in \text{Scol}_n^{\leq}$ with $1 \leq k \leq i_1$ and $k < i_2$, and l is minimal such that (c_i^l, c_j^{m+1}) is a row and $c_i^{l-1} < c_j^m < c_i^l$.

b) $\delta_{c_i, c_j}^\beta : |_{p'} \Rightarrow |_{p'}$ $|_{q'} \Rightarrow |_{q'}$, indexed by columns c_i, c_j , and positions p', q' , such that $c_i|_{(i_1+i_2-q)} c_j \notin \text{Scol}_n^{\leq}$, where p is maximal such that $c_i|_p c_j \in \text{Scol}_n^{\leq}$, and n is minimal such that (c_i^n, c_j^{t+1}) is a row with $c_i^{n-1} < c_j^t < c_i^n$.

	c_j^1
	\vdots
	c_j^1
	\vdots
	c_j^r
	\vdots
c_i^p	
\vdots	\vdots
$c_i^{i_1}$	c_j^q
\vdots	\vdots
	$c_j^{i_2}$
	\vdots
	$c_i^{i_1}$

	c_j^1
	\vdots
c_i^1	c_j^{r-1}
c_i^2	c_j^r
\vdots	\vdots
c_i^{n-1}	c_j^{t-1}
c_i^n	c_j^{t+1}
\vdots	\vdots
	c_j^q
\vdots	\vdots
c_i^p	$c_j^{i_2}$
\vdots	\vdots
$c_i^{i_1}$	

iii) \mathcal{TS}_n is the set of *left top sliding rules* that move columns to the top as follows:

a) $\gamma_{c_i, c_j} : |_{p'} \Rightarrow |_{p'-(s-r)}$ $|_{q'} \Rightarrow |_{q'}$, indexed by columns c_i, c_j and positions p', q' ,

	c_j^1
	\vdots
	c_j^m
	\vdots
c_i^r	
\vdots	\vdots
c_i^s	$c_j^{i_2}$
\vdots	\vdots
$c_i^{i_1}$	

	c_j^1
	\vdots
c_i^1	c_j^m
\vdots	\vdots
c_i^r	
\vdots	\vdots
c_i^s	$c_j^{i_2}$
\vdots	\vdots
$c_i^{i_1}$	

such that $c_i|_r c_j \in \text{Scol}_n^{\leq}$ with $1 \leq r < i_2$, or $c_i|_r c_j$ is not row connected with $c_i^1 \leq c_j^{i_2}$, and s is

maximal such that $c_i|_s c_j \in \text{Scol}_n^{\leq}$ and $s \leq i_2$.

$$\text{b) } \delta_{c_i, c_j} : |_{p'} \begin{array}{c} c_i^1 \\ \vdots \\ c_i^q \\ c_i^p \\ c_i^{i_1} \\ \vdots \\ c_i^{i_2} \end{array} \begin{array}{c} \\ \\ c_j^1 \\ c_j^p \\ c_j^{i_1} \\ \vdots \\ c_j^{i_2} \end{array} |_{q'} \Rightarrow |_{p'} \begin{array}{c} c_i^1 \\ \vdots \\ c_i^q \\ c_i^p \\ c_i^{i_1} \\ \vdots \\ c_i^{i_2} \end{array} \begin{array}{c} c_j^1 \\ c_j^p \\ c_j^{i_1} \\ \vdots \\ c_j^{i_2} \end{array} |_{q'+(i_1-p)},$$

indexed by columns c_i, c_j , and positions p', q' , such that $c_i|_{(i_1+i_2-p)} c_j \notin \text{Scol}_n^{\leq}$ and $c_i|_{i_2} c_j \in \text{Scol}_n^{\leq}$, or $c_i|_{(i_1+i_2-p)} c_j \in \text{Scol}_n^{\leq}$ and $i_1 > p$, or $c_i|_k c_j$ is not row connected with $k > i_1$ and $c_i^{i_1} \leq c_j^1$.

In the sequel, if there is no possible confusion, we will omit the subscripts c_i, c_j in the notation of the rules. Moreover, for any rule μ in \mathcal{FS}_n , we will denote by μ^* any composition of rewriting sequences involving the rules μ and ending on a normal form with respect to μ .

3.2.2. Example. The rectification of the skew tableau w from Example 3.1.4 is computed with the following reduction of \mathcal{FS}_n :

$$w = \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 1 & 2 \\ \hline 1 & 2 \\ \hline 3 & \\ \hline \end{array} \xrightarrow{\gamma_{c_2, c_3}} \begin{array}{|c|c|} \hline 1 & 1 & 2 \\ \hline 2 & 3 & \\ \hline 1 & & \\ \hline 3 & & \\ \hline \end{array} \xrightarrow{\gamma_{c_1, c_2}} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 & 2 \\ \hline 2 & 2 & 3 & \\ \hline 1 & 2 & 3 & \\ \hline 3 & & & \\ \hline \end{array} \xrightarrow{\beta_{c_1, c_2}} \begin{array}{|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 2 \\ \hline 2 & 2 & 3 & 3 & \\ \hline 1 & 2 & 3 & 3 & \\ \hline 3 & & & & \\ \hline \end{array} \xrightarrow{\alpha_{c_2, c_3}} \begin{array}{|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 2 \\ \hline 2 & 2 & 3 & 3 & \\ \hline 1 & 2 & 3 & 3 & \\ \hline 3 & & & & \\ \hline \end{array} = \pi_{tq}(w).$$

3.2.3. Theorem. The rewriting system \mathcal{FS}_n satisfies the following conditions:

- i) \mathcal{FS}_n is convergent.
- ii) The normal form of any skew tableau with respect to \mathcal{FS}_n is a Young tableau.

For every word w in $[n]^*$, we have

- iii) $(\emptyset \leftarrow_{S_r} w) = \rho_{\mathcal{FS}_n}^\top([w]_s)$ and $(w \rightsquigarrow_{S_l} \emptyset) = \rho_{\mathcal{FS}_n}^\perp([w]_s)$,
- iv) $C_{Y_n^r}(w) = \text{Nf}([w]_s, \mathcal{FS}_n)$ and $C_{Y_n^c}(w) = \text{Nf}([w]_s, \mathcal{FS}_n)$.

The rest of this subsection is devoted to the proof of this result. Lemmata 3.2.8 and 3.2.9 show that the rewriting system \mathcal{FS}_n is convergent. As a consequence, we obtain that the normal forms are Young tableaux. We prove in 3.2.6 that right and left Schensted's insertion algorithm coincide respectively with the leftmost and rightmost normalization strategy of \mathcal{FS}_n . Condition **iii)** and convergence of \mathcal{FS}_n yield Condition **iv)**.

3.2.4. Lemma. For any rule $|c_{i_1}|c_{i_2}| \Rightarrow |c_{j_1}|c_{j_2}|$ in \mathcal{FS}_n , the following equality $|c_{i_1} \star_{S_r} c_{i_2}| = \rho_{\mathcal{FS}_n}^\top(|c_{j_1}|c_{j_2}|)$ holds. Moreover, for all $c_i = (c_i^1, \dots, c_i^{i_1})$ and $c_j = (c_j^1, \dots, c_j^{i_2})$ in Col_n , we have

$$c_i \star_{S_r} c_j = \rho_{\mathcal{FS}_n}^\top(c_i|_1 c_j), \quad c_j \star_{S_l} c_i = \rho_{\mathcal{FS}_n}^\perp(c_i|_1 c_j). \quad (3.2.5)$$

3. Convergence of the jeu de taquin

Proof. Prove that for any rule $|c_{i_1}|c_{i_2}| \Rightarrow |c_{j_1}|c_{j_2}|$ in \mathcal{FS}_n , we have $|c_{i_1} \star_{S_r} c_{i_2}| = \rho_{\mathcal{FS}_n}^\top(|c_{j_1}|c_{j_2}|)$. The rules α_{c_i, c_j} , δ_{c_i, c_j}^α , β_{c_i, c_j} and δ_{c_i, c_j}^β followed by β^* yield to the Young tableau $|c_i \star_{S_r} c_j|$. Consider now the rule γ_{c_i, c_j} . If $|c_j| \leq |c_i|$ and $c_i|_{|c_j|}c_j \in \text{Scol}_n$ then the target of γ_{c_i, c_j} is equal to $|c_i \star_{S_r} c_j|$. If $|c_i| < |c_j|$ and $c_i|_{|c_j|}c_j \in \text{Scol}_n$, then the rule γ_{c_i, c_j} is followed by the rule α_{c_i, c_j} in order to obtain $|c_i \star_{S_r} c_j|$. If $c_i|_{|c_j|}c_j \notin \text{Scol}_n$ then the rule γ_{c_i, c_j} followed by α_{c_i, c_j} and then by β^* , or only followed by β^* yield to $|c_i \star_{S_r} c_j|$. Consider finally the rule δ_{c_i, c_j} . If $|c_j| \leq |c_i|$ then the target δ_{c_i, c_j} is equal to $|c_i \star_{S_r} c_j|$. Otherwise, if $|c_i| < |c_j|$ then δ_{c_i, c_j} is followed by α_{c_i, c_j} in order to obtain $|c_i \star_{S_r} c_j|$.

Prove the first equality of (3.2.5) by induction on $|c_j|$, the proof being similar for \star_{S_r} . Suppose that $|c_j| = 1$, we consider the following two cases. If $c_i^1 \leq c_j^1$, then $c_i|_1c_j$ is equal to $c_i \star_{S_r} c_j$. If $c_j^1 < c_i^1$, then by applying δ^β on $c_i|_1c_j$ we obtain $c_i \star_{S_r} c_j$. Suppose the equality holds when $|c_j| = i_2 - 1$, and prove it when $|c_j| = i_2$. First consider the case when $c_i^1 \leq c_j^{i_2}$. Suppose that by induction we have

$$\rho_{\mathcal{FS}_n}^\top \left(\begin{array}{c|c} c_j^{i_2} & \\ \hline \vdots & \\ c_i^1 & c_j^{i_2} \\ \hline \vdots & \\ c_i^{i_1} & \end{array} \right) = \begin{array}{c|c} x_1 & y_1 \\ \hline \vdots & \vdots \\ & y_t \\ \hline \vdots & \\ x_s & \end{array} = c_i \star_{S_r} (c_j^{i_2}, \dots, c_j^{i_2}),$$

where x_i and y_j are elements of c_i and c_j with $t \leq s$. We prove that $\rho_{\mathcal{FS}_n}^\top \left(\begin{array}{c|c} & c_j^1 \\ \hline x_1 & y_1 \\ \hline \vdots & \vdots \\ & y_t \\ \hline \vdots & \\ x_s & \end{array} \right) = c_i \star_{S_r} c_j$, by

considering the following two cases.

Case 1. $x_1 \leq c_j^1$ and $x_{k+1} \leq y_k$, for all $k = 1, \dots, t-1$:

$$\begin{array}{c|c} & c_j^1 \\ \hline x_1 & y_1 \\ \hline \vdots & \vdots \\ x_s & y_t \end{array} \xrightarrow{\gamma} \begin{array}{c|c} x_1 & c_j^1 \\ \hline x_2 & y_1 \\ \hline \vdots & \vdots \\ x_s & y_{t-1} \\ \hline & y_t \end{array} \xrightarrow{\alpha} \begin{array}{c|c} x_1 & c_j^1 \\ \hline x_2 & y_1 \\ \hline \vdots & \vdots \\ x_s & y_{t-1} \\ \hline & y_t \end{array} = c_i \star_{S_r} c_j, \text{ for } s = t$$

$$\text{or} \quad \begin{array}{c|c} & c_j^1 \\ \hline x_1 & y_1 \\ \hline \vdots & \vdots \\ & y_t \\ \hline \vdots & \\ x_s & \end{array} \xrightarrow{\gamma} \begin{array}{c|c} x_1 & c_j^1 \\ \hline x_2 & y_1 \\ \hline \vdots & \vdots \\ & y_t \\ \hline \vdots & \\ x_s & \end{array} = c_i \star_{S_r} c_j, \text{ for } s > t.$$

Case 2. $x_1 \leq c_j^1$ (resp. $c_j^1 < x_1$) and let x_i be minimal such that $x_{i-1} < y_{i-1} < x_i$:

$$\begin{array}{|c|c|} \hline & c_j^1 \\ \hline x_1 & y_1 \\ \hline \vdots & \vdots \\ \hline x_{i-1} & y_{i-1} \\ \hline x_i & y_i \\ \hline \vdots & \vdots \\ \hline & y_t \\ \hline \vdots & \\ \hline x_s & \\ \hline \end{array} \xRightarrow{\beta} \begin{array}{|c|c|} \hline x_1 & c_j^1 \\ \hline x_2 & y_1 \\ \hline \vdots & \vdots \\ \hline y_{i-1} & y_i \\ \hline x_i & y_{i+1} \\ \hline \vdots & \vdots \\ \hline & y_t \\ \hline \vdots & \\ \hline x_s & \\ \hline \end{array} = c_i \star_{S_r} c_j. \quad (\text{ resp. } \begin{array}{|c|c|} \hline & c_j^1 \\ \hline x_1 & y_1 \\ \hline \vdots & \vdots \\ \hline & y_t \\ \hline \vdots & \\ \hline x_s & \\ \hline \end{array} \xRightarrow{\beta} \begin{array}{|c|c|} \hline c_j^1 & y_1 \\ \hline x_1 & y_2 \\ \hline \vdots & \vdots \\ \hline & y_t \\ \hline \vdots & \\ \hline x_s & \\ \hline \end{array} = c_i \star_{S_r} c_j.)$$

Suppose finally that $c_i^1 > c_j^{i_2}$. We obtain:

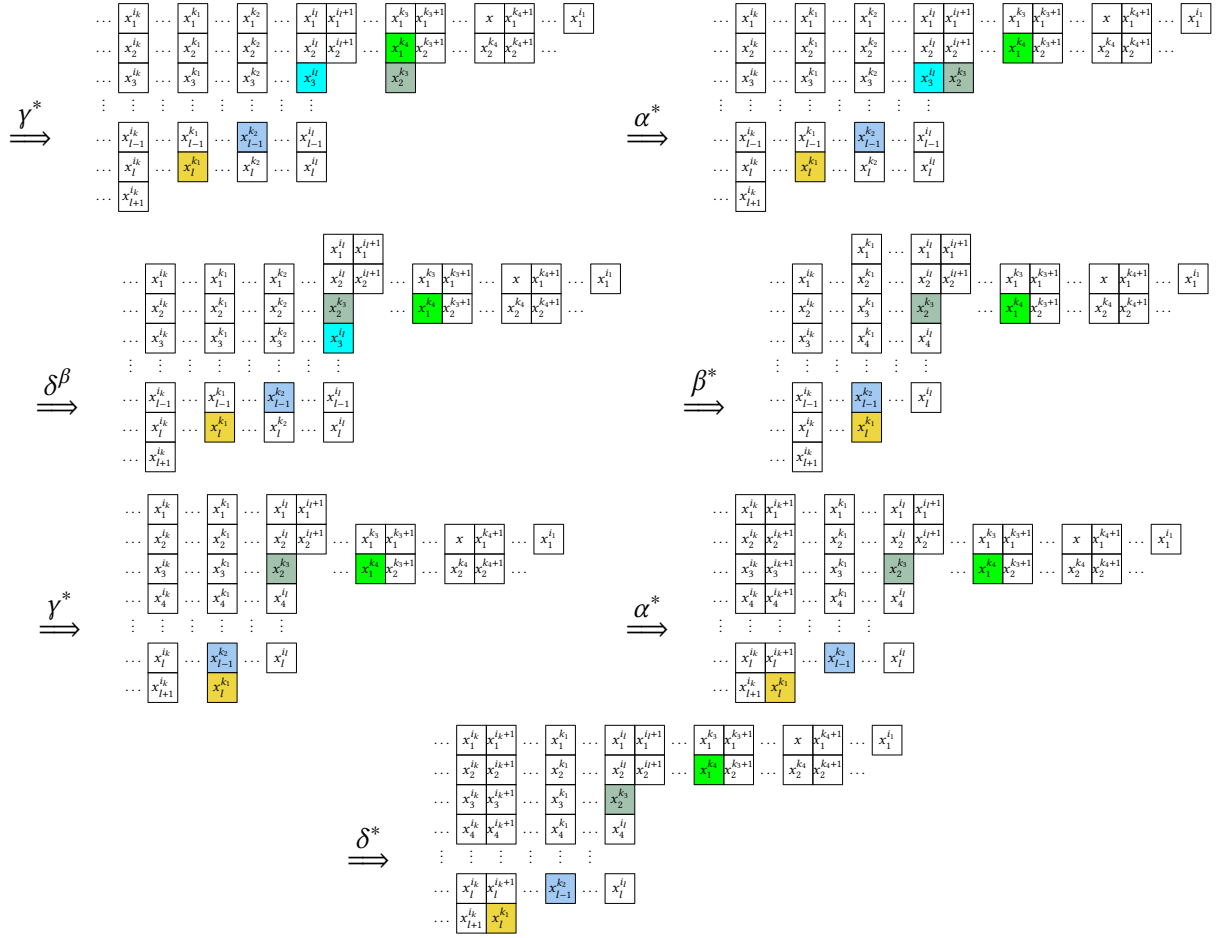
$$\begin{array}{|c|c|} \hline & c_j^1 \\ \hline c_i^1 & c_j^{i_2} \\ \hline \vdots & \\ \hline c_i^{i_1} & \\ \hline \end{array} \xRightarrow{\delta^\beta} \begin{array}{|c|} \hline c_j^1 \\ \hline \vdots \\ \hline c_j^{i_2-1} \\ \hline c_j^{i_2} \\ \hline c_i^1 \\ \hline \vdots \\ \hline c_i^{i_1} \\ \hline \end{array} \xRightarrow{\beta^*} \begin{array}{|c|} \hline c_j^1 \\ \hline \vdots \\ \hline c_j^{i_2} \\ \hline c_i^1 \\ \hline \vdots \\ \hline c_i^{i_1} \\ \hline \end{array} = c_i \star_{S_r} c_j. \quad \square$$

3.2.6. Proof of Theorem 3.2.3 iii). We prove the first equality by induction on the number of columns in $[w]_s$, the proof being similar for the insertion S_l . When $[w]_s$ is of length 2, then the equality is a consequence of Lemma 3.2.4. For $k \geq 3$, suppose that the equality holds for words of length $k - 1$, and consider $[w]_s = c_1 | 1 \dots | 1 c_{k-1}$. By the induction hypothesis, we have $(\emptyset \leftarrow_{S_r} w) = \rho_{\mathcal{F}S_n}^\top(c_1 | 1 \dots | 1 c_{k-1}) \star_{S_r} c_k$. Let us show that

$$\rho_{\mathcal{F}S_n}^\top(c_1 | 1 \dots | 1 c_{k-1}) \star_{S_r} c_k = \rho_{\mathcal{F}S_n}^\top(c_1 | 1 \dots | 1 c_k). \quad (3.2.7)$$

Since inserting c_k into $\rho_{\mathcal{F}S_n}^\top(c_1 | 1 \dots | 1 c_{k-1})$ consists into inserting its elements one by one from bottom to top, it suffices to prove (3.2.7) for $c_k = (x)$. If x is bigger or equal than the last element $x_1^{i_1}$ of the first row of $\rho_{\mathcal{F}S_n}^\top(c_1 | 1 \dots | 1 c_{k-1})$, then $\rho_{\mathcal{F}S_n}^\top(c_1 | 1 \dots | 1 c_{k-1}) | 1 \boxed{x}$ is a Young tableau which is equal to $\rho_{\mathcal{F}S_n}^\top(c_1 | 1 \dots | 1 c_{k-1}) \star_{S_r} \boxed{x}$. Otherwise, if $x < x_1^{i_1}$, we first apply a rule δ^β in order to slide the box containing x to the top of the one containing $x_1^{i_1}$. We then apply the following reduction rules as shown in the following reduction diagrams. Note that, the elements in the colored boxes represent the ones that are bumped when inserting x into the tableau $\rho_{\mathcal{F}S_n}^\top(c_1 | 1 \dots | 1 c_{k-1})$.

3. Convergence of the jeu de taquin



The resulted Young tableau is equal to $\rho_{\mathcal{F}S_n}^\top(c_1 | \dots | c_{k-1}) \star_{S_r} \boxed{x}$, showing the claim.

3.2.8. Lemma. The rewriting system $\mathcal{F}S_n$ is terminating.

Proof. We prove that for any reduction $w \Rightarrow w'$ with respect to $\mathcal{F}S_n$, we have $w \ll_{tl} w'$ for the order \ll_{tl} defined 2.2.9. If $w \Rightarrow w'$ is a reduction with respect to $\mathcal{L}S_n$, then $\|w\| = \|w'\|$ and $tl(w) <_{reollex} tl(w')$, showing that $w \ll_{tl} w'$. Suppose now that the reduction is with respect to $\mathcal{I}S_n$. There are two cases depending on the number of columns in the targets of the rules β and δ^β . If the targets consist only of one column then $\|w\| < \|w'\|$. If they consist of two columns then $\|w\| = \|w'\|$ and $tl(w) <_{reollex} tl(w')$. Then, if $w \Rightarrow w'$ is a reduction with respect to $\mathcal{I}S_n$, we obtain $w \ll_{tl} w'$. Finally, for any reduction $w \Rightarrow w'$ with respect to $\mathcal{T}S_n$, we have $\|w\| = \|w'\|$, $tl(w) = tl(w')$ and $d^\top(w) <_{lex} d^\top(w')$, showing that $w \ll_{tl} w'$. \square

3.2.9. Lemma. The rewriting system $\mathcal{F}S_n$ is confluent.

Proof. Following Lemma 2.2.7, we prove that the rewriting system $\mathcal{F}S_n$ is confluent by showing the confluence of all its critical branchings. Consider first the rewriting system $\mathcal{R}(\text{Col}_n, \mathbb{Y}'_n)$ whose rules are of the form $\gamma_{c,c'} : |c|c'| \Rightarrow |c \star_{S_r} c'|$, for all c, c' in Col_n such that $c|y c' \neq c \star_{S_r} c'$. Prove that starting from a string of columns consisting of three columns $|c_i|c_j|c_k|$, we lead to the Young tableau $|c_i \star_{S_r} c_j \star_{S_r} c_k|$

3. Convergence of the jeu de taquin

consider $R_{SW}(s(\alpha)) = c_i^{i_1} \dots c_i^1 c_j^{i_2} \dots c_j^{m+1} c_j^m \dots c_j^1$ and $R_{SW}(t(\alpha)) = c_j^{i_2} \dots c_j^{m+1} c_i^{i_1} \dots c_i^1 c_j^m \dots c_j^1$. On one hand, we have

$$\begin{aligned} c_i^{i_1} \dots c_i^2 c_i^1 c_j^{i_2} \dots c_j^{m+1} c_j^m \dots c_j^1 &\stackrel{(3.1.2)}{=} \dots \stackrel{(3.1.2)}{=} c_i^{i_1} c_j^{i_2} c_i^{i_1-1} \dots c_i^2 c_i^1 c_j^{i_2-1} \dots c_j^{m+1} c_j^m \dots c_j^1 \\ &\stackrel{(3.1.2)}{=} \dots \stackrel{(3.1.2)}{=} c_i^{i_1} c_j^{i_2} c_j^{i_2-1} c_i^{i_1-1} \dots c_i^2 c_i^1 c_j^{i_2-2} \dots c_j^{m+1} c_j^m \dots c_j^1 \stackrel{(3.1.2)}{=} \\ c_j^{i_2} c_i^{i_1} c_j^{i_2-1} c_i^{i_1-1} \dots c_i^2 c_i^1 c_j^{i_2-2} \dots c_j^{m+1} c_j^m \dots c_j^1 &\stackrel{(3.1.2)}{=} \dots \stackrel{(3.1.2)}{=} c_j^{i_2} \dots c_j^{m+1} c_j^m c_i^{i_1} \dots c_i^1 c_j^{m-1} \dots c_j^1. \end{aligned}$$

In an other hand, we have $c_j^{i_2} \dots c_j^{m+1} c_i^{i_1} \dots c_i^2 c_i^1 c_j^m \dots c_j^1 \stackrel{(3.1.2)}{=} \dots \stackrel{(3.1.2)}{=} c_j^{i_2} \dots c_j^{m+1} c_j^m c_i^{i_1} \dots c_i^1 c_j^{m-1} \dots c_j^1$. Hence, $R_{SW}(s(\alpha)) \approx_{P_n} R_{SW}(t(\alpha))$. Similarly, we show the property for rules β , δ^α and δ^β .

Prove now that for all $w, w' \in [n]^*$, $w \approx_{P_n} w'$ implies $\text{Nf}([w]_s, \mathcal{FS}_n) = \text{Nf}([w']_s, \mathcal{FS}_n)$. Since \mathcal{FS}_n is convergent, we show that for all $w, w' \in [n]^*$, $w \approx_{P_n} w'$ implies $[w]_s \approx_{\mathcal{FS}_n} [w']_s$. Suppose first that $w = ux_1zxy_1v$ and $w' = ux_1zxy_1v$, for all $1 \leq x \leq y < z \leq n$, $u, v \in [n]^*$ and $x_1, y_1 \in [n]$, and show that $[w]_s \approx_{\mathcal{FS}_n} [w']_s$. We consider the following cases:

Case 1. $x_1 \leq x$ and $y \leq y_1$.

$$\begin{array}{ccc} ux_1zxy_1v & \xrightarrow{\quad} & ux_1xzy_1v \\ [w']_s \downarrow & & \downarrow [w]_s \\ [ux_1]_s \mid \begin{array}{|c|} \hline x & y \\ \hline z & \\ \hline \end{array} \mid [y_1v]_s & & [ux_1]_s \mid \begin{array}{|c|} \hline y \\ \hline x & z \\ \hline \end{array} \mid [y_1v]_s \\ & \swarrow \alpha & \searrow \gamma \\ & [ux_1]_s \mid \begin{array}{|c|} \hline x & y \\ \hline z & \\ \hline \end{array} \mid [y_1v]_s & \end{array}$$

Case 2. $x_1 \leq x$ and $y > y_1$. Suppose that $v = y_2 \dots y_q y'v'$ such that $y_1 > y_2 > \dots > y_q$ and $y_q \leq y'$.

$$\begin{array}{ccc} ux_1zxy_1v & \xrightarrow{\quad} & ux_1xzy_1v \\ [w']_s \downarrow & & \downarrow [w]_s \\ [ux_1]_s \mid \begin{array}{|c|} \hline y_q \\ \hline \vdots \\ \hline y_1 \\ \hline x & y \\ \hline z & \\ \hline \end{array} \mid [y'v']_s & & [ux_1]_s \mid \begin{array}{|c|} \hline y_q \\ \hline \vdots \\ \hline y_1 \\ \hline x & z \\ \hline \end{array} \mid [y'v']_s \\ & & \searrow \beta^* & & \searrow \gamma \\ & & [ux_1]_s \mid \begin{array}{|c|} \hline y_q \\ \hline \vdots \\ \hline x & y_i \\ \hline y_{i+1} \\ \hline \vdots \\ \hline y_1 \\ \hline y \\ \hline z \\ \hline \end{array} \mid [y'v']_s & \xleftarrow{\quad} & [ux_1]_s \mid \begin{array}{|c|} \hline y_q \\ \hline \vdots \\ \hline x & y_i \\ \hline \vdots \\ \hline y_1 \\ \hline y \\ \hline z \\ \hline \end{array} \mid [y'v']_s \end{array}$$

Case 3. $x < z < x_1$ and $y \leq y_1$. Suppose that $u = u'x'x_p \dots x_1$ such that $x' \leq x_p$ and $x_p > \dots > x_1$.

$$\begin{array}{ccc} ux_1zxy_1v & \xrightarrow{\quad} & ux_1xzy_1v \\ [w']_s \downarrow & & \downarrow [w]_s \\ [u'x']_s \mid \begin{array}{|c|} \hline x & y \\ \hline z & \\ \hline x_1 \\ \hline \vdots \\ \hline x_p \\ \hline \end{array} \mid [y_1v]_s & \xleftarrow{\quad} & [u'x']_s \mid \begin{array}{|c|} \hline y \\ \hline x & z \\ \hline x_1 \\ \hline \vdots \\ \hline x_p \\ \hline \end{array} \mid [y_1v]_s \end{array}$$

The case $x < z < x_1$ and $y > y_1$ is studied in the same way.

Case 4. $x < x_1 \leq z$ and $y > y_1$. We study similarly the case $x < x_1 \leq z$ and $y \leq y_1$. Suppose that $u = u'x'x_p \dots x_1$ and $v = y_2 \dots y_q y'v'$ such that $x' \leq x_p > \dots > x_1$ and $y_1 > \dots > y_q \leq y'$.

$$\begin{array}{ccc}
 ux_1zxxy_1v & \xrightarrow{\quad\quad\quad} & ux_1xzyy_1v \\
 \downarrow [w']_s & & \downarrow [w]_s \\
 [u'x']_s \mid \begin{array}{c} \boxed{y_q} \\ \vdots \\ \boxed{y_1} \\ \boxed{x} \quad \boxed{y} \\ \boxed{x_1} \quad \boxed{z} \\ \vdots \\ \boxed{x_p} \end{array} \mid [y'v']_s & & [u'x']_s \mid \begin{array}{c} \boxed{y_q} \\ \vdots \\ \boxed{y_1} \\ \boxed{x} \\ \boxed{z} \\ \boxed{x_1} \\ \vdots \\ \boxed{x_p} \end{array} \mid [y'v']_s \\
 \downarrow \beta & \xrightarrow{\delta^\beta} & \downarrow \beta^* \\
 [u'x']_s \mid \begin{array}{c} \boxed{y_q} \\ \vdots \\ \boxed{y_1} \\ \boxed{x} \quad \boxed{z} \\ \boxed{x_1} \\ \vdots \\ \boxed{x_p} \end{array} \mid [y'v']_s & & [u'x']_s \mid \begin{array}{c} \boxed{y_q} \\ \vdots \\ \boxed{y_1} \\ \boxed{x} \\ \boxed{z} \\ \boxed{x_1} \\ \vdots \\ \boxed{x_p} \end{array} \mid [y'v']_s
 \end{array}$$

Suppose now that $w = uy_1yzxx_1v$ and $w' = uy_1yxzx_1v$, for all $1 \leq x < y \leq z \leq n$, $u, v \in [n]^*$ and $x_1, y_1 \in [n]$, and show that $[w]_s \approx_{\mathcal{F}\mathcal{S}_n} [w']_s$. If $y_1 \leq y$ and $x \leq x_1$, then

$$\begin{array}{ccc}
 uy_1yzxx_1v & \xrightarrow{\quad\quad\quad} & uy_1yxzx_1v \\
 [w]_s \downarrow & & \downarrow [w']_s \\
 [uy_1]_s \mid \begin{array}{c} \boxed{x} \\ \boxed{y} \quad \boxed{z} \end{array} \mid [x_1v]_s & \xrightarrow{\beta} & [uy_1]_s \mid \begin{array}{c} \boxed{x} \\ \boxed{z} \\ \boxed{y} \end{array} \mid [x_1v]_s
 \end{array}$$

The cases $(y_1 > y$ and $x > x_1)$, $(y_1 \leq y$ and $x > x_1)$ and $(y_1 > y$ and $x \leq x_1)$ are studied similarly. \square

As a consequence of Theorem 3.2.3 and Theorem 3.3.1, we recover that the set of Young tableaux satisfies the cross-section property for the plactic monoid, we also deduce the commutation of Schensted's left and right insertion algorithm and that the rectification map defines a surjective morphism of monoids between the sets $d\mathbb{S}_n^r$ and \mathbb{Y}_n^r equipped with the insertion products $\star_{I_n^r}$ and $\star_{S_n^r}$.

3.3.2. Corollary (cross-section property). *The following conditions hold*

- i) for all w, w' in $[n]^*$, we have $w \approx_{\mathcal{P}_n} w'$ if and only if $C_{\mathbb{Y}_n^r}(w) = C_{\mathbb{Y}_n^r}(w')$,
- ii) the equality $C_{\mathbb{Y}_n^r}(R_{SW}(d)) = C_{\mathbb{Y}_n^r}(R_{SW}(\pi_{tq}(d)))$ holds in Yt_n , for any d in $d\text{Sk}_n$.

Proof. Prove Condition i). Consider two words w and w' over $[n]$. If $w \approx_{\mathcal{P}_n} w'$, then by Theorem 3.3.1 we have $\text{Nf}([w]_s, \mathcal{F}\mathcal{S}_n) = \text{Nf}([w']_s, \mathcal{F}\mathcal{S}_n)$ and thus $C_{\mathbb{Y}_n^r}(w) = C_{\mathbb{Y}_n^r}(w')$ by Theorem 3.2.3. Suppose now that $C_{\mathbb{Y}_n^r}(w) = C_{\mathbb{Y}_n^r}(w')$. Following Theorem 3.2.3, we have $\text{Nf}([w]_s, \mathcal{F}\mathcal{S}_n) = \text{Nf}([w']_s, \mathcal{F}\mathcal{S}_n)$, and then $w \approx_{\mathcal{P}_n} w'$ by Theorem 3.3.1.

Prove Condition ii). Following Theorem 3.2.3, for any d in $d\text{Sk}_n$, we have $\pi_{tq}(d) = C_{\mathbb{Y}_n^r}(R_{SW}(d))$, showing the claim. \square

3. Convergence of the jeu de taquin

3.3.3. Corollary (commutation of insertion algorithms). **i)** For all d in dSk_n and x in $[n]$, we have $\pi_{tq}(d \leftarrow_{I_r^a} x) = \pi_{tq}(d) \leftarrow_{S_r} x$ and $\pi_{tq}(x \rightsquigarrow_{I_r^a} d) = x \rightsquigarrow_{S_l} \pi_{tq}(d)$.

ii) The insertion algorithms S_r and S_l commute, that is the following equality

$$y \rightsquigarrow_{S_l} (t \leftarrow_{S_r} x) = (y \rightsquigarrow_{S_l} t) \leftarrow_{S_r} x$$

holds in Yt_n , for all t in Yt_n and x in $[n]$.

Proof. Prove Condition **i)**. By Theorem 3.3.1, we have $R_{SW}(\pi_{tq}(d \leftarrow_{I_r^a} x)) \approx_{\mathbb{P}_n} R_{SW}(d \leftarrow_{I_r^a} x)$. Moreover, we have $R_{SW}(d \leftarrow_{I_r^a} x) = R_{SW}(d)x$, hence $R_{SW}(\pi_{tq}(d \leftarrow_{I_r^a} x)) \approx_{\mathbb{P}_n} R_{SW}(d)x$. On the other hand, the following equalities holds in Yt_n :

$$\pi_{tq}(d) \leftarrow_{S_r} x = C_{\mathbb{Y}_n^r}(R_{SW}(\pi_{tq}(d))x) = \text{Nf}([R_{SW}(\pi_{tq}(d))x]_s, \mathcal{F}S_n).$$

Then $\pi_{tq}(d) \leftarrow_{S_r} x \approx_{\mathcal{F}S_n} [R_{SW}(\pi_{tq}(d))x]_s$, and thus by Theorem 3.3.1, we deduce

$$R_{SW}(\pi_{tq}(d) \leftarrow_{S_r} x) \approx_{\mathbb{P}_n} R_{SW}(\pi_{tq}(d))x \approx_{\mathbb{P}_n} R_{SW}(d)x,$$

showing that $R_{SW}(\pi_{tq}(d \leftarrow_{I_r^a} x)) \approx_{\mathbb{P}_n} R_{SW}(\pi_{tq}(d) \leftarrow_{S_r} x)$. Finally, following the cross-section property, we obtain $\pi_{tq}(d \leftarrow_{I_r^a} x) = \pi_{tq}(d) \leftarrow_{I_r^a} x$. Similarly, we show that $\pi_{tq}(x \rightsquigarrow_{I_r^a} d) = x \rightsquigarrow_{S_l} \pi_{tq}(d)$.

Prove Condition **ii)**. Following Condition **i)**, we have

$$\pi_{tq}(y \rightsquigarrow_{I_r^a} (d \leftarrow_{I_r^a} x)) = y \rightsquigarrow_{S_l} (\pi_{tq}(d \leftarrow_{I_r^a} x)) = y \rightsquigarrow_{S_l} (\pi_{tq}(d) \leftarrow_{S_r} x),$$

and

$$\pi_{tq}(y \rightsquigarrow_{I_r^a} (d \leftarrow_{I_r^a} x)) = \pi_{tq}((y \rightsquigarrow_{I_r^a} d) \leftarrow_{S_r} x) = (y \rightsquigarrow_{S_l} \pi_{tq}(d)) \leftarrow_{S_r} x,$$

for all d in D and x, y in $[n]$. By commutation of I_r^a and I_r^a , we deduce the following equality

$$y \rightsquigarrow_{S_l} (\pi_{tq}(d) \leftarrow_{S_r} x) = (y \rightsquigarrow_{S_l} \pi_{tq}(d)) \leftarrow_{S_r} x.$$

The map π_{tq} being surjective, we deduce that S_r and S_l commute. \square

3.3.4. Corollary (morphism of monoids). The map π_{tq} induces a morphism of monoids between $(\text{dSk}_n^r, \star_{I_r^a})$ and $(\mathbb{Y}_n^r, \star_{S_r})$.

Proof. Following Condition **i)** of Corollary 3.3.3, we first prove that $\pi_{tq}(d \leftarrow_{I_r^a} u) = \pi_{tq}(d) \leftarrow_{S_r} u$, for all d in dSk_n and u in $[n]^*$ by induction on $|u|$. Suppose the equality holds when $|u| = k - 1$, then for y in $[n]$ we have

$$\begin{aligned} \pi_{tq}(d \leftarrow_{I_r^a} uy) &= \pi_{tq}((d \leftarrow_{I_r^a} u) \leftarrow_{I_r^a} y) = \pi_{tq}((d \leftarrow_{I_r^a} u)) \leftarrow_{S_r} y \\ &= (\pi_{tq}(d) \leftarrow_{S_r} u) \leftarrow_{S_r} y = \pi_{tq}(d) \leftarrow_{S_r} uy. \end{aligned}$$

In an other hand, following Condition **ii)** of Corollary 3.3.2, we have $C_{\mathbb{Y}_n^r}(R_{SW}(d')) = C_{\mathbb{Y}_n^r}(R_{SW}(\pi_{tq}(d')))$, for all d' in dSk_n , hence $C_{\mathbb{Y}_n^r}(R_{SW}(\pi_{tq}(d))R_{SW}(d')) = C_{\mathbb{Y}_n^r}(R_{SW}(\pi_{tq}(d))R_{SW}(\pi_{tq}(d')))$. As a consequence, we have

$$\begin{aligned} \pi_{tq}(d \star_{I_r^a} d') &= \pi_{tq}(d \leftarrow_{I_r^a} R_{SW}(d')) = \pi_{tq}(d) \leftarrow_{S_r} R_{SW}(d') \\ &= \pi_{tq}(d) \leftarrow_{S_r} R_{SW}(\pi_{tq}(d')) = \pi_{tq}(d) \star_{S_r} \pi_{tq}(d'), \end{aligned}$$

for all d, d' in dSk_n , showing the claim. \square

3.3.5. Remark. Note that Schensted's insertion algorithms are related to the jeu de taquin by the following formulas

$$t \leftarrow_{S_r} x = \pi_{tq}([R_{SW}(t)]_s \leftarrow_{I_r^a} x), \quad x \rightsquigarrow_{S_l} t = \pi_{tq}(x \rightsquigarrow_{I_l^a} [R_{SW}(t)]_s),$$

for all t in Yt_n and x in $[n]$. Note also that the associativity of \star_{S_r} is also deduced from the morphism π_{tq} . Indeed, for all t in Yt_n , and x in $[n]$, we have $t \leftarrow_{S_r} x = \pi_{tq}(t|_1 \overline{x})$, and thus $t \star_{S_r} t' = \pi_{tq}(t|_s t')$, for all $t, t' \in Yt_n$. By Theorem 3.2.3, we obtain

$$(t \star_{S_r} t') \star_{S_r} t'' = \pi_{tq}(t|_s t'|_s t'') = t \star_{S_r} (t' \star_{S_r} t''),$$

for any $t, t', t'' \in Yt_n$.

4. CONCLUSION AND PERSPECTIVES

In this article, we have introduced the notion of string of columns rewriting system as rewriting systems over glued sequences of columns. This gives a rewriting framework to prove the confluence of Schützenberger's jeu de taquin algorithm defined on the structure of tableaux. Our construction leads us to formulate several perspectives:

- In [10], we make explicit the relations among the relations of the Knuth relations for the plactic monoid of type A. We expect that the rewriting presentation of the jeu de taquin introduced in this article could make explicit the relations among the relations for the cross-section property on sequence of columns for the plactic monoid of type A. Such a study of the relations among the relations for the presentations of a monoid constitutes the first step in an explicit construction of a cofibrant approximation of the monoid in the category of $(\omega, 1)$ -categories and of actions of the monoid on categories, see [5, 6, 10, 11].
- Schützenberger's jeu de taquin gives a proof of the Littlewood–Richardson rule which is a combinatorial description of the coefficients that arise when decomposing a product of two Schur polynomials as a linear combination of other Schur polynomials. In particular, these coefficients count certain types of skew tableaux that are rectified by the jeu de taquin to Young tableaux. We expect that these coefficients could be also described by a rewriting approach using a zigzag sequences of reductions from certain types of skew tableaux to their normal forms with respect to the rewriting presentation of the jeu de taquin introduced in this article.
- The construction applied in this article on the jeu de taquin could be also applied on similar algorithms defined on other structures of tableaux, [9, 15, 16, 20, 23, 24]. In particular, we expect constructive proofs by rewriting of the properties relating these algorithms to the plactic monoids of classical types and the super plactic monoid which are respectively related to the representations of the finite dimensional semisimple Lie algebras of classical types and the general Lie superalgebra.

REFERENCES

- [1] David Aldous and Persi Diaconis. Longest increasing subsequences: from patience sorting to the Baik-Deift-Johansson theorem. *Bull. Amer. Math. Soc. (N.S.)*, 36(4):413–432, 1999.

REFERENCES

- [2] Alan J. Cain, Robert D. Gray, and António Malheiro. Crystal monoids & crystal bases: rewriting systems and biautomatic structures for plactic monoids of types A_n , B_n , C_n , D_n , and G_2 . *J. Combin. Theory Ser. A*, 162:406–466, 2019.
- [3] Neil O Connell. Conditioned random walks and the RSK correspondence. *Journal of Physics A: Mathematical and General*, 36(12):3049–3066, mar 2003.
- [4] William Fulton. *Young tableaux*, volume 35 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 1997. With applications to representation theory and geometry.
- [5] Stéphane Gaussent, Yves Guiraud, and Philippe Malbos. Coherent presentations of Artin monoids. *Compos. Math.*, 151(5):957–998, 2015.
- [6] Yves Guiraud and Philippe Malbos. Higher-dimensional normalisation strategies for acyclicity. *Adv. Math.*, 231(3-4):2294–2351, 2012.
- [7] Yves Guiraud and Philippe Malbos. Polygraphs of finite derivation type. *Math. Structures Comput. Sci.*, 28(2):155–201, 2018.
- [8] Nohra Hage. Finite convergent presentation of plactic monoid for type C. *Internat. J. Algebra Comput.*, 25(8):1239–1263, 2015.
- [9] Nohra Hage. Super jeu de taquin and combinatorics of super tableaux of type A. arxiv:2105.07819, May 2021.
- [10] Nohra Hage and Philippe Malbos. Knuth’s coherent presentations of plactic monoids of type A. *Algebr. Represent. Theory*, 20(5):1259–1288, 2017.
- [11] Nohra Hage and Philippe Malbos. Super Chinese syzygies by insertions. *Semigroup Forum*, 104(1):88–108, 2022.
- [12] Masaki Kashiwara and Toshiki Nakashima. Crystal graphs for representations of the q -analogue of classical Lie algebras. *J. Algebra*, 165(2):295–345, 1994.
- [13] Donald E. Knuth. Permutations, matrices, and generalized Young tableaux. *Pacific J. Math.*, 34:709–727, 1970.
- [14] Alain Lascoux and Marcel-P. Schützenberger. Le monoïde plaxique. In *Noncommutative structures in algebra and geometric combinatorics (Naples, 1978)*, volume 109 of *Quad. “Ricerca Sci.”*, pages 129–156. CNR, Rome, 1981.
- [15] Cédric Lecouvey. Schensted-type correspondence, plactic monoid, and jeu de taquin for type C_n . *J. Algebra*, 247(2):295–331, 2002.
- [16] Cedric Lecouvey. Schensted-type correspondences and plactic monoids for types B_n and D_n . *J. Algebraic Combin.*, 18(2):99–133, 2003.
- [17] M. Lothaire. *Algebraic combinatorics on words*, volume 90 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2002.
- [18] Maxwell Newman. On theories with a combinatorial definition of “equivalence”. *Ann. of Math. (2)*, 43(2):223–243, 1942.
- [19] Jean-Christophe Novelli. On the hypoplactic monoid. volume 217, pages 315–336. 2000. Formal power series and algebraic combinatorics (Vienna, 1997).

REFERENCES

- [20] Dan Romik and Piotr Śniady. Jeu de taquin dynamics on infinite Young tableaux and second class particles. *The Annals of Probability*, 43(2):682–737, 2015.
- [21] C. Schensted. Longest increasing and decreasing subsequences. *Canad. J. Math.*, 13:179–191, 1961.
- [22] M.-P. Schützenberger. La correspondance de Robinson. pages 59–113. *Lecture Notes in Math.*, Vol. 579, 1977.
- [23] Vasu Tewari. Backward jeu de taquin slides for composition tableaux and a noncommutative pieri rule. *Electron. J. Comb.*, 22:P1.42, 2015.
- [24] Hugh Thomas and Alexander Yong. A jeu de taquin theory for increasing tableaux, with applications to k -theoretic schubert calculus. *Algebra and Number Theory*, 3(2):121–148, 2009.
- [25] Marc A. A. van Leeuwen. The Littlewood-Richardson rule, and related combinatorics. In *Interaction of combinatorics and representation theory*, volume 11 of *MSJ Mem.*, pages 95–145. Math. Soc. Japan, Tokyo, 2001.
- [26] Alfred Young. On Quantitative Substitutional Analysis. *Proc. London Math. Soc. (2)*, 28(4):255–292, 1928.

NOHRA HAGE
nohra.hage@univ-catholille.fr
Faculté de Gestion, Economie & Sciences (FGES),
Université Catholique de Lille,
60 bd Vauban,
CS 40109, 59016 Lille Cedex, France

PHILIPPE MALBOS
malbos@math.univ-lyon1.fr
Univ Lyon, Université Claude Bernard Lyon 1
CNRS UMR 5208, Institut Camille Jordan
43 blvd. du 11 novembre 1918
F-69622 Villeurbanne cedex, France