

AN INSERTION ALGORITHM FOR CATABOLIZABILITY

JONAH BLASIAK

ABSTRACT. Our recent work in [1] exhibits a canonical basis of the Garsia-Procesi module R_λ with cells labeled by standard tableaux of catabolizability $\trianglerighteq \lambda$. Through our study of the Kazhdan-Lusztig preorder on this basis, we found a way to transform a standard word labeling a basis element into a word inserting to the unique tableau of shape λ . This led to an algorithm that computes the catabolizability of the insertion tableau of a standard word. We deduce from this a characterization of catabolizability as the statistic on words invariant under Knuth transformations, certain (co)rotations, and a new set of transformations we call catabolism transformations. We further deduce a Greene's Theorem-like characterization of catabolizability and a result about how cocyclage changes catabolizability, strengthening a similar result in [8].

1. INTRODUCTION

An important and difficult class of problems in algebraic combinatorics is to give positive combinatorial formulae for the characters of graded representations of the symmetric group \mathcal{S}_n . Lascoux, in part with Schützenberger, [7], [6] give beautiful combinatorial descriptions of the Garsia-Procesi modules (discussed below) using operations on tableaux called cyclage and catabolism.

Cyclage is the operation on tableaux induced by cyclic permutations of words under the correspondence of words with tableaux given by the RSK insertion algorithm. Cyclage makes the set of tableaux with fixed content into a graded poset, with rank function given by cocharge (see §2.2, §2.3). A catabolism is the operation of cyclaging all the entries of a tableau T not in the first row of T . The catabolizability of a standard tableau T , denoted $\text{ctype}(T)$, is a partition associated to the sequence of catabolisms taking T to the single-row tableau (see §2.4). Generalizations and variations of catabolism are also described in [8] and [5], many of which remain quite mysterious. These operations are easy to experiment with and make conjectures about, but proving such conjectures is often extremely difficult. Here we show that the basic catabolism of [6] can be simplified significantly. We have yet to find a similar simplification for the set of R -catabolizable tableaux defined in [8].

In [6], Lascoux defines a set of standard tableaux $\mathcal{A}(\lambda)$ called an atom, for each partition λ , in terms of embeddings of the cyclage poset on semistandard tableaux into the cyclage poset on standard tableaux. The atoms are studied in detail and it is asserted that $\mathcal{A}(\lambda)$ is also the set of standard tableaux with catabolizability λ . Lascoux proves, with some proofs and clarification supplied by Shimozono-Weyman [8], the following combinatorial formula for a variant of Hall-Littlewood polynomials $\tilde{H}_\lambda(t)$:

$$(1) \quad \tilde{H}_\lambda(t) = \sum_{\substack{T \in \text{SYT} \\ \text{ctype}(T) \supseteq \lambda}} t^{\text{cocharge}(T)} s_{\text{sh}(T)},$$

where SYT is the set of standard Young tableaux, \supseteq is dominance order on partitions, and $s_{\text{sh}(T)}$ is the Schur function corresponding to the shape of T . The symmetric functions $\tilde{H}_\lambda(t)$ with coefficients in $\mathbb{C}[t]$ are called the cocharge variant transformed Hall-Littlewood polynomials in [4], and are related to the cocharge Kostka-Foulkes polynomials $\tilde{K}_{\mu\lambda}(t)$ by $\tilde{H}_\lambda(t) = \sum_{\mu} \tilde{K}_{\mu\lambda}(t) s_{\mu}$.

Our work in [1] begins with the goal of strengthening the connection between this combinatorial formula and the following representation theoretic interpretation of Hall-Littlewood polynomials. The ring of coinvariants $R_{1^n} = \mathbb{C}[y_1, \dots, y_n]/(e_1, \dots, e_n)$, thought of as a $\mathbb{C}\mathcal{S}_n$ -module with \mathcal{S}_n acting by permuting the variables, is a graded version of the regular representation. It has the Garsia-Procesi modules R_λ as quotients (see [3]). The work of Hotta-Springer establishes that the Frobenius series $\mathcal{F}_{R_\lambda}(t)$ of R_λ is equal to $\tilde{H}_\lambda(t)$ (see [4]).

In [1] we exhibit a q -analogue \mathcal{R}_{1^n} of the ring of coinvariants that is endowed with a canonical basis and possesses q -analogues \mathcal{R}_λ of the R_λ as cellular quotients. The elements of the canonical basis are in bijection with standard words (permutations of $1, \dots, n$), the cells in bijection with SYT, and there is a natural grading on the cells that corresponds to cocharge under this bijection. Tableaux of shape λ do not appear in degree less than $n(\lambda) = \sum_i \lambda_i(i-1)$ and there is a unique occurrence of a tableau of shape λ in this degree. We call this tableau the *Garnir tableau of shape λ* or Z_λ^* . In our investigations of the \mathcal{R}_λ , we found a way to go from any standard word w to a word inserting to the Garnir tableau of shape $\text{ctype}(P(w))$ by a sequence of relations in the Kazhdan-Lusztig preorder.

Following this sequence of relations gives an algorithm for computing $\text{ctype}(P(w))$ for any standard word w . This allows us to characterize catabolizability as the statistic on words invariant under Knuth transformations, non-zero (co)rotations (defined in §2.3), and catabolism transformations (defined in §3.1) and equal to λ on the reading word of Z_λ^* . We then use this to prove a Greene's Theorem-like characterization of catabolizability.

After reviewing the definitions of cocyclage and catabolism in §2, we present this algorithm and its corollaries in §3.

2. COCYCLAGE AND CATABOLISM

We recall the notions of cocyclage and catabolizability as defined in [7], [6], and [8], but restrict to the special case of standard tableaux and words.

2.1. We will use the following notational conventions in this paper. Tableaux are drawn using English notation, so that entries strictly increase from north to south along columns and weakly increase from west to east along rows. The notation $|z|$ denotes the length of the word z . We let $P(z)$ denote the insertion tableau produced by the Robinson-Schensted-Knuth (RSK) algorithm applied to the word z . For a tableau T , $|T|$ is the number of squares in T and $\text{sh}(T)$ is its shape. The symbols u, v , and w will always denote standard words, and y, z will denote words that are not necessarily standard. The notation $[a, b]$ for $a, b \in \mathbb{Z}$ denotes the set $\{i : a \leq i \leq b\}$ and $[n] := [1, n]$.

2.2. The *cocharge labeling* of a word w , denoted w^{cc} , is a (non-standard) word of the same length as w , and its numbers are thought of as labels of the numbers of w . It is obtained from w by reading the numbers of w in increasing order, labeling the 1 of w with a 0, and if the i of w is labeled by k , then labeling the $i + 1$ of w with a k (resp. $k + 1$) if $i + 1$ appears to the right (resp. left) of i in w . An example of a standard word and its cocharge labeling is

$$\begin{aligned} w &= 1\ 6\ 8\ 4\ 2\ 9\ 5\ 7\ 3 \\ w^{\text{cc}} &= 0\ 2\ 3\ 1\ 0\ 3\ 1\ 2\ 0. \end{aligned}$$

The sum of the numbers in the cocharge labeling of w is the *cocharge* of w or $\text{cocharge}(w)$. We also set $\text{cocharge}(w^{\text{cc}}) = \text{cocharge}(w)$.

Let s_i be the simple reflection of \mathcal{S}_n that transposes i and $i + 1$. Thinking of a standard word $w = w_1 \cdots w_n$ as being the map $w : [n] \rightarrow [n]$, $i \mapsto w_i$, we can act on w on the right by \mathcal{S}_n ; then ws_i is the word obtained from w by swapping the numbers in positions i and $i + 1$. We can also act on cocharge labelings with this same right action, however this does not always result in the cocharge labeling of a standard word. It is not hard to see that a standard word can be recovered from its cocharge labeling. A word z with c_0 0's, c_1 1's, \dots , is the cocharge labeling of some standard word if and only if there is an l such that $c_i > 0$ for $i \in [0, l]$, $c_i = 0$ for $i > l$, and some i appears to the left of some $i - 1$ for all $i \in [l]$. The following are easily seen

to be equivalent.

- (2)
$$\begin{aligned} (i) \quad & w^{\text{cc}} s_i = (ws_i)^{\text{cc}}. \\ (ii) \quad & w^{\text{cc}} \text{ and } (ws_i)^{\text{cc}} \text{ have the same content.} \\ (iii) \quad & \text{cocharge}(w) = \text{cocharge}(ws_i). \\ (iv) \quad & |w_i - w_{i+1}| \neq 1. \end{aligned}$$

If any (all) of these holds, then the transformation $w \rightsquigarrow ws_i$ is *cocharge-preserving*.

2.3. Cyclage and cocyclage are the operations on tableaux induced by cyclic permutations of words. We use the terminology rotation and corotation to distinguish certain cyclic permutations of words.

For a word w and number $a \neq 1$, aw (resp. wa) is a *corotation* (resp. *rotation*) of wa (resp. of aw). It is a *non-zero corotation* (resp. *rotation*) of wa (resp. of aw) if, in the cocharge labeling of wa , a is labeled with a number greater than 0. Similarly, define *zero (co)rotations* for the case a is labeled with a 0.

(Co)rotations respect cocharge labeling in the following way: v is a corotation of w if and only if $v^{\text{cc}} = a + 1 y$ and $w^{\text{cc}} = ya$ for y a word and a a number. For example,

$$\begin{aligned} w &= 1\ 6\ 8\ 4\ 2\ 9\ 5\ 7\ 3 \\ v &= 3\ 1\ 6\ 8\ 4\ 2\ 9\ 5\ 7 \\ w^{\text{cc}} &= 0\ 2\ 3\ 1\ 0\ 3\ 1\ 2\ 0 \\ v^{\text{cc}} &= 1\ 0\ 2\ 3\ 1\ 0\ 3\ 1\ 2. \end{aligned}$$

There is a *cocyclage* from the tableau T to the tableau T' , written $T \xrightarrow{\text{cc}} T'$, if there exist words u, v such that v is the corotation of u and $P(u) = T$ and $P(v) = T'$. The *cocyclage poset* is the poset on the set of SYT generated by the relation $\xrightarrow{\text{cc}}$.

Define the cocharge labeling T^{cc} of a tableau T to be $P(\text{rowword}(T)^{\text{cc}})$, and $\text{cocharge}(T)$ to be the sum of the entries in T^{cc} . Here $\text{rowword}(T)$ denotes the row reading word of T . The tableau T^{cc} is also $P(w^{\text{cc}})$ for any w inserting to T . This follows from the fact that Knuth transformations do not change left descent sets. A cocyclage $T \xrightarrow{\text{cc}} T'$ is a *non-zero (resp. zero) cocyclage* if any (equivalently, every) corotation inducing it is non-zero (resp. zero).

The *cyclage poset* is the poset dual to the cocyclage poset, i.e., the poset obtained by reversing all relations.

Theorem 2.1 ([7]). *The cyclage poset is graded, with rank function given by cocharge.*

This theorem holds more generally for the cyclage poset on the set of semistandard tableaux with fixed content. The theorem is quite easy in the standard tableaux case.

2.4. Lascoux defines [6] an operation on tableaux called catabolism to describe the atoms $\mathcal{A}(\lambda)$ mentioned in the introduction. We recall Lascoux's definition and some of its variations from [8].

Definition 2.2. Let Z_λ be the superstandard tableau of shape λ with 0's in the first row, 1's in the second row, etc., and let Z_λ^* be the standard tableau such that $Z_\lambda^{*cc} = Z_\lambda$.

The tableau Z_λ^* is the Garnir tableau mentioned in the introduction. We warn the reader that in [1] the Garnir tableau is actually a tableau with entries in \mathbb{N} , equal to the entry-wise sum $nZ_\lambda + Z_\lambda^*$, where $n = \lambda$.

Definition 2.3. For a tableau T , if $\lambda \subseteq \text{sh}(T)$, then T_λ is the subtableau of T of shape λ .

Let T be a standard tableau. Write $\text{rowword}(T) = ww'$, with w' the first row of T . The operation $T = P(ww') \rightarrow P(w'w) =: K(T)$ is a *catabolism*. Let $d(U)$, for a standard tableau U , be the largest integer m such that $U_{(m)} = Z_{(m)}^*$, i.e. the largest m such that the first row of U contains $1, 2, \dots, m$. The *catabolizability* of T , denoted $\text{ctype}(T)$ is the sequence

$$(d(T), d(K(T)) - d(T), \dots, d(K^i(T)) - d(K^{i-1}(T)), \dots).$$

The variations of catabolism of [8] are defined in terms of certain slicing operations on tableaux. (This language is also useful for the more general notion of R -catabolizable tableaux defined there.) For a skew tableau T and index r (resp. index c), let $H_r(T) = P(T_n T_s)$ (resp. $V_c(T) = P(T_e T_w)$), where T_n and T_s (resp. T_e and T_w) are the north and south (resp. east and west) subtableaux obtained by slicing T horizontally (resp. vertically) between its r -th and $(r+1)$ -th rows (resp. c -th and $(c+1)$ -th columns). Recall that we are using English notation for tableaux.

If $(m) \subseteq \text{sh}(T)$, then define the m -catabolism (resp. m -column catabolism) of T , notated $\text{Cat}_m(T)$ (resp. $\text{CCat}_m(T)$), to be the tableau $H_1(T - T_{(m)})$ (resp. $V_m(T - T_{(m)})$). For a partition $\lambda \vdash n := |T|$, λ -catabolizability (resp. λ -column catabolizability) is defined inductively as follows: T is λ -(column) catabolizable if $T_{(\lambda_1)}$ contains the λ_1 smallest entries of T and the λ_1 -(column) catabolism of T is $\widehat{\lambda}$ -(column) catabolizable, where $\lambda = (\lambda_1, \widehat{\lambda})$; the empty tableau is \emptyset -(column) catabolizable.

Example 2.4. The following sequence of catabolisms shows that $\text{ctype}(T) = (3, 2, 2, 2)$:

$$T = \begin{array}{|c|c|c|c|c|} \hline 1 & 2 & 3 & 6 & 7 \\ \hline 4 & 5 & 8 & 9 & \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 2 & 3 & 4 & 5 & 8 & 9 \\ \hline 6 & 7 & & & & & \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|c|c|c|} \hline 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 8 & 9 & & & & & \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline \end{array}.$$

The sequence of m -catabolisms which shows T to be $(3, 2, 2, 2)$ -catabolizable is similar:

$$\begin{array}{|c|c|c|c|c|} \hline 1 & 2 & 3 & 6 & 7 \\ \hline 4 & 5 & 8 & 9 & \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|} \hline 4 & 5 & 8 & 9 \\ \hline 6 & 7 & & \\ \hline \end{array} \rightarrow \begin{array}{|c|c|} \hline 6 & 7 \\ \hline 8 & 9 \\ \hline \end{array} \rightarrow \boxed{8 \ 9}.$$

Contrast this with the sequence of m -column catabolisms which shows T to be $(3, 2, 2, 2)$ -column catabolizable:

$$\begin{array}{|c|c|c|c|c|} \hline 1 & 2 & 3 & 6 & 7 \\ \hline 4 & 5 & 8 & 9 & \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|} \hline 4 & 5 & 8 \\ \hline 6 & 7 & \\ \hline 9 & & \\ \hline \end{array} \rightarrow \begin{array}{|c|c|} \hline 6 & 7 \\ \hline 8 & 9 \\ \hline \end{array} \rightarrow \boxed{8 \ 9}.$$

The variations of catabolism are related as follows.

Proposition 2.5 ([8]). *The sequence $\text{ctype}(T)$ is a partition, and T is λ -catabolizable if and only if $\text{ctype}(T) \succeq \lambda$.*

A consequence of this proposition is that catabolizability can be computed by performing the sequence of catabolisms to T in which Cat_m is applied with the largest m such that $T_{(m)} = Z_{(m)}^*$. We define the column catabolizability of T to be the partition obtained by performing the sequence of column catabolisms to T in which CCat_m is applied with the largest m such that $T_{(m)} = Z_{(m)}^*$. With the definitions of λ -(column) catabolizability above, the following proposition is quite tricky.

Proposition 2.6 ([8, Proposition 49]). *A standard tableau is λ -catabolizable if and only if it is λ -column catabolizable.*

We reprove this result in the next section using the catabolism insertion algorithm.

3. CATABOLISM INSERTION

3.1. We now describe the *catabolism insertion algorithm*, which takes a standard word w as input and outputs a partition $F(w)$ that we will show to be equal to the catabolizability of the insertion tableau of w .

Let $\epsilon_i \in \mathbb{Z}^n$ be the standard basis vector with a 1 in its i -th coordinate and 0's elsewhere.

Algorithm 3.1. Let f be the function below, which takes a pair consisting of a (non-standard) word and a partition to another such pair. Let $x = ya$, y a word and a a number.

$$(3) \quad f(x, \nu) = \begin{cases} (y, \nu + \epsilon_{a+1}) & \text{if } \nu + \epsilon_{a+1} \text{ is a partition,} \\ (a + 1 \ y, \nu) & \text{otherwise.} \end{cases}$$

Given the input standard word w , first determine the cocharge labeling z of w .

Next, apply f to (z, \emptyset) repeatedly, obtaining the sequence of pairs $f^{(i)}(z, \emptyset)$, stopping when the word of the pair is empty. Output the partition of this final pair, and denote this output $F(w)$.

The transition from (x, ν) to $f(x, \nu)$ is a *step* of the algorithm. We say that a is *presented* to ν , and in the top case of (3), a is *inserted* into ν , while in the bottom case, a is *corotated*. The step of the algorithm in the top case is an *insertion*, and in the bottom a *corotation*.

It is convenient to think of the ν in the algorithm as the tableau Z_ν , as illustrated by the following example.

Example 3.2. The word $w = 1\ 6\ 8\ 4\ 2\ 9\ 5\ 7\ 3$ has cocharge labeling $z = 0\ 2\ 3\ 1\ 0\ 3\ 1\ 2\ 0$. The sequence of word-partition pairs produced by the algorithm applied to w is

i	$f^{(i)}(z, \emptyset)$	i	$f^{(i)}(z, \emptyset)$
0	023103120 \emptyset	7	44302 $\begin{array}{ c c } \hline 0 & 0 \\ \hline 1 & 1 \\ \hline \end{array}$
1	02310312 $\begin{array}{ c } \hline 0 \\ \hline \end{array}$	8	4430 $\begin{array}{ c c } \hline 0 & 0 \\ \hline 1 & 1 \\ \hline 2 \\ \hline \end{array}$
2	30231031 $\begin{array}{ c } \hline 0 \\ \hline \end{array}$	9	443 $\begin{array}{ c c c } \hline 0 & 0 & 0 \\ \hline 1 & 1 & \\ \hline 2 \\ \hline \end{array}$
3	3023103 $\begin{array}{ c } \hline 0 \\ \hline 1 \\ \hline \end{array}$	10	44 $\begin{array}{ c c c } \hline 0 & 0 & 0 \\ \hline 1 & 1 & \\ \hline 2 \\ \hline 3 \\ \hline \end{array}$
4	4302310 $\begin{array}{ c } \hline 0 \\ \hline 1 \\ \hline \end{array}$	11	4 $\begin{array}{ c c c } \hline 0 & 0 & 0 \\ \hline 1 & 1 & \\ \hline 2 \\ \hline 3 \\ \hline 4 \\ \hline \end{array}$
5	430231 $\begin{array}{ c c } \hline 0 & 0 \\ \hline 1 \\ \hline \end{array}$	12	5 $\begin{array}{ c c c } \hline 0 & 0 & 0 \\ \hline 1 & 1 & \\ \hline 2 \\ \hline 3 \\ \hline 4 \\ \hline \end{array}$
6	43023 $\begin{array}{ c c } \hline 0 & 0 \\ \hline 1 & 1 \\ \hline \end{array}$	13	\emptyset $\begin{array}{ c c c } \hline 0 & 0 & 0 \\ \hline 1 & 1 & \\ \hline 2 \\ \hline 3 \\ \hline 4 \\ \hline 5 \\ \hline \end{array}$

This may be compared to the sequence $\text{rowword}(T), \text{rowword}(K(T)), \dots$, where $T = P(w)$:

$\text{rowword}(K^i(T))$	$\text{rowword}(K^i(T))^{\text{cc}}$	$d(\text{rowword}(K^i(T)))$
6 8 4 5 9 1 2 3 7	2 3 1 1 3 0 0 0 2	3
7 6 8 1 2 3 4 5 9	2 1 2 0 0 0 0 0 2	5
9 7 1 2 3 4 5 6 8	2 1 0 0 0 0 0 0 2	6
8 1 2 3 4 5 6 7 9	1 0 0 0 0 0 0 0 1	7
9 1 2 3 4 5 6 7 8	1 0 0 0 0 0 0 0 0	8
1 2 3 4 5 6 7 8 9	0 0 0 0 0 0 0 0 0	9

Taking differences of the last column yields $\text{ctype}(T) = (3, 2, 1, 1, 1, 1)$.

Definition 3.3. Letting z be the cocharge labeling of w , a *catabolism transformation* of w (or of z) is an operation taking w to ws_i (z to zs_i) if $|z_i - z_{i+1}| > 1$:

$$(4) \quad \cdots z_i z_{i+1} \cdots \longleftrightarrow \cdots z_{i+1} z_i \cdots$$

It is easy to see from (2) (iv) that a catabolism transformation is cocharge-preserving.

Some properties of Algorithm 3.1 are more easily seen from the following variant, which is clearly equivalent to it.

Algorithm 3.4. A step of Algorithm 3.1 from (ya, ν) to $f(ya, \nu)$ is rephrased as follows. Instead of keeping track of a partition ν , keep track of the corresponding superstandard tableau Z_ν . Replace presenting a to ν with column-inserting a into Z_ν . In the insertion case, this produces the same result as in Algorithm 3.1.

The corotation case is broken into three parts, the first of which is this column-insertion; let T be the tableau resulting from this insertion. The tableau T contains Z_ν and $T - Z_\nu$ is a single square containing an a . The corresponding number of $\text{rowword}(T)$ is at least two more than all the numbers to the right of it. The second part performs the sequence of catabolism transformations taking $\text{rowword}(T)$ to $\text{rowword}(Z_\nu) a$. The third part then corotates

$$y \text{ rowword}(Z_\nu) a$$

to obtain

$$a + 1 \ y \ \text{rowword}(Z_\nu),$$

and this word is output as the pair $(a + 1 \ y, Z_\nu)$.

Remark 3.5. In light of Algorithm 3.4, we have chosen to call the algorithms for catabolizability in this paper insertion algorithms (despite the fact that their output is only a partition). Also, the related sign-insertion

algorithm of [1] produces tableaux other than Z_λ -tableaux that encode more data than just a partition.

Remark 3.6. Lascoux's definition of catabolizability is, roughly speaking, a measure of how fast a word can be transformed into the word $1, 2, \dots, n$ by a sequence of rotations and Knuth transformations. The catabolism insertion algorithm, in contrast, finds a sequence of corotations, catabolism transformations, and Knuth transformations from a word to $\text{rowword}(Z_\lambda^*)$ for some λ . Note that there is not always a sequence of corotations and Knuth transformations from w to $\text{rowword}(Z_\lambda^*)$ with $\lambda = \text{ctype}(P(w))$. For example, there is no such sequence from $w = 4\ 3\ 5\ 1\ 2$ to $5\ 4\ 3\ 1\ 2$.

Theorem 3.7. *Algorithm 3.1, with input a standard word u of length n , satisfies:*

- (i) *after every step, the word-partition pair (x, ν) is such that $x \text{ rowword}(Z_\nu)$ is the cocharge labeling of some standard word,*
- (ii) *terminates (in at most $n + \binom{n}{2}$ steps),*
- (iii) *$F(u) = F(v)$ if v is a non-zero corotation of u ,*
- (iv) *$F(u) = F(v)$ if $u \rightsquigarrow v$ is a catabolism transformation,*
- (v) *$F(u) = F(v)$ if $u \rightsquigarrow v$ is a Knuth transformation,*
- (vi) *$F(u) = \text{ctype}(P(u))$.*

Proof. Statement (i) is easy to see from Algorithm 3.4. It realizes each step as a sequence of Knuth transformations, catabolism transformations, or non-zero (co)rotations on the word $x \text{ rowword}(Z_\nu)$, all of which could be applied by converting back to a standard word, performing the operation, and then taking the cocharge labeling.

The algorithm terminates in at most $n + \binom{n}{2}$ steps because for any step from (x, ν) to $(x^*, \nu^*) := f(x, \nu)$, either

$$|x^*| = |x| - 1 \quad \text{or} \\ \text{cocharge}(x^* \text{ rowword}(Z_{\nu^*})) = \text{cocharge}(x \text{ rowword}(Z_\nu)) + 1;$$

cocharge never exceeds $\binom{n}{2}$.

Statement (iii) is clear.

For (iv), suppose that $u^{\text{cc}} \rightsquigarrow v^{\text{cc}}$ is the catabolism transformation

$$(5) \quad \dots ab \dots \rightsquigarrow \dots ba \dots$$

Whether a is corotated or inserted does not depend on what happens to b and similarly whether b is corotated or inserted does not depend on what happens to a . Therefore, after a and b are presented, the resulting pair (x, ν) is the same for the algorithm applied to u and applied to v , unless a and b are both corotated. In this case, the result follows by induction

since we can apply the same argument to the catabolism transformation $\cdots a + 1 b + 1 \cdots \rightsquigarrow \cdots b + 1 a + 1 \cdots$.

To show (v), we prove the slightly stronger statement that, after any step of the algorithm, the pair (x, ν) can be replaced by (x', ν) with $x \rightsquigarrow x'$ a Knuth transformation without changing the output. We may assume that $x \rightsquigarrow x'$ is a Knuth transformation in the last three numbers of x and is written as

$$(6) \quad \cdots bac \rightsquigarrow \cdots bca,$$

where $a < b \leq c$. Given (iv), we may further assume that $b = c$, $a = b - 1$ (we must also check the case $\cdots acb \rightsquigarrow \cdots cab$, where $b = a$ and $c = b + 1$, but this is similar). If $a > 0$ and $\nu_a = \nu_{a+1} = \nu_b = \nu_{b+1}$, then a, b , and c are corotated and the result follows by induction from the Knuth transformation

$$(7) \quad b + 1 a + 1 c + 1 \cdots \rightsquigarrow b + 1 c + 1 a + 1 \cdots.$$

Otherwise, one checks that $f^{(3)}(x, \nu) = f^{(3)}(x', \nu)$.

Given (v), after any step, we are free to replace x with something Knuth equivalent to it without changing the final output. Also, after any step, we may run the algorithm on anything Knuth equivalent to x rowword(Z_ν) and get the same output. Thus for (vi), run the algorithm on rowword($P(u)$). The first steps of the algorithm corotate the numbers in the first row of $P(u)^{\text{cc}}$ that are not 0. The next m steps are insertions of 0's, where m is the number of 0's in $P(u)^{\text{cc}}$, also the largest integer such that $P(u)_{(m)} = Z_{(m)}^*$. Let (x, ν) be the pair at this stage, x' be the result of subtracting 1 from all numbers in x , and v be the standard word with $v^{\text{cc}} = x'$. Statement (vi) is proved by verifying the equalities

$$(8) \quad F(u) = (m, F(v)) = (m, \text{ctype}(P(v))) = \text{ctype}(P(u)).$$

The leftmost equality holds because the word x contains at most m 1's, and the middle equality is the inductive statement $F(v) = \text{ctype}(P(v))$. The rightmost equality holds because the standard tableau $P(v)$ is the same as the result of subtracting m from all entries in $\text{Cat}_m(P(u))$ and $\nu = (m)$. \square

From now on, we write $\text{ctype}(u)$ for $\text{ctype}(P(u))$. From the theorem and remarks in the proof of (i), we obtain the following.

Corollary 3.8. *Catabolizability is characterized as the statistic on standard words that is invariant under non-zero (co)rotations, catabolism transformations, and Knuth transformations, and satisfies*

$$\text{ctype}(\text{rowword}(Z_\lambda^*)) = \lambda.$$

A similar proof to that of (vi) gives

Corollary 3.9. *The catabolizability of a tableau T equals the column catabolizability of T .*

Note that this is not exactly the same as the claim that T is λ -catabolizable if and only if it is λ -column catabolizable (Proposition 2.6). To prove this, it is convenient to modify the catabolism insertion algorithm, which we do in the next subsection.

3.2. We present one more variant of the catabolism insertion algorithm that determines whether a tableau is λ -catabolizable rather than its catabolizability.

Algorithm 3.10. This takes a standard word w of length n and a partition λ of n as input. It is the same as Algorithm 3.1 except that the partition ν of the word-partition pair is forced to satisfy $\nu_i \leq \lambda_i$ for all i . Precisely, f is replaced by

$$(9) \quad f_\lambda(ya, \nu) = \begin{cases} (y, \nu + \epsilon_{a+1}) & \text{if } \nu + \epsilon_{a+1} \text{ is a partition and } \nu_{a+1} + 1 \leq \lambda_{a+1} \\ (a+1 \ y, \nu) & \text{otherwise.} \end{cases}$$

This algorithm repeatedly applies f_λ to $(w^{\text{cc}}, \emptyset)$ until either a number a is presented such that $\lambda_{a+1} = 0$, or the word of the pair is empty. The algorithm outputs false if the former occurs and true if the latter occurs.

Definition 3.11. An *ascent* of w (or of z) is a transformation of the form $z \rightsquigarrow zs_i$ provided it is cocharge-preserving and $z_i > z_{i+1}$.

Theorem 3.12. *Algorithm 3.10, with input a standard word u of length n and a partition λ of n , outputs true if and only if $P(u)$ is λ -catabolizable.*

Proof. The proof is essentially the same as the proof of Theorem 3.7. The most significant difference is for the proof of (i). The only case in which Algorithms 3.1 and 3.10 differ is when a is presented and $\nu + \epsilon_{a+1}$ is a partition but $\nu_{a+1} + 1 > \lambda_{a+1}$. Such a corotation step of Algorithm 3.10 can also be broken into several steps as in Algorithm 3.4. If $\lambda_{a+1} > 0$, then this step can be broken into a sequence of Knuth transformations, ascents, and catabolism transformations followed by a corotation. These operations could be applied by converting back to a standard word, performing the operation, and then taking the cocharge labeling. If on the other hand, $\lambda_{a+1} = 0$, then the algorithm terminates and outputs false. \square

A corollary to this theorem is that a tableau is λ -catabolizable if and only if it is λ -column catabolizable (Proposition 2.6).

3.3. Corollary 3.8 allows us to characterize catabolizability in a similar way to the Greene's Theorem interpretation of the shape of the insertion tableau of a word (see [2, Lemma A1.1.7]). This is reminiscent of the combinatorial description of two-sided cells in the affine Weyl group of type A (see, for instance, [9]), and also of the usual way of computing cocharge of semistandard words.

Let w be a standard word and z its cocharge labeling. Define $\tilde{w} : \mathbb{Z}_{\leq n} \rightarrow \mathbb{Z}_{\geq 0}$ by $i \mapsto z_{i+kn} + k$, where k is the unique integer so that $i + kn \in [n]$. Also let \tilde{z} refer to this same map. Let $\bar{\cdot} : \mathbb{Z}_{\leq n} \rightarrow \mathbb{Z}/n\mathbb{Z}$ be the map sending an integer to its congruence class mod n .

A *chain* of \tilde{w} of *length* $k' + 1$ is a sequence $\mathbf{j} = (j_{k'}, j_{k'-1}, \dots, j_0)$ satisfying

- (i) $j_{k'} < j_{k'-1} < \dots < j_0$,
- (ii) $\tilde{w}(j_i) = i$, for all $i \in [0, k']$,
- (iii) $\bar{j}_i \neq \bar{j}_{i'}$, for all $i, i' \in [0, k']$ with $i \neq i'$.

The underlying set of \mathbf{j} is $\mathbf{j}^* := \{j_{k'}, j_{k'-1}, \dots, j_0\}$.

A k -*bounded chain family* of \tilde{w} is a set $\mathfrak{A} = \{A_1, \dots, A_l\}$ of chains of \tilde{w} of lengths at most k such that the subsets \bar{A}_i of $\mathbb{Z}/n\mathbb{Z}$ are disjoint. The *support* of \mathfrak{A} is $\cup_i A_i^* \subseteq \mathbb{Z}_{\leq n}$ and the *size* of a k -bounded chain family is the cardinality of its support. The maximum size of a k -bounded chain family of \tilde{w} is denoted $I_k(\tilde{w})$.

Let $s_d : \mathbb{Z}_{\leq n} \rightarrow \mathbb{Z}_{\leq n}$ for $d \in [n-1]$ be the affine versions of the simple reflections of \mathcal{S}_n : s_d transposes $d + kn$ and $d + 1 + kn$ for all $k \leq 0$. Certainly $\tilde{u}s_d = \tilde{u}\tilde{s}_d$ if $u \rightsquigarrow us_d$ is cocharge-preserving. The action of \mathcal{S}_n on chains is given by $s_d(\mathbf{j}) = (s_d(j_{k'}), s_d(j_{k'-1}), \dots, s_d(j_0))$.

Now assume that standard words u and v differ from each other by a simple transposition, $v = us_d$, so that the transformation $u \rightsquigarrow v$ is cocharge-preserving.

Lemma 3.13. *With u, v as above, suppose $u_d^{cc} \neq u_{d+1}^{cc} + 1$. Then, if \mathbf{j} is a chain of \tilde{u} , then $s_d(\mathbf{j})$ is a chain of \tilde{v} .*

Proof. First observe

$$(10) \quad \tilde{v}(s_d(\mathbf{j})) = \tilde{u}(s_d(s_d(\mathbf{j}))) = \tilde{u}(\mathbf{j}).$$

The only way $s_d(\mathbf{j})$ is not a chain of \tilde{v} is if $j_i = j_{i-1} - 1 \equiv d \pmod{n}$ for some i , which is excluded by the assumption $u_d^{cc} \neq u_{d+1}^{cc} + 1$. \square

Theorem 3.14. *With the notation above,*

$$(11) \quad \sum_{i=1}^k \text{ctype}(w)_i = I_k(\tilde{w}).$$

Proof. By Corollary 3.8, it suffices to check that $I_k(\tilde{w})$ is $\sum_{i=1}^k \lambda_i$ for $w^{\text{cc}} = \text{rowword}(Z_\lambda)$, is invariant under non-zero (co)rotations, catabolism transformations, and Knuth transformations.

For $w^{\text{cc}} = \text{rowword}(Z_\lambda)$, $I_k(\tilde{w}) = \sum_{i=1}^k \lambda_i$. There holds

$$|\{\bar{i} : \tilde{w}(i) \leq k - 1\}| = \sum_{i=1}^k \lambda_i$$

so that $I_k(\tilde{w})$ cannot possibly exceed this number. It is easy to exhibit a k -bounded chain family of \tilde{w} of size $\sum_{i=1}^k \lambda_i$.

Non-zero (co)rotations. If v is a non-zero corotation of u , then $\tilde{v}(i) = \tilde{u}(i - 1)$ and $\tilde{u}(n) \neq 0$. Thus the support of a k -bounded chain family of \tilde{u} cannot contain n and the notions of a k -bounded chain family for \tilde{u} and \tilde{v} differ only by shifting indices by 1.

Catabolism transformations. This follows from the special case of Lemma 3.13 in which $|u_d^{\text{cc}} - u_{d+1}^{\text{cc}}| \neq 1$.

Knuth transformations. We may assume that $u \rightsquigarrow v$ is a Knuth transformation with $v = us_d$ and $u_d^{\text{cc}} = a + 1$, $u_{d+1}^{\text{cc}} = a$, $u_{d+2}^{\text{cc}} = a$.

$$(12) \quad u = \cdots a + 1 \ a \ a \cdots \rightsquigarrow v = \cdots a \ a + 1 \ a \ \cdots$$

By Lemma 3.13 and its proof, any k -bounded chain family of \tilde{v} yields one of the same size for \tilde{u} , and any k -bounded chain family \mathfrak{A} of \tilde{u} yields one of the same size for \tilde{v} provided any $\mathbf{j} \in \mathfrak{A}$ does not satisfy $j_i = j_{i-1} - 1 \equiv d \pmod n$ for some i . If this is the case, then one checks that $\{s_d s_{d+1}(\mathbf{j}) : \mathbf{j} \in \mathfrak{A}\}$ is a k -bounded chain family of \tilde{v} , clearly of the same size as \mathfrak{A} . \square

This proof differs in an important way with that of the Greene's Theorem interpretation of insertion tableau [2, Lemma A1.1.7]. None of the alterations of chains in the proof change the lengths of chains. This allows us to conclude the stronger statement:

Theorem 3.15. *If $\text{ctype}(w) = \lambda$ then \tilde{w} has an $\ell(\lambda)$ -bounded chain family consisting of chains of lengths $\lambda'_1, \lambda'_2, \dots, \lambda'_{\lambda_1}$, where λ' is the conjugate partition of λ .*

Applying Lemma 3.13 and Theorem 3.14, we obtain

Corollary 3.16. *If $u \rightsquigarrow v$ is an ascent, then $\text{ctype}(u) \supseteq \text{ctype}(v)$.*

The next corollary is a strengthening of [8, Lemma 51].

Corollary 3.17. *If v is a non-zero corotation of u , then $\text{ctype}(v) = \text{ctype}(u)$. If v is a zero corotation of u , then $\text{ctype}(v)$ is strictly less than $\text{ctype}(u)$ in dominance order.*

Remark 3.18. It is tempting to conjecture that if $u \rightsquigarrow v$ is an ascent or a zero corotation and $\text{ctype}(v) \neq \text{ctype}(u)$, then $\text{ctype}(v) \trianglelefteq \text{ctype}(u)$ is a covering relation in the dominance order poset. This is false. For example, if $u = 0\ 0\ 1\ 1\ 0$, $v = 1\ 0\ 0\ 1\ 1$, then $\text{ctype}(u) = (3, 1, 1)$, which does not cover $\text{ctype}(v) = (2, 1, 1, 1)$; if $u = 1\ 1\ 2\ 2\ 0\ 0\ 1\ 0$, $v = 1\ 1\ 2\ 2\ 0\ 0\ 0\ 1$, then $\text{ctype}(u) = (3, 3, 1, 1)$, which does not cover $\text{ctype}(v) = (3, 2, 1, 1, 1)$.

ACKNOWLEDGMENTS

This paper would not have been possible without the generous advice from and many detailed discussions with Mark Haiman. I am also grateful to Michael Phillips and Ryo Masuda for help typing and typesetting figures.

REFERENCES

- [1] J. Blasiak, Cyclage, catabolism, and the affine Hecke algebra, preprint (2010), arXiv:1001.1569v1.
- [2] S. Fomin, Knuth equivalence, jeu de taquin, and the Littlewood-Richardson rule, Appendix I in *Enumerative Combinatorics, vol. 2*. Cambridge University Press, 1999.
- [3] A. M. Garsia and C. Procesi, On certain graded \mathcal{S}_n -modules and the q -Kostka polynomials, *Advances in Mathematics*. **94** (1992), 82–138.
- [4] M. Haiman, Combinatorics, symmetric functions, and Hilbert schemes, *Current Developments in Mathematics*. (2002), 39–111.
- [5] L. Lapointe, A. Lascoux, and J. Morse, Tableau atoms and a new Macdonald positivity conjecture, *Duke Math. J.* **116** (2003), no. 1, 103–146.
- [6] A. Lascoux, Cyclic permutations on words, tableaux and harmonic polynomials, *Proceedings of the Hyderabad Conference on Algebraic Groups, 1989*. Manoj Prakashan, Madras (1991), 323–347.
- [7] A. Lascoux and M. Schützenberger, Le monoïde plaxique, *Noncommutative Structures in Algebra and Geometric Combinatorics (Naples, 1978)*, *Quaderni della Ricerca Scientifica*, **109** A. de Luca (ed.), CNR, Roma, 1981, pp. 129–156.
- [8] M. Shimozono and J. Weyman, Graded characters of modules supported in the closure of a nilpotent conjugacy class, *Europ. J. Combinatorics*. **21** (2000), 257–288.
- [9] N. Xi, The based ring of two-sided cells of affine Weyl groups of type \tilde{A}_{n-1} , *Mem. Amer. Math. Soc.* **157** (2002), 749.

DEPARTMENT OF MATHEMATICS, UC BERKELEY