Canonical Decompositions of Affine Permutations, Affine Codes, and Split k-Schur Functions

Tom Denton*

Department of Mathematics
York University
Toronto, Ontario, Canada

sdenton4@gmail.com

Submitted: Apr 12, 2012; Accepted: Oct 8, 2012; Published: Nov 8, 2012
Mathematics Subject Classifications: 05E05, 05E15

Abstract

We develop a new perspective on the unique maximal decomposition of an arbitrary affine permutation into a product of cyclically decreasing elements, implicit in work of Thomas Lam [Lam06]. This decomposition is closely related to the affine code, which generalizes the k-bounded partition associated to Grassmannian elements. We also prove that the affine code readily encodes a number of basic combinatorial properties of an affine permutation. As an application, we prove a new special case of the Littlewood-Richardson Rule for k-Schur functions, using the canonical decomposition to control for which permutations appear in the expansion of the k-Schur function in noncommuting variables over the affine nil-Coxeter algebra.

*Supported by York University and the Fields Institute.

Contents

1 Introduction **2**
1.1 Further Directions. 5
1.2 Overview 5
1.3 Acknowledgements 6

2 Background and Definitions **7**
2.1 The Affine Nilcoxeter Algebra and Affine Permutations. 7
2.2 Cyclic Elements in $\mathbb{A}_k$. 11
2.3 k-Schur Functions. 12

3 Canonical Cyclic Decompositions and k-Codes **14**
3.1 Maximal Cyclic Decompositions and k-Codes 15
3.2 Maximizing Moves on k-Codes. 18
3.3 Insertion Algorithm. 22
3.4 Bijection Between Affine Permutations and k-Codes. 23
3.5 Relating the Various Cyclic Decompositions of an Affine Permutation. . . . 26
3.6 Affine Codes and k-Codes 28
3.7 Grassmannian and i-Dominant Elements. 29
3.8 Generalized Pieri Rule. 30

4 Multiplication of Cyclically Increasing Elements by Cyclically Decreasing Elements. **31**
4.1 Products of Cyclically Increasing and Decreasing Elements 34

5 The k-Littlewood-Richardson Rule for Split k-Schur Functions **35**

1 Introduction

The affine permutation group $\tilde{S}_{k+1}$ was originally described by Lusztig [Lus83] as a combinatorial realization of the affine Weyl group of type $A_k^{(1)}$. The affine permutations have since been extensively studied; a very good overview of the basic results may be found in [BB05]. The group $\tilde{S}_{k+1}$ is important for a variety of reasons; for example, new results on $\tilde{S}_{k+1}$ often generalize or give new results in the classical symmetric group. Additionally, $\tilde{S}_{k+1}$ is the affine Weyl group of type $A_k^{(1)}$, and new combinatorics in the affine symmetric group suggest new directions of exploration for general affine Weyl groups. Finally, the affine nil-Coxeter algebra, which is closely related to the affine permutation group, has proven very useful in the study of symmetric functions, via the construction of Schur (and k-Schur) functions in non-commuting variables [FG98][Lam06].

As our primary objective, we develop new machinery for finding the unique maximal decomposition of an arbitrary affine permutation. This may be interpreted as a canonical

reduced decomposition of each affine permutation. This composition is encoded in the *affine code*, or k-*code*, which may be interpreted as a weak composition with $k+1$ parts, at least one of which is zero[1]. We interpret the diagram of an affine code as living on a cylinder. (Before the connection to the affine code was noticed, we called this object a k-castle, because when the affine code's Ferrer's diagram is drawn on a cylinder, it resembles the ramparts of a castle. The requirement that one part of the composition is zero means that the castle always has a "gate." See Figure 4.) One may then quickly determine whether two affine permutations given by reduced words are equal by putting each in their canonical form: Thus, we provide an alternative solution of the word problem for the affine symmetric group. The affine code readily yields other useful information about the affine permutation, including its (right) descent set and length. Furthermore, the Dynkin diagram automorphism on $\tilde{S}_{k+1}$ may be realized by simply rotating the affine code.

While in review, it was noticed that the k-code coincides with the affine code, and that the unique maximal decomposition is implicit in the work of Lam [Lam06][Theorem 13]. Our work here gives an alternative proof of the existence and uniqueness of the maximal decomposition, as well as introducing the perspective of the two-row moves on cyclic decompositions of an element. This added perspective is imminently useful when considering combinatorial problems arising in the affine symmetric group; this utility is demonstrated in Section 5, and will be further demonstrated in future work.

We furthermore describe an insertion algorithm on affine codes, which gives rise to the notion of a set of standard recording tableau in bijection with the set of reduced words for an affine permutation with affine code α. We also generalize a number of constructions that arise in the study of k-Schur functions (described below) to general affine permutations. In particular, the notions of k-conjugation and weak strip appear and generalize naturally in the study of affine codes.

Initially we developed this machinery in order to prove a special case of the k-Littlewood-Richardson rule describing the multiplication of k-Schur functions. The k-Schur functions $s_\lambda^{(k)}$ are indexed by k-bounded partitions, and give a basis for the ring $\Lambda^{(k)}$ defined as the algebraic span of the complete homogeneous functions h_i with $i \leqslant k$.

The k-Schur functions were originally defined combinatorially in terms of k-atoms, and conjecturally provide a positive decomposition of the Macdonald polynomials [LLM03]. Since their original appearance, these functions have attracted much attention, but many basic properties remain elusive. As of this writing, the author estimates that there are at least five different definitions, all of which are conjecturally equivalent. A good overview of the state-of-the-art in the study of k-Schur functions, including many of the various definitions, is [LLM$^+$12].

One definition of the k-Schur functions is given by the k-Pieri rule. The k-bounded partitions are are in bijection with $(k+1)$-cores and Grassmannian affine permutations. Lam demonstrated that the cyclically decreasing elements in the affine nil-Coxeter algebra commute and satisfy the same multiplication as the h_i's [Lam06]. As such, the k-Pieri rule may be used to construct elements in the affine nil-Coxeter algebra which mimic the

[1]The affine code generalizes both the inversion vector and the Lehmer code of a classical permutation.

k-Schur functions. This is the realization of the k-Schur functions we use throughout this paper.

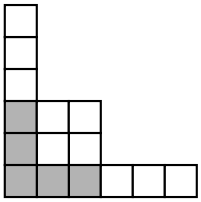
Definition 1. Given a shape μ, the *k-boundary* $\partial_k(\mu)$ of μ is the skew shape obtained by removing all boxes with hook $> k$. A skew shape is *connected* if any box may be reached from any other box by a sequence of vertical and horizontal steps. A $k+1$-core μ *splits* if the k-boundary $\partial_k(\mu)$ is not connected. If μ splits, then each connected component of $\partial_k(\mu)$ is the boundary of some $k+1$ core ρ_i. These cores ρ_i are the *components* of μ. Any collection of diagonally-stacked connected components may similarly be associated to a core; such a collection we call a *factor*, in anticipation of the main result.

Our main application is the following special case of the k-Littlewood-Richardson rule, which appears as Theorem 70:

Theorem 2. *Suppose μ splits into components μ_i. Then*

$$s_\mu^{(k)} = \prod s_{\mu_i}^{(k)}.$$

Example 3. Consider the 5-core $(6,3,3,1,1,1)$, associated to the 4-bounded partition $(3,2,2,1,1,1)$:



The 5-boundary is in white, while the non-boundary boxes are shaded grey. The boundary splits into three connected components, $(1,1,1)$, $(2,2)$, and (3). Then the theorem states that:

$$s_{(6,3,3,1,1,1)}^{(4)} = s_{(3)}^{(4)} s_{(2,2)}^{(4)} s_{(1,1,1)}^{(4)}.$$

This special case of the k-Schur Littlewood Richardson rule is similar in flavor to one proven by Lapointe and Morse [LM07]. Their special case, the k-rectangle rule, involves multiplication of a k-Schur function indexed by a rectangle R with maximal hook $(k+1)$ by an arbitrary k-Schur function λ. In this case:

$$s_R^{(k)} s_\lambda^{(k)} = s_{R\cup\lambda}^{(k)},$$

where $R \cup \lambda$ is the partition obtained by stacking the Ferrer's diagrams of R and λ and then "down-justifying"[2] the resulting shape to obtain a k bounded partition. Given the k-rectangle rule, the multiplication of the k-Schur functions for a fixed k is then fully determined by the multiplication of the k-Schur functions indexed by shapes strictly contained within a k-rectangle. The k-rectangle rule was given a combinatorial interpretation in the affine nil-Coxeter setting in [BBTZ11].

The splitting condition we consider here is distinct from the k-rectangle rule, and provides some products of k-Schur functions contained strictly within a k-rectangle, and thus advances our overall understanding of the k-Littlewood-Richardson rule.

[2]Or "up-justifying," if you prefer the English notation for partitions.

1.1 Further Directions.

Our results on affine codes suggest a number of questions for further exploration. In particular, we expect that our perspective will be helpful in problems relating to reduced decompositions of affine permutations, especially those relating to the affine Stanley symmetric functions $\mathfrak{S}_x$, originally studied in [Lam06]. The affine Stanley symmetric function may be defined as a sum over decompositions of an affine permutation into a product of cyclically decreasing elements; our framework gives a natural way to relate these various decompositions, which we will explore in further work. The affine codes may also be helpful in the enumeration of reduced words for either classical or affine permutations, a problem which has proven especially difficult.

As noted in [Lam06], the problem of expanding the k-Schur functions over the nil-Coxeter algebra $\mathbb{A}_k$ is equivalent to finding the k-Littlewood-Richardson coefficients. A number of the supporting results in this work determine coefficients in the expansion of the k-Schur function for special elements, using the affine code constructions. We expect that more information about the coefficients in the expansion may be gleaned from further study, which in turn will illuminate the k-Littlewood-Richardson coefficients.

1.2 Overview

In Section 2 we review basic concepts from the literature and establish notation that will be used throughout the paper. This includes a review of affine permutations, the affine nil-Coxeter algebra, cyclically decreasing elements, and the expression of the k-Schur functions in non-commuting variables over the affine nil-Coxeter algebra..

The bulk of the paper is in Section 3. In this section, we construct the bijection(s) between affine permutations and k-codes, via maximal decompositions. In Subsection 3.1, we prove that every affine permutation has a unique maximal decomposition as a product of cyclically decreasing elements. This provides the first main result of the paper, Theorem 20. The proof of the theorem is constructive, and provides a fast algorithm for computing the maximal decomposition.

In Subsection 3.2, we establish 'moves' between various reduced decompositions of an affine permutation into cyclically decreasing elements. These allow us to prove Proposition 27, which establishes that the maximal decomposition of any affine permutation into cyclically decreasing elements satisfies a 'shifted containment' property, which is key in the identification of the decomposition with a weak composition.

In Subsection 3.3, we reinterpret the maximizing moves to establish an insertion algorithm on k-codes. This algorithm is reversible, which allows us to associate a set of recording tableaux to each k-code, and thus to each affine permutation. By construction, these recording tableaux are in bijection with reduced words for the affine permutation. This is the content of Theorem 29.

In Subsection 3.4, we prove our main result, Theorem 38, which establishes the bijection between k-codes and affine permutations. We also establish a relationship between descents of k-codes and descents of the affine permutations they correspond to.

In Subsection 3.5, it is observed that there are actually four different bijections between affine permutations and k-codes, according to different choices for the maximal decomposition: One can build either a decomposition into cyclically decreasing or increasing elements, from the right side or the left side. Here we investigate the relationships between the four k-codes assigned to a given affine permutation. The increasing and decreasing decompositions are related by a generalization of the k-conjugate, a vital construction on k-bounded partitions. We also note that the k-codes of the left and right decreasing decompositions are related by a permutation (Proposition 43).

We then focus on Grassmannian elements in Subsection 3.7. These are affine permutations with right descent set $\{0\}$ or $\emptyset$. They are of particular interest because they index the k-Schur functions: Grassmannian elements are in bijection with k-bounded partitions, which may be interpreted as a k-code α with only one descent at α_0. We show that the usual k-conjugate of k-bounded partitions corresponds to switching between two maximal decompositions of the associated Grassmannian element (Proposition 51). This allows us to define the k-conjugate on arbitrary affine permutations.

The k-Pieri rule is used to define the k-Schur functions, and an important characterization of the Pieri rule is by weak horizontal strips. In particular, consider k-bounded partitions $\lambda \subset \mu$, and let the k-conjugate of λ and μ be λ' and μ' respectively. (The k-conjugate is defined in Section 2.3.) Then we say that the skew shape μ/λ is a *weak strip* if no column of μ/λ contains two boxes, and no row of μ'/λ' contains two boxes. Suppose the affine permutations associated to λ and μ are x and y respectively. Indeed, μ/λ is a weak strip if and only if there exists a cyclically decreasing element d_A such that $y = d_A x$.

In Subsection 3.8, we generalize the combinatorial Pieri rule by showing that multiplying any affine permutation by a cyclically decreasing element adds at most one box to each row of its k-code, while multiplication by a cyclically increasing element adds at most one box to each column of its k-code.

Section 4 investigates the results of multiplying cyclically increasing and decreasing elements together. In particular, we find near-commutation rules: for cyclically increasing and decreasing elements u_B and d_A, there exist A', B' such that $u_B d_A = d_{A'} u_{B'}$. The main result is Proposition 65.

Finally, we use the machinery of the previous sections to prove our special case of the k-Littlewood Richardson rule in Section 5. The main result is Theorem 70.

1.3 Acknowledgements

I would like to thank the Fields Institute for providing space and hosting our weekly Algebraic Combinatorics seminar, where many of the ideas in this paper were discussed, argued, and strengthened. Many thanks are also due to Chris Berg, Nantel Bergeron, Zhi Chen, Anne Schilling, Luis Serrano, Nicolas Thiéry and Mike Zabrocki for helpful comments and conversation. Funding during this research was provided by Nantel Bergeron and York University. Thanks are also due to the LACIM group at UQAM for their repeated hospitality.

One reviewer of the article patiently indicated many small errors in the text and supplied many generally helpful comments, for which I am grateful. Of course, any remaining errors are my own.

Finally, much of the work in this project made extensive use of the Sage computer algebra system and the Sage-Combinat project [S+09][SCc09], and was particularly reliant on code contributions from Chris Berg, Franco Saliola, Anne Schilling, and Mike Zabrocki.

2 Background and Definitions

In this section we review background material and fix notations for the remainder of the paper.

2.1 The Affine Nilcoxeter Algebra and Affine Permutations.

We begin by defining the affine nil-Coxeter algebra, and reviewing some basic facts and definitions relating to affine permutations. Good references on affine Coxeter groups in general and the affine symmetric group in particular include [Hum90], [BB05].

Let k be a positive integer. Let I indicate the index set $\mathbb{Z}_{k+1} = \{0, 1, \ldots, k\}$, which correspond to nodes in the Dynkin diagram of type $A_k^{(1)}$. Indices from I are thus always considered modulo $k+1$. The *Dynkin diagram* of type $A_k^{(1)}$ is the cyclic graph with vertices labeled by elements of I, and an edge connecting each pair of indices i and $i+1$. For brevity, we let $[p, q] := \{p, p+1, \ldots, q-1, q\} \subsetneq I$ for $p \neq q-1$. (For example, with $k = 5$, the set $[4, 2] = \{4, 5, 0, 1, 2\}$.) We call a subset $A \subsetneq I$ *connected* if the corresponding subgraph of the Dynkin diagram is connected; *i.e.,* $A = [i, l]$ for some i, l. A connected component of an arbitrary $A \subsetneq I$ is a maximal connected subset of A.

Definition 4. The *affine nil-Coxeter monoid* $\hat{\mathbb{A}}_k$ is generated by the alphabet $\{a_i \mid i \in I\}$, subject to the relations:

- $a_i^2 = 0$,
- $a_i a_j = a_j a_i$ for all $j > i$ with $j - i > 1$,
- $a_i a_{i+1} a_i = a_{i+1} a_i a_{i+1}$ for all i, and
- $x0 = 0x = 0$ for all $x \in \hat{\mathbb{A}}_k$.

The affine nil-Coxeter algebra $\mathbb{A}_k$ is the monoid algebra of $\hat{\mathbb{A}}_k$. The *classical nil-Coxeter monoid* $\hat{\mathbb{A}}_k^0$ and corresponding monoid algebra $\mathbb{A}_k^0$ are obtained as a (parabolic) quotient of $\mathbb{A}_k$ by evaluating $a_0 = 0$.

We will also occasionally use the *affine symmetric group* $\tilde{S}_k$ generated by the alphabet $\{s_i \mid i \in I\}$, subject to the relations:

- $s_i^2 = 1$,
- $s_i s_j = s_j s_i$ for all $j > i$ with $j - i > 1$, and
- $s_i s_{i+1} s_i = s_{i+1} s_i s_{i+1}$ for all i.

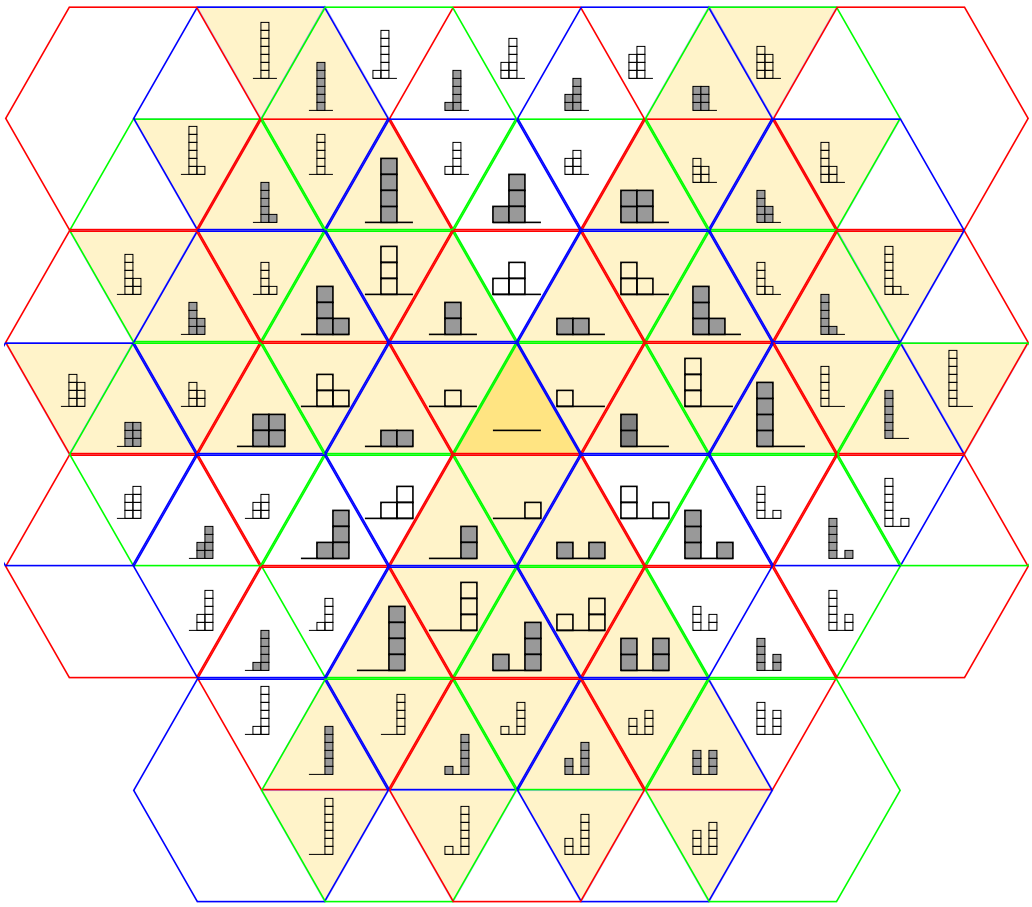


Figure 1: All 2-castles with $\leqslant 7$ boxes, drawn in the alcoves corresponding to the various affine permutations in $\tilde{S}_3$. The colored walls of the alcoves indicate which simple transposition is used to cross that wall. Blue is s_0, green is s_1, and red is s_2. The orange-shaded regions indicate the three dominant cones; in particular, the 0-dominant elements are in the cone which opens to the upper-right. The shading in the boxes of the castles themselves corresponds to length; odd-length permutations have no shading, while even-length permutations are shaded gray.

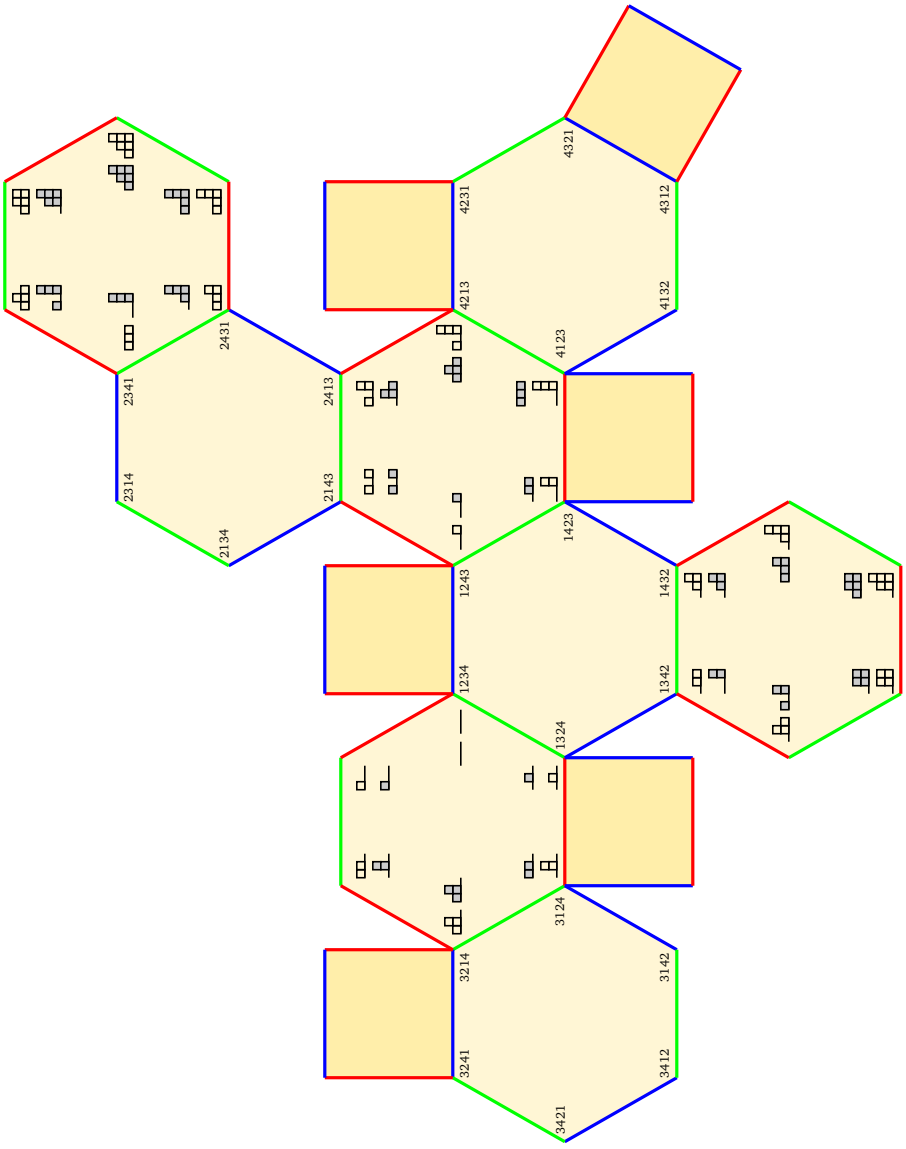


Figure 2: DIY Permutahedron: Cut out and glue together to get a 3-dimensional representation of this paper's results, restricted to the classical symmetric group S_4. For best results, print large and in color on heavy card-stock. Each vertex is labeled by a permutation, and the k-codes associated to the right decreasing (in white) and increasing (gray) decompositions of the permutation. Edges correspond to right multiplication by a simple transposition: red for s_1, green for s_2 and blue for s_3. You can use the blank square faces to mark your own favorite permutation statistics!

Elements of the affine symmetric group may be considered as (affine) permutations $x : \mathbb{Z} \to \mathbb{Z}$ subject to the additional requirements that:

- $\sum_{i=1}^{k+1} x(i) = \binom{k+2}{2}$, and
- $x(i+k+1) = x(i) + k + 1$.

Any affine permutation x is then completely specified by its *window notation*, given by the vector $(x(1), x(2), \ldots, x(k+1))$. Affine permutations may be considered as bi-infinite sequences, setting $x_l := x(l)$. These affine permutations are in bijection with non-zero elements of the nil-Coxeter monoid.

The generators s_i may be considered as the simple transpositions exchanging $m(k+1)+i$ with $m(k+1)+i+1$ for every $m \in \mathbb{Z}$. The set of affine permutations admit a *left action* and a *right action* by the generators s_i. Considering x as a bi-infinite sequence $(\ldots, x(-1), x(0), x(1), \ldots)$, we may consider the left action of the generators as an action on values (exchanging $m(k+1)+i$ with $m(k+1)+i+1$ for every $m \in \mathbb{Z}$), while the right action is on positions (exchanging the values in positions $m(k+1)+i$ and $m(k+1)+i+1$ for every $m \in \mathbb{Z}$).

A *reduced word* or *reduced expression* for x is minimal length sequence $(w_1, \ldots, w_l)$ with $w_l \in I$ such that $x = s_{w_1} \cdots s_{w_l}$. The number l is the *length* of x, which we denote $\mathrm{len}(X)$. It is a consequence of basic Coxeter theory that an expression is reduced if and only if $a_{w_1} \cdots a_{w_l} \neq 0$. We mainly consider affine permutations as elements of the nil-Coxeter monoid, partially because this is the natural setting to work in for the k-Schur functions, and partially to avoid worrying about whether a given expression is reduced.

To save space, we will often write words in $\mathbb{A}_k$ as a subscript: for example, we write $a_1 a_2 a_1$ as $a_{1,2,1}$.

Let x be an affine permutation. We recall the *set of right descents* $D_R(x) \subsetneq I$ of an element x. We say that x has a *right descent* at $i \in I$, and write $i \in D_R(x)$,

- $x(i) > x(i+1)$,
- x has a reduced word ending with the generator s_i.

Likewise, we define the left descents $D_L(x) \subsetneq I$. Recall that x has a *left descent* at $i \in I$, and $i \in D_L(x)$ if either of the following two equivalent statements hold:

- i appears to the right of $i+1$ in x considered as a bi-infinite sequence,
- x has a reduced word beginning with the generator s_i.

Note that for any x, $D_R(x) \neq I$. (If $D_R(x) = I$, then x would be a longest element in $\tilde{S}_k$. But such elements do not exist in affine Coxeter groups for a variety of reasons. [Hum90])

In Figure 1, illustrating the bijection between 2-castles and affine permutations in $\tilde{S}_3$, we make use of the *alcove model* for affine permutations, for which we refer the unfamiliar reader to [Hum90]. In short, each triangle in the picture is an 'alcove,' corresponding to a particular affine permutation. Crossing a wall of an alcove to reach an adjacent alcove corresponds to multiplication by a simple transposition.

Lemma 5 (Extended Braid Relation)**.** *For any set $[i,j] \subsetneq I$, we have:*

$$a_{i,i+1,\ldots,j-1,j,j-1,\ldots,i+1,i} = a_{j,j-1,\ldots,i+1,i,i+1,\ldots,j-1,j}.$$

Proof. This follows from repeated application of the braid relation. □

The Dynkin diagram of type $A_k^{(1)}$ admits a cyclic symmetry, which descends to an algebra automorphism of $\mathbb{A}_k$.

Definition 6. The Dynkin Diagram automorphism $\Psi : \mathbb{A}_k \to \mathbb{A}_k$ is defined by its action on the generators:

$$\Psi(a_i) = a_{i+1}.$$

We observe that $\Psi^{(k+1)}$ is the identity.

2.2 Cyclic Elements in $\mathbb{A}_k$.

Definition 7. Given a subset $A \subsetneq I$ with $|A| = n$ we define the cyclically decreasing element d_A (d for 'down') to be the product $d_A := a_{i_1} \cdots a_{i_n}$ for $i_l \in A$, where if $j, j-1 \in A$ then j appears to the left of $j-1$ in any reduced word for d_A. The cyclically increasing element u_A (u for 'up') is defined similarly, where if j and $j-1 \in A$ then j appears to the right of $j-1$ in any reduced word for d_A.

Then we define:

$$h_i := \sum_{|A|=i} d_A, \text{ and } e_i := \sum_{|A|=i} u_A.$$

For a partition $\lambda = (\lambda_1, \ldots, \lambda_n)$, $h_\lambda = \prod h_{\lambda_i}$, and $e_\lambda = \prod e_{\lambda_i}$.

We frequently use the notation $A - 1 := \{i - 1 \mid i \in A\}$ and occasionally $A + 1 := \{i + 1 \mid i \in A\}$.

A *cyclically increasing* (respectively, *cyclically decreasing*) element of $\mathbb{A}_k$ is an element specified by ordered collection of subsets $A_i \subsetneq I$, given by $u_{A_n} u_{A_{n-1}} \cdots u_{A_1}$ (respectively, $d_{A_n} d_{A_{n-1}} \cdots d_{A_1}$). We abbreviate such products using the notation $\vec{A} := \{A_1, \ldots, A_n\}$, so that $u_{\vec{A}} := u_{A_n} \cdots u_{A_1}$ and $d_{\vec{A}} := d_{A_n} \cdots d_{A_1}$. A cyclically increasing (resp. decreasing) product x is *maximal* if the *shape of* $\vec{A}$ given by the vector $\mathrm{sh}(A) := (|A_1|, |A_2|, \ldots, |A_n|)$ is lexicographically maximal amongst all cyclically increasing (resp. decreasing) expressions for x.

Example 8. Let $k = 5$, so that $I = \{0, 1, 2, 3, 4, 5\}$. Set $A = \{0, 2, 4, 5\}$. Then $d_A = a_0 a_5 a_4 a_2$, and $u_A = a_2 a_4 a_5 a_0$. There is a bijection between proper subsets of I and cyclically decreasing elements.

Theorem 9 ([Lam06])**.** *The elements h_i generate a commutative subalgebra of $\mathbb{A}_k$.*

Definition 10. The *right descent set* of an element $w \in \mathbb{A}_k$ is the set $D_R(w) := \{p \in I \mid ws_p = 0\}$. The left descent set $D_L(w)$ is defined similarly.

For $p \in I$, an element $w \in \mathbb{A}_k$ is *p-dominant* if $D_R(w) \subset \{p\}$. When $p = 0$, such elements are also known as *Grassmannian elements*.

Lemma 11. *A cyclically decreasing (or increasing) element is connected if and only if it is i-dominant for some i.*

Proof. If not connected, then the element has multiple descents. If it is connected, no relations may be applied to the element, and so there is only one right descent. □

2.3 k-Schur Functions.

The literature on k-Schur functions is extensive, but an excellent overview is given in "A Primer on k-Schur Functions," by Schilling and Zabrocki [LLM+12]. Additional background on the realization of the k-Schur functions in non-commuting variables over the affine nil-Coxeter algebra may be found in [BBTZ11].

Definition 12. A k-bounded partition is a partition $\lambda = (\lambda_1, \ldots, \lambda_n)$ with each $\lambda_i \leqslant k$. A $k+1$-core is a partition μ with no hooks of length $k+1$. Given a $k+1$-core μ, the $k+1$-*boundary* $\partial_{k+1}(\mu)$ is the skew shape obtained by deleting all boxes of μ with hook length greater than $k+1$. When k is not ambiguous, we will just write $\partial(\mu)$.

There is a well-known bijection between k-bounded partitions and $k+1$ cores. The bijection is defined by an algorithm on the bounded partition: starting with the first row of the Ferrer's diagram for λ, if the first box b of a given row has hook length $\geqslant k+1$, we add boxes to the beginning of the row until the box b has hook length $\leqslant k$. We perform this operation on each row of λ sequentially to obtain a $k+1$-core. We may recover the k-bounded partition by taking the k-boundary of the $k+1$-core and pushing all of the boxes in the resulting skew shape to the left to form a partition.

For a k-bounded partition λ, we write $\mathfrak{c}(\lambda)$ for the associated $k+1$-core, and for a $k+1$-core μ, write $\mathfrak{p}(\mu)$ for the associated k-bounded partition. Thus $\mathfrak{p}(\mathfrak{c}(\lambda)) = \lambda$.

Of considerable importance in the study of k-Schur functions is the k-conjugate. For a k-bounded partition λ, the k-conjugate $\lambda^{(k)} := \mathfrak{p}(\mathfrak{c}(\lambda)')$, where $\mathfrak{c}(\lambda)'$ denotes the usual conjugate of the $k+1$-core associated to λ. Notice that the usual conjugate of a k-bounded partition need not be k-bounded; the k-conjugate returns another k-bounded partition. (See Figure 3.) The k-conjugate combinatorially implements on the level of k-Schur functions the automorphism of the symmetric functions which exchanges h_i and e_i.

Recall that for λ, ν partitions of n, ν *dominates* λ if $0 \leqslant \sum_{i=1}^{j}(\nu_i - \lambda_i)$ for every j (possibly padding one of the partitions with zeroes if their lengths are unequal). In this case, we write $\nu \succ \lambda$.

The k-Schur functions are indexed by k-bounded partitions, and may be defined by the Pieri rule. The Pieri rule gives an inductive definition of the k-Schur functions, by setting $s^{(k)}_{(l)} := h_l$, and then expressing $h_l s^{(k)}_\mu = \sum_\nu s^{(k)}_\nu$ according to some restrictions on ν. In particular, the partitions ν satisfy a triangularity property with respect to the dominance order, allowing recursive definition of the k-Schur functions.

There are different interpretations of the Pieri rule in different contexts, but the primary definition is by weak horizontal strips. Given partitions $\lambda \subset \mu$, we say that the skew shape $\mu \setminus \lambda$ is a *horizontal strip* if each column of $\mu \setminus \lambda$ contains at most one box. Likewise, it is a *vertical strip* if each row contains at most one box. If $\lambda \subset \mu$ are k-bounded partitions, we say that the skew shape $\mu \setminus \lambda$ is a *weak horizontal strip* if $\mu \setminus \lambda$ is a horizontal strip and $\mu^{(k)} \setminus \lambda^{(k)}$ is a vertical strip. Then the Pieri rule may be stated as:

$$h_l s^{(k)}_\mu = \sum_\nu s^{(k)}_\nu,$$

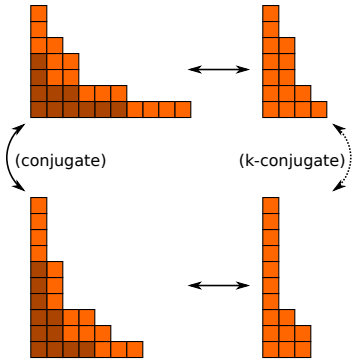


Figure 3: Illustration of the bijection between k-bounded partitions and $k+1$-cores and the k-conjugation operation. In this case, $k = 4$. On the left are two 5-cores with the boxes with hook > 5 shaded darker, and on the right are the corresponding 4-bounded partitions. The 5-cores are related by conjugation; the 4-bounded partitions are related by k-conjugation.

where $\nu \setminus \mu$ is a weak horizontal strip [LM03]. As a consequence, we can observe that if l is less than the last part of μ, then

$$h_l s_\mu^{(k)} = s_{\mu \cup l}^{(k)} + \sum_\nu s_\nu^{(k)},$$

where each $\nu \setminus \mu$ is a vertical strip. Furthermore, each ν dominates $\mu \cup l$.

Recall that there is a bijection between Grassmannian (or 0-dominant) affine permutations in $\tilde{S}_k$ and k-bounded partitions. Their relation to the k-Schur functions is described by the following theorem, which arises as a consequence of the Pieri rule:

Proposition 13. *For $l \in \{1, \ldots, k\}$, $s_{(l)}^{(k)} := h_l$. Each k-Schur function $s_\lambda^{(k)}$ appears with multiplicity one in h_λ. Furthermore, in its expansion in $\mathbb{A}_k$, $s_\lambda^{(k)}$ contains a unique 0-dominant summand, w_λ.*

There is a second interpretation of the Pieri rule in the context of the affine nil-Coxeter algebra. Take x to be a 0-dominant element of the affine nil-Coxeter monoid. Then:

$$h_l s_x^{(k)} = \sum_y s_y^{(k)},$$

where the sum is over Grassmannian elements y such that $y = d_A x$ for some $A \subsetneq I$ with $|A| = l$.

Corollary 14. *Each k-Schur function $s_\lambda^{(k)}$ contains a unique i-dominant summand for each $i \in I$.*

Proof. The statement is true for $i = 0$. One may obtain an i-dominant summand in $s_\lambda^{(k)}$ by applying $\Psi^i(w\lambda)$. This summand is unique, or else we could apply Ψ^{k-i} to obtain more than one 0-dominant summand. □

An important part of our later proofs in this paper will rely on finding coefficients of certain elements in the expansion of h_λ, e_λ, or $s_\lambda^{(k)}$. For this, we employ the notation:

$$[x]f := \text{ coefficient of } x \text{ in } f.$$

For example, in h_l, we have:

$$[d_A]h_l = \delta_{|A|,l}.$$

3 Canonical Cyclic Decompositions and k-Codes

We first consider products of cyclically decreasing elements. All of the results in this section may be adapted to products of cyclically increasing elements with small modifications. For example, results concerning k-bounded partitions for products of cyclically decreasing elements become statements about k-column bounded partitions for cyclically increasing products. These different decompositions are explored in Section 3.5.

Suppose we have a collection of subsets $\vec{A} = \{A_1, \ldots, A_n\}$ such that each $A_i \subsetneq I$. Then we can form a cyclically decreasing product $d_{\vec{A}} = d_{A_n} \cdots d_{A_1}$. Trivially every element $w \in \mathbb{A}_k$ has an expression as a cyclically decreasing product, by taking any reduced expression of length n and considering the element as a product of n cyclically decreasing elements of length 1.

Our primary goal for this section is to show that every affine permutation has a maximal expression as a product of cyclically decreasing elements, in the sense that the vector $(|A_1|, \ldots, |A_n|)$ is lexicographically maximal amongst all cyclically decreasing decompositions of x. Given such a maximal decomposition, we may associate it with its k-code, defined immediately below. We then show that k-codes are in bijection with affine permutations. Along the way we will create algorithms analogous to jeu de taquin and insertion on k-codes, corresponding respectively to maximizing a cyclically decreasing decomposition and multiplying by a single generator.

Definition 15. A *k-code* is a function $\alpha : \mathbb{Z}_{k+1} \to \mathbb{Z}_{\geqslant 0}$ such that there exists at least one i with $\alpha(i) = 0$. The *window notation* for α is the vector $[\alpha_0, \alpha_1, \ldots, \alpha_{k-1}, \alpha_k]$. We usually identify α with its window notation.

The *diagram of a k-code* α is a Ferrer's diagram on a cylinder with $k+1$ columns, indexed by $\mathbb{Z}_{k+1}$ where the i-th column contains α_i boxes. A *k-code filling* is a marking of the diagram of α with residues from $\mathbb{Z}_{k+1}$, with the box in the ith column and jth row marked with residue $i - j$. We may *flatten* a k-code's diagram by cutting out a column j with $\alpha_j = 0$. A *reading word* of a k-code filling is obtained by reading the rows of this flattened filled diagram from right to left, beginning with the last row.

A *non-maximal k-code filling* $S = \vec{A}$ is given by a collection of subsets $\{A_1, \ldots, A_n\}$ with $A_i \subsetneq I$. The ith row of the diagram of S contains the residues in A_i.

A *k-column castle tableau* is defined similarly, but on a cylinder with $k+1$ rows marked with residues. In this case, the flattening is obtained by removing a row j with $\alpha_j = 0$. The reading word is obtained from a flattened tableau by reading the columns top-to-bottom, beginning with the right-most column.

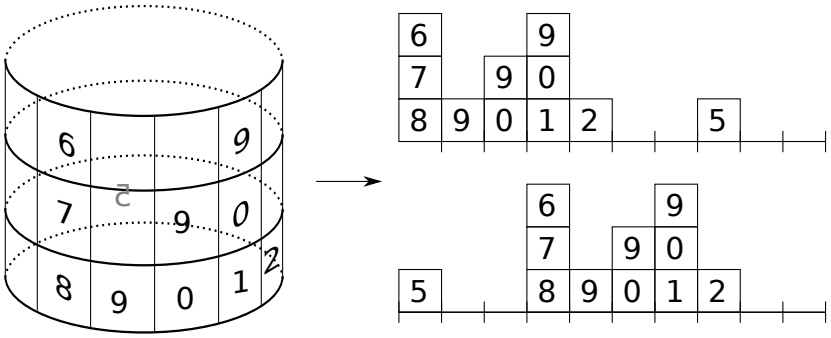


Figure 4: On the left is the diagram of the 9-core with window notation $(2, 3, 1, 0, 0, 1, 0, 0, 3, 1)$ drawn on a cylinder. On the right are two possible flattenings of this tableau. The reading word of the top flattening is $(9, 6, 0, 9, 7, 5, 2, 1, 0, 9, 8)$. The reading word of the bottom flattening is $(9, 6, 0, 9, 7, 2, 1, 0, 9, 8, 5)$. These two reading words are related by commutation relations in $\mathbb{A}_9$.

Because there is a unique k-code filling constructed from each k-code, we will commonly identify these two objects, referring to both as a k-code. We will develop an insertion algorithm in Section 3.3, which will produce two 'tableaux'. The first tableau is just the k-code filling, and the second is a 'recording tableau,' which yields a new combinatorial object in bijection with cyclic decompositions of an affine permutation. The k-shape filling may be considered as a tableau by analogy to the RSK algorithm, rather than being a chain in a poset.

Examples are provided in Figures 4 and 5.

The rows of a k-code filling each correspond to a cyclically decreasing element with the residues appearing in that row. This cyclically decreasing element is invariant under different choices of flattening for the tableau; the reading words of flattened tableaux will be related by commutation relations in $\mathbb{A}_k$.

Note that the number of boxes in a k-code filling is equal to the number of letters in the decomposition $d_{\vec{A}}$, providing a natural grading on k-codes which will correspond to the length grading on affine permutations.

We will show that k-codes are in bijection with affine permutations in Theorem 38.

3.1 Maximal Cyclic Decompositions and k-Codes

Our first objective is to show that there exists a unique maximal set $A \subsetneq I$ such that $x = yd_A$ for some y. The process is constructive, and provides a simple algorithm for finding A. Given any x, for each $i \in D_R(x)$ we find the largest possible set $D_i := [i, j] \subsetneq I$ such that:

$$x = ya_j a_{j-1} \cdots a_i.$$

We ultimately show (Lemma 18) that A is the union of the sets D_i, and is thus uniquely specified. But first we need a Lemma describing the relationship between these various

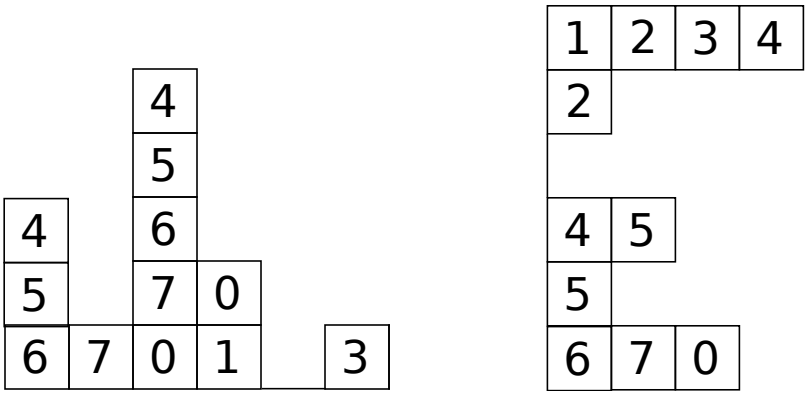


Figure 5: On the left is a flattened 7-castle tableau for the 7-castle $\alpha = (5, 2, 0, 1, 0, 0, 3, 1)$. The reading word for this tableau is $(4, 5, 6, 4, 0, 7, 5, 3, 1, 0, 7, 6)$. On the right is a 7-column bounded tableau for the k-code $\rho = (0, 4, 1, 0, 2, 1, 3, 0)$, whose reading word is $(4, 3, 0, 2, 5, 7, 1, 2, 4, 5, 6)$.

sets D_i.

Definition 16. Let x be an affine permutation, considered as an element of the nilCoxeter monoid, and $i \in D_R(x)$. Define the set $D_i = [i, j]$ to be the maximal set such that

$$x = ya_j a_{j-1} \cdots a_i,$$

for some $y \in \hat{\mathbb{A}}_k$.

Lemma 17. *Suppose x is an affine permutation, and $A = D_i, B = D_j \subsetneq I$. Then if $A \cap B \neq \emptyset$, either $A \subset B$ or $B \subset A$. Furthermore, if $A \cap B = \emptyset$ then $A \cup B$ is not connected.*

Proof. We begin by constructing sequences of residues A_0 and B_0 in the following way. Set $A_0 = (i+n, i+n-1, \ldots, i+1, i)$ with n maximal such that $x = yu_{i+n} \cdots u_i$; we may have $n > k$, so that the A_0 contains repetitions of the same index. Likewise, construct $B_0 = (p+m, p+m-1, \ldots, p+1, p)$, such that m is maximal and and $x = yu_{p+m} \cdots u_p$. Our initial goal will be to show that if B_0 and A_0 share any indicies, then we must have $A_0 \subset B_0$ or vice versa.

As x is an affine permutation, we may consider x as a doubly-infinite sequence of integers without repetitions. We set $x_l := x(l)$. Recall that if $l \in D_R(x)$, we have $x_l > x_{l+1}$, and x has a reduced word ending in l. Since $x = yu_{i+n} \cdots u_{i+1} u_i$, we have $x_i > x_{i+1}$. Removing the descent at i, we obtain $yu_{i+n} \cdots u_{i+1}$, and so this element has a right descent at $i+1$. But here x_i appears in position $i+1$, and so we have $x_i > x_{i+2}$. We may continue peeling off generators from the right to show that $x_i > x_k$ for each $k \in \{i+1, i+2, \ldots, i+n+1\}$.

Now we consider two cases.

- Case 1: $|A_0|, |B_0| \leqslant k$. Then we can set A and B to be the set of indices appearing in A_0 and B_0 respectively. If $p = D_R(d_B)$ appears in A, then $x_i > x_p$. But $x_p > x_k$ for all $k \in [p+1, p+m+1]$, so that $x_i > x_l$ for all $l \in [i+1, p+m+1]$. Thus, $B \subset A$.
- Case 2: $|A_0| \geqslant k+1$. Then $p \in A_0$, so that either $x_i > x_p$ or $x_i > x_{p+k+1}$ if $p < i$ as an integer. In the case where $x_i > x_p$, we have $x_{i+k+1} = x_i + k + 1 > x_p$, so that $i \notin B_0$. Additionally, if $i-1$ were in B_0, we would have $x_p > x_i > x_{i+1}$, which would in turn mean $i \in B_0$. Thus, $i-1, i \notin B_0$. The same reasoning holds if $p > i$ as an integer.

  As such, B_0 is a proper subset of the index set, containing neither $i-1$ or i. We set $B = \{p, p+1, \ldots, p+m\}$ and $A = \{i, i+1, \ldots, i-2\} \subsetneq I$. Then we have shown that $B \subset A$.

Thus, if $A \cap B \neq \emptyset$, we have either $B \subset A$ or vice versa.

Now suppose $A \cap B = \emptyset$ and $A \cup B$ connected as a subgraph of the Dynkin diagram, and let $A = [i, j]$ and $B = [p, q]$. If $|A|, |B| < k$, we have $p = j+1$,and $x_i > x_{j+1} = x_p > x_k$ for all $k \in [p+1, q+1]$. But then $x_i > x_k$ for all $k \in [i+1, q+1]$, so we can find C such that $A \subsetneq C$ and $x = wd_C$, contradicting the maximality of A. If $|A| = k$, we must have $|B| = 1$, so we may repeat the same argument and show that B was not maximal. Thus we have $A \cup B$ disconnected. □

Corollary 18. *For any affine permutation x, there exists a unique maximal $A \subsetneq I$ such that $x = yd_A$ with* $\mathrm{len}(x) = \mathrm{len}(y) + |A|$.

Proof. Consider $D_R(x)$. For each $i \in D_R(x)$, we can construct a maximal set $A_i = [i, j]$ for some j such that $x = y_i d_{A_i}$. By Lemma 17, if we consider any pair of these sets, they are either disjoint with their union disconnected, or one is contained in the other. Thus, the union of the A_i gives a set A such that $x = yd_A$ for some y. By construction, A is maximal.

For uniqueness, suppose B is another such set. Then $D_R(d_B) \subset D_R(x)$; by construction of A, we have either $A = B$ or $B \subsetneq A$. Then maximality of B implies $B = A$. □

Corollary 19. *For any affine permutation $x \in \mathbb{A}_k$, suppose A is the unique maximal $A \subsetneq I$ such that $x = yd_A$. Suppose $B \subsetneq I$ and $z \in \mathbb{A}_k$ such that $x = yd_A = zd_B$. Then $B \subset A$.*

Proof. This is a direct consequence of the proof of Lemma 17. □

Theorem 20. *Every affine permutation has a unique maximal decomposition into cyclically decreasing elements.*

Proof. This follows immediately by repeated application of Corollary 18. □

Remark 21 (Algorithm for Computing the Canonical Decomposition.). The proofs of these results directly translate into an algorithm for finding the canonical decreasing decomposition of any affine permutation x. For each $i \in D_R(x)$ we associate a set D_i

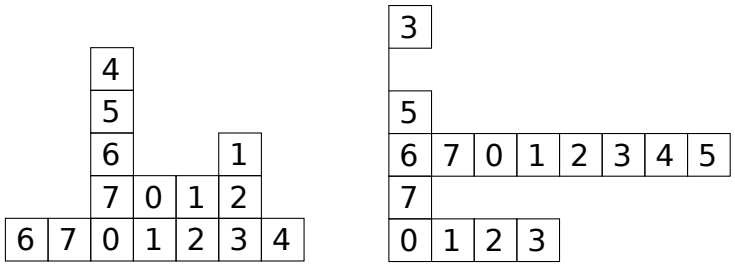


Figure 6: The maximal cyclically decreasing and increasing k-code fillings for the affine permutation with window notation $[-4, 1, 2, 0, 5, 14, 7, 11]$.

obtained by finding the largest connected cyclically decreasing word ending in u_i such that $x = yd_{D_i}$. Then set A_1 to be the union of the sets D_i, so that $x = x_1 d_{A_1}$ for some x_1. Repeat this procedure on x_1 to obtain A_2, and so on.

Example 22. Let $k = 7$. Consider the affine permutation x with base window is $[-4, 1, 2, 0, 5, 14, 7, 11]$. Then $D_R(x) = \{0, 3, 6\}$. We form the sets D_i: $D_0 = \{0, 1, 2, 3\}$, $D_3 = \{3\}$, and $D_6 = \{6, 7, 0, 1, 2, 3, 4\}$, so that their union $A_1 = D_6 = \{6, 7, 0, 1, 2, 3, 4\}$. Then we can find x_1 such that $x = x_1 d_{A_1}$.

This x_1 has base window $[1, 2, 0, 5, 6, 7, 11, 4]$. We have $D_R(x_1) = \{2, 7\}$, and find the sets $D_2 = \{2\}$ and $D_7 = \{7, 0, 1, 2\}$, so that $A_2 = \{7, 0, 1, 2\}$, and $x_1 = x_2 d_{A_2}$.

The permutation x_2 has base window $[2, 0, 3, 5, 6, 7, 4, 9]$, and $D_R(x_2) = \{6, 1\}$. Then we form the sets $D_1 = \{1\}$ and $D_6 = \{6\}$, so that $A_3 = \{1, 6\}$.

Finishing things up, one may derive $A_4 = \{5\}$ and $A_5 = \{4\}$, so that:

$$x = u_4 u_5 u_{1,6} u_{2,1,0,7} u_{4,3,2,1,0,7,6}.$$

This is the maximal decomposition of x. This is depicted as a k-code filling in Figure 6.

Using a similar algorithm, we may find a cyclically *increasing* decomposition of x. This decomposition turns out to be $x = u_5 u_4 u_3 u_2 u_{1,3} u_{0,2} u_{1,7} u_{3,5,6,7,0}$

This is depicted as a k-column castle tableau in Figure 6.

3.2 Maximizing Moves on k-Codes.

Given a non-maximal cyclically decreasing decomposition, there are a number of 'moves' we can apply in sequence to obtain the maximal decomposition. Because of the close link between decompositions and k-codes, we will develop these 'moves' in tandem in both contexts. These moves bear some similarity to moves on rc-graphs [BB93] or may be thought of as a k-bounded variation on jeu de taquin, as they may be used to obtain a k-code from a non-maximal k-code.

We first examine the action of a single generator applied to a single cyclically decreasing element d_A:

- Commutation. Suppose $i - 1, i, i + 1 \notin A$. Then $s_i d_A = d_{A \cup \{i\}}$.

- Zero. Suppose $i \in D_L(d_A)$. Then $s_i d_A = 0$.
- Braid. Suppose $i \in A, i \notin D_L(d_A)$. Then $s_i d_A = d_A s_{i+1}$.

These all follow directly from the definition of the cyclically decreasing elements and the relations in $\mathbb{A}_k$.

Now consider the product of two cyclically decreasing elements, $d_B d_A$. Using the above single-generator moves, we establish a number of 'moves' for merging elements of B into A. This allows us to maximize the vector $(|A|, |B|)$ lexicographically.

Lemma 23 (Two-Row Moves.)**.** *The following identities hold for products of cyclically decreasing elements d_A and d_B:*

- *Commutation. Suppose $i, i-1, i+1 \notin A$, and $i \in D_R(d_B)$. Then $d_B d_A = d_{B\setminus\{i\}} d_{A\cup\{i\}}$.*
- *Chute Move. Suppose $A = [i, j]$ and $B = [i-1, p]$ with $p \notin A$ and $j \in B$. Then $d_B d_A = d_{B\setminus\{j\}} d_{A\cup\{i-1\}}$.*
- *Zero. Suppose $A = [i, j]$ and $B = [p, q]$ with $p \in A$ and $j \in B$. Then $d_B d_A = 0$.*

Proof. The commutation rule follows directly from the single generator moves. The final two identities follow from applying a sequence of braid and commutation relations in the product. (And in fact, the Zero move can be derived from the Chute Move.) □

The two-row moves translate directly into operations on (skew) k-codes. In the product, d_B and d_A correspond to two stacked rows containing the residues in B and A. The two-row moves are illustrated in Figure 7.

Given a skew k-code $d_B d_A$, application of a two-row move reduces the size of B by one and increases the size of A by one. All of the two-row moves are reversible, and so we also have a set of *reverse two-row moves* which increase the size of B by one and reduce the size of A by one.

We now provide a useful technical lemma with a very nice proof!

Lemma 24. *In any product $d_B d_A \neq 0$, there exists j such that $j \notin A$, and $j - 1 \notin B$. In the two-row notation for $d_B d_A \neq 0$, there is an empty column.*

Proof. The two statements are equivalent. We suppose there is no empty column in the two-row notation, and show that the product $d_A d_B$ is unreduced.

Since there is no empty column, we have three possible states for each column.

- State TOP: $i - 1 \in B$ and $i - 2 \notin A$,
- State DOWN: $i - 1 \notin B$ and $i - 2 \in A$, or
- State CHARM: $i - 1 \in B$ and $i - 2 \in A$.

Since $B \subsetneq I$, there exists $i \notin B$; since there is no empty column, this gives $i - 1 \in A$, so there exists a TOP column. We now consider each residue j in decreasing order, beginning with $i - 1$.

If the current column is of type TOP, one of three cases holds:

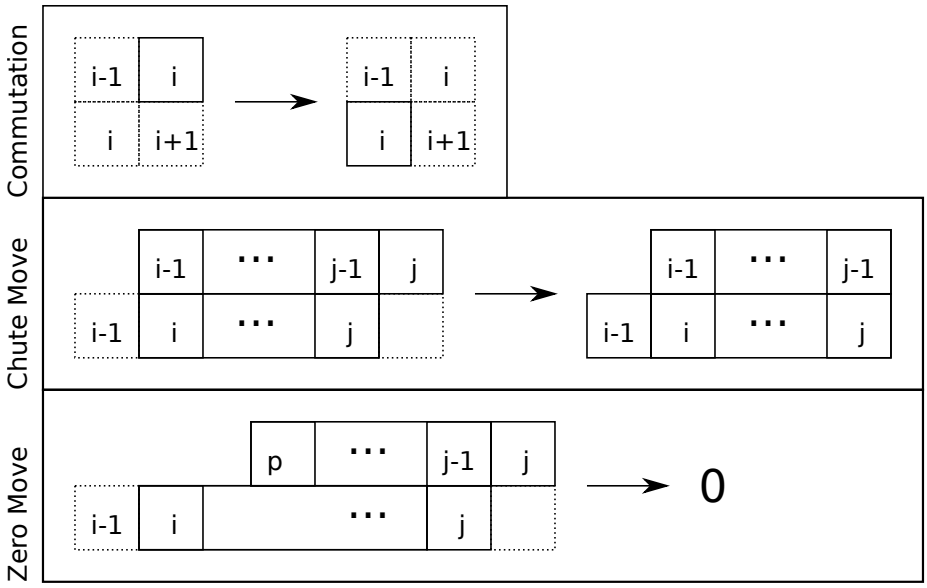


Figure 7: The two-row moves as they act on k-code fillings.

- If $j-1 \in B$ and $j-2 \notin A$, then the product $d_A d_B$ is unreduced, by the commutation two-row move.
- If $j-1 \in B$ and $j-2 \in A$, the next column if of type CHARM.
- If $j-1 \notin B$ and $j-2 \in A$, the next column if of type TOP.

So the next column is either of type CHARM or TOP.

If the current column is of type CHARM, one of three cases holds:

- If $j-1 \in B$ and $j-2 \notin A$, then the product $d_A d_B$ is unreduced, by the chute move.
- If $j-1 \in B$ and $j-2 \in A$, the next column if of type CHARM.
- If $j-1 \notin B$ and $j-2 \in A$, the next column if of type TOP.

So the next column is either of type CHARM or TOP.

So for every column, the next column is of type CHARM or TOP. Both of these cases have the residue $j-1 \in A$, so that every residue must be in A. But $A \subsetneq I$, providing a contradiction. □

For any $S = \{s_1, \ldots, s_i\} \subset I$, let $S-1$ denote the set $\{s_1 - 1, s_2 - 1, \ldots, s_i - 1\}$.

Lemma 25. *Given two sets $A, B \subsetneq I$ with $d_B d_A \neq 0$, there exist sets A', B' such that $d_B d_A = d_{B'} d_{A'}$ and $B' \subset A' - 1$. In particular, in the k-code filling for $d_{B'} d_{A'}$, every residue in B' sits directly above a residue in A'.*

Proof. We establish an explicit algorithm for maximizing the product $d_A d_B \neq 0$ using a sequence of two-row moves.

By Lemma 24, there exists a residue i such that $i \notin A, i-1 \notin B$. We set $E := i$ to be the current empty column. From the current empty column, we will read columns in increasing order. If the next column is empty, we set $E := E+1$ to be the current empty column and continue. Otherwise, we have one of three possibilities for the adjacent column:

- TOP: We apply a commutation move. The current empty column becomes of type DOWN, and the next column becomes empty. We set the current empty column to the next column.
- DOWN: We set $N = E+1$, and continue reading to the right incrementing N to keep track of the current non-empty column. If column $N+1$ is of type TOP, the product is unreduced. If column $N+1$ is empty, we set $E := N+1$ and continue. If column $N+1$ is of type DOWN, we set $N := N+1$ and continue.

  The last case is when $N+1$ is of type CHARM; in this case we keep reading (set $N := N+1$), but have a new set of possibilities. The next column $N+1$ may be of type CHARM or DOWN, either of which is ok: set $N := N+1$ and carry on. If the next column is of type TOP, then the product is unreduced by the chute move. Finally, if the next column is empty, we set that column to be the current empty column $E := N+1$ and continue.
- CHARM: We set $N := E+1$ and read columns as in the case DOWN. The only difference is that if we meet a TOP column before meeting a DOWN column, we may apply a chute move. Then the current empty column becomes of type DOWN, and the TOP column becomes empty. We set the current empty column to be the newly created empty column, and continue.

In all cases where a box is moved, a box moves from the top row to the bottom row. This implies that this process must stabilize at some point. In all cases we eliminate columns of type TOP, so that the final expression will contain no TOP columns. Thus, $A' \subset B' - 1$. □

Example 26. Let $k = 9$, with $A = \{2,3,5,8\} and B = \{0,1,2,3,4,7,8\}$. We find A', B' such that the product $d_{B'}d_{A'}$ is maximal. We apply a series of moves:

$$\begin{aligned} d_B d_A &= (a_{87}a_{43210})(a_8 a_5 a_{32}) \\ &= (a_{87}a_{4321})(a_8 a_5 a_{32} a_0)(\text{commutation}) \\ &= (a_{87}a_4 a_{21})(a_8 a_5 a_{3210})(\text{chute move}) \\ &= (a_7 a_4 a_{21})(a_{87} a_5 a_{3210})(\text{chute move}). \end{aligned}$$

Thus, we have $A' = \{0,1,2,3,5,7,8\}$ and $B' = \{1,2,4,7\}$.

For a product of more than two cyclically decreasing elements $d_{\vec{A}}$, we may progressively apply two-row moves to pairs of adjacent cyclically decreasing elements, eventually obtaining a decomposition with $A_{i+1} \subset A_i - 1$ for each i. Such a decomposition can be

represented by a k-code fillings by selecting in row i the residues in A_i. Thus, we have the following Lemma:

Proposition 27 (Maximal Cyclic Products)**.** *For any $w \in \mathbb{A}_k$, if $w = d_{\vec{A}}$ is a maximal expression for w as a cyclically decreasing product then $\vec{A}$ has $A_{i+1} \subset A_i - 1$ for each i. In particular, we observe that* $\mathrm{sh}(\vec{A})$ *is a partition.*

Thus, we have shown that any reduced decomposition may have a series of two-row moves applied to it to obtain a decomposition corresponding to a k-bounded tableau.

3.3 Insertion Algorithm.

Consider x an affine permutation with $x = d_{\vec{A}}$, $\vec{A} = \{A_1, \ldots, A_n\}$, giving the maximal decomposition of x. We consider the product xa_p for $p \notin D_R(x)$, and find an algorithm for determining the k-code filling T' for xa_p. To do this, we attempt to insert the residue p into the set A_j, beginning with $j = 1$. One of following possibilities occurs:

- (Inclusion I.) If $p-1, p, p+1 \notin A_j$: By the commutation relation, we may include p into A_j. Include p into A_j and halt the algorithm.
- (Inclusion II.) If $p-1, p \notin A_j$, but $p+1 \in A_j$: We have $p+1 \in D_R(A_j)$, and may include p into A_j. So again, include p into A_j and halt the algorithm.
- (Bump Move.) If $p \notin A_j$ and $p-1 \in A_j$, we have $d_{A_j} u_p = u_{p-1} d_{(A_j \setminus \{p-1\}) \cup \{p\}}$. In other words, bump the residue $p-1$ from A_j and replace it with the residue p.
- (Braid Move.) If $p \in A_j$ but $p \notin D_R(A_j)$, then $d_{A_j} u_p = u_{p-1} d_{A_j}$ by a braid relation. In this case, leave A_j unchanged and continue the process, trying to insert $p-1$ into A_{j+1}.

These cover all possibilities. When the product is non-zero, this gives us a way to insert a new box into the k-code.

We remove the explanations from the different cases to obtain a reduced list of insertion moves:

- (Inclusion.) If $p-1, p \notin A_j$: Include p into A_j and halt the algorithm.
- (Bump Move.) If $p \notin A_j$ and $p-1 \in A_j$, Remove the residue $p-1$ and include residue p in A_j. Continue the insertion with the residue $p-1$ into row $j+1$.
- (Braid Move.) If $p \in A_j$ but $p \notin D_R(A_j)$: Leave row A_j unchanged, and continue the insertion algorithm with residue $p-1$ into row $j+1$.

Definition 28. Let S be a k-code and p a residue. We denote the insertion of p into S by $S \leftarrow p$.

Notice that in both the braid move and the bump move, the residue $p-1$ is in the (possibly modified) $A_j - 1$. As a result, inserting a residue i into a k-code will produce another k-code, so long as the product $xu_i \neq 0$. Luckily, we can use Corollary 36 to read

off the right descents of x from its k-code, making it immediately clear whether a given value can be inserted or not.

We may form a *recording tableau* Q in the usual way. Suppose $w = [w_1, \ldots, w_n]$ is a word in the alphabet I which inserts to a k-code P. On inserting the jth letter of w, we write a j in the final box in the insertion of $[w_1, \ldots, w_j]$. The only special case is the bump move, which replaces the box with residue p with the box with residue $p+1$. Suppose the residue p box was marked with an l in the recording tableau: We simply put this l in the box with residue $p+1$ and delete the box with residue p. (This is illustrated in Figure 8)

Any reduced word w for a given permutation x may be inserted to the empty k-code to obtain a tableau Q which depends on the reduced word that was inserted. In fact, all of the insertion moves are invertible, allowing a reverse insertion algorithm. Then given a recording tableau Q one may recover the reduced expression w.

Theorem 29. *Let $\mathfrak{Q} = \{Q_i\}$ be the set of recording tableaux associated to a k-code α obtained from a maximal decomposition of an affine permutation x. Then $\mathfrak{Q}$ is in bijection with the set of reduced words for x.*

Call a recording tableau Q *standard* if it arises as the recording tableau of some reduced expression for an affine permutation. Then it is clear that there is a bijection between standard recording tableaux of a given shape and reduced expressions for the affine permutation with the associated k-code.

Problem 30. Find a combinatorial description of the recording tableaux.

Example 31. At $k = 1$, there are no relations between the generators u_0 and u_1. In this case, there are exactly two k-codes of size n ($(n, 0)$ and $(0, n)$), and exactly two non-zero words on n letters, one with right descent 0 and one with right descent 1.

Example 32. With $k = 3$, let $w = [0, 3, 1, 2, 1, 0]$. Then the insertion of w is the k-code $\alpha = (2, 1, 3, 0)$. But the recording tableau has first row $[2, 5, 1]$, which is not standard in the usual sense for tableaux.

3.4 Bijection Between Affine Permutations and k-Codes.

Our goal in this section is to prove the bijection between k-codes and affine permutations. We begin by restating the results of Theorem 20 and Proposition 27 in a consolidated statement:

Theorem 33 (Canonical Cyclically Decreasing Decomposition.)**.** *Every affine permutation x admits a unique maximal decomposition as a product of cyclically decreasing elements $x = d_{\vec{A}}$. This decomposition has $A_{i+1} \subset A_i - 1$ for each i, and thus $\mathrm{sh}(\vec{A})$ is a partition.*

Proof. This follows from repeated application of Corollary 18 to obtain a complete decomposition of the affine permutation x as a product of maximal cyclically decreasing elements. By construction, this decomposition is maximal. It must also satisfy $A_{i+1} \subset A_i - 1$

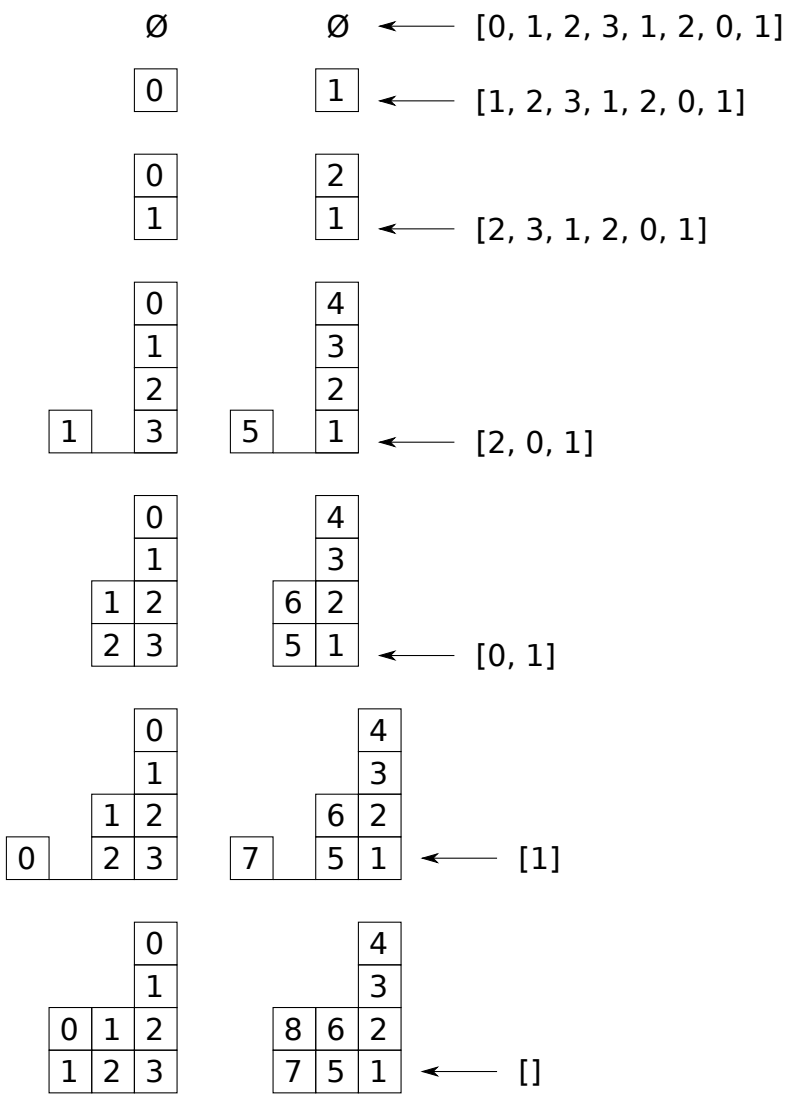


Figure 8: The insertion algorithm for $k = 4$ with the word $w = [0, 1, 2, 3, 1, 2, 0, 1]$. The left tableaux are the k-code fillings obtained at each step, and the right tableaux are the recording tableaux.

for each i, or else we could apply a two-row move to obtain a new decomposition greater in lexicographic order. □

Definition 34. We refer to the maximal decomposition of x as the *canonical decreasing decomposition* of x, denoted $\mathrm{RD}(x)$. The corresponding maximal decomposition into cyclically increasing elements is the *canonical increasing decomposision* of x, denoted $\mathrm{RI}(x)$.

We define a map σ from affine permutations to k-code fillings. For x an affine permutation with canonical decreasing decomposition $x = d_{\vec{A}}$, take $\sigma(x)$ to be the k-code filling whose ith row is given by the set of residues A_i.

Definition 35. A *descent* of a k-code α is an index i such that $\alpha_{i-1} < \alpha_i$.

Corollary 36 (Descent Sets from k-code fillings.)**.** *Given a maximal k-code filling $T = \sigma(x)$ for an affine permutation $x = d_{\vec{A}}$, then $r \in D_R(x)$ if and only if r appears in the first row of T and the column containing this box contains a right descent for one of the d_{A_i}.*

Proof. These descents occur by repeated use of the braid relation to move a right descent in d_{A_i} to the beginning of a reduced expression for x. □

Corollary 37. *Let x be an affine permutation with decomposition $x = d_{\vec{A}}$. Then this decomposition is maximal if and only if $A_{i+1} \subset A_i - 1$ for every i.*

Proof. The forward direction is given by Lemma 27.

On the other hand, if $A_{i+1} \subset A_i$ for all i, we may apply the algorithm in Remark 21 to obtain a maximal decomposition $x = d_{\vec{B}}$. We can also associate a k-code filling T to the decomposition $d_{\vec{A}}$. The algorithm constructs sets D_j for each $j \in D_R(x)$ and takes B_1 to be the union of the sets D_j. By Corollary 36, we may then observe that $A_1 = B_1$. We may then repeat this process to show that $A_i = B_i$ for every i. Thus, $d_{\vec{A}}$ is the maximal decomposition of x. □

Theorem 38. *The set of k-codes is in bijection with affine permutations in $\tilde{S}_{k+1}$.*

Proof. The map σ takes permutations of length n to k-codes with n boxes, so we may consider σ as a graded map on finite sets. Additionally, we may also recover x by taking the reading word of $\sigma(x)$. By Corollary 37, for any $x \neq y$, we have $\sigma(x) \neq \sigma(y)$, so σ is one-to-one. Then we only need to show that every k-code T is a maximal decomposition of some affine permutation; equivalently, that element obtained by the reading word of T is non-zero in $\mathbb{A}_k$.

For this, we induct on the number of boxes n in T. At $n = 1$, the single box corresponds to a simple transposition, and the statement holds. Suppose that S is the tableau of shape α with $n+1$ boxes and $\alpha_{j-1} = 0, \alpha_j \neq 0$. Then j is a descent of S. Removing the box j, we may apply a sequence of commutation two-row moves to remove one box from column j and shift it to position $j-1$. Since S was a k-code, the resulting S' is also a k-code, but on n boxes. As such, it is equal to $\sigma(x)$ for some x. This x has $j \notin D_R(x)$, so $xu_j \neq 0$. Reinserting j into S' yields S, so we see that $\sigma(xu_j) = S$. This completes the proof. □

Corollary 39. *Consider an affine permutation x with k-code filling T. Then x is i-dominant if and only if T has a flattening which is a k-bounded partition with residue i in the lower left box.*

Proof. This follows immediately from Corollary 36. □

Note that this gives an alternate proof of the bijection between k-bounded partitions and 0-dominant (or Grasssmannian) elements.

This allows us to prove a theorem on the nil-Coxeter realization of the k-bounded symmetric functions.

We complete this subsection with a simple statement relating the k-codes to symmetric functions in non-commuting variables.

Theorem 40. *Suppose that* $\mathrm{sh}(\sigma(x)) = \lambda$. *Then:*

$$[x]h_\lambda = 1, \text{ and } [x]s_\lambda^{(k)} = 1.$$

Proof. In both h_λ and $[x]s_\lambda^{(k)}$, all coefficients are integers $\geqslant 0$. We have $[x]h_\lambda = 1$ by the uniqueness of the cyclic decomposition of x. Furthermore, $h_\lambda = s_\lambda^{(k)} + \sum_\mu s_\mu^{(k)}$, where each μ dominates λ. If we had some μ with $[x]s_\mu^{(k)} = 1$, we could then have $[x]h_\mu = 1$, contradicting the maximality of the decomposition of x. □

3.5 Relating the Various Cyclic Decompositions of an Affine Permutation.

The constructions of this section may be modified to provide four different k-codes associated to any affine permutation x. These are obtained by finding maximal cyclically increasing and decreasing decompositions for x from the right and from the left. The decomposition from the left finds $x = d_{A_n} d_{A_{n-1}} \cdots d_{A_1}$ maximizing $(|A_n|, |A_{n-1}|, \ldots, |A_1|)$ lexicographically. This may be found by modifying the algorithm for generating a k-code to consider the left descents of x instead of the right descents.

Definition 41. Let x be an affine permutation. Set $\mathrm{RD}(x)$ to be the k-code corresponding to the maximal right decomposition of x into a product of cyclically decreasing elements. Likewise, set $\mathrm{RI}(x)$ to be the k-code from the right increasing decomposition, and $\mathrm{LD}(x)$ and $\mathrm{LI}(x)$ be the k-codes from the left decreasing and increasing decompositions, respectively.

Example 42. Let $k = 3$, and $x = a_{2,1,0,3,0,1,2,1,0,3,1,2,0,1,0}$. Then x has the following maximal cyclic decompositions:

- Decreasing Right:
$$x = a_2 a_3 a_0 a_1 a_{3,2} a_{0,3,2} a_{1,0,3} a_{2,1,0},$$
so $\mathrm{RD}(x) = (3, 8, 4, 0)$.

- Increasing Right:
$$x = a_2a_1a_0a_3a_2a_1a_0a_3a_{2,3}a_{1,2}a_{3,0,1},$$
so $\mathrm{RI}(x) = (11, 3, 0, 1)$.
- Decreasing Left:
$$x = a_{2,1,0}a_{3,2,1}a_{0,3,2}a_{3,1}a_2a_3a_0a_1,$$
so $\mathrm{LD}(x) = (4, 3, 8, 0)$.
- Increasing Left:
$$x = a_{2,3,1}a_{1,3}a_{0,2}a_3a_2a_1a_0a_3a_2a_1a_0,$$
so $\mathrm{LI}(x) = (3, 0, 11, 1)$.

An alternative way to produce $\mathrm{LD}(x)$ from $\mathrm{RD}(x)$ is to use the reverse two-row moves to 'up-justify' $\mathrm{RD}(x)$. The resulting object's reading word gives the left decomposition of x into cyclically decreasing elements.

We can establish a more direct relationship between $\mathrm{LD}(x)$ and $\mathrm{RD}(x)$.

Proposition 43. $\mathrm{RD}(x)$ *is a permutation of* $\mathrm{LD}(x)$*, and* $\mathrm{RI}(x)$ *is a permutation of* $\mathrm{LI}(x)$*.*

Proof. Suppose $x = d_{\vec{A}}$ with $\mathrm{sh}(\vec{A}) = \lambda$. Recall that $h_\lambda = h_{\lambda_n} \cdots h_{\lambda_1}$. By Theorem 40, we have $[x]h_\lambda = 1$, corresponding to the unique maximal cyclically decreasing decomposition of x. But because the h_i's commute, we have $[x]h_{\lambda_1} \cdots h_{\lambda_n} = 1$. Since x appears in $h_{\lambda_n} \cdots h_{\lambda_1}$, there exists a cyclically decreasing decomposition for $x = d_{\vec{B}}$ of shape $(\lambda_n, \ldots, \lambda_1)$. This decomposition is maximal as a left cyclically decreasing decomposition, or else commutativity of the h_i's would imply that our original decomposition of x was not maximal. Then $\mathrm{sh}(\vec{B})$ is the reverse of λ, and $B_{i-1} \subset B_i - 1$ for each i, implying that the entries in $\mathrm{LD}(x)$ are the same as the entries in $\mathrm{RD}(x)$, up to some reordering.

The proof for the increasing case is identical. □

Problem 44. Describe the permutation relating $\mathrm{RD}(x)$ and $\mathrm{LD}(x)$ for arbitrary x. Is there a straightforward way to calculate the permutation, short of directly computing the left maximal decomposition?

Lemma 45. *For any affine permutation* x*, the descent sets of* $\mathrm{RI}(x)$ *and* $\mathrm{RD}(x)$ *are equal. Also, the descent sets of* $\mathrm{LI}(x)$ *and* $\mathrm{LD}(x)$ *are equal.*

Proof. The descent sets of $\mathrm{RI}(x)$ and $\mathrm{RD}(x)$ are equal to $D_R(x)$, by Corollary 36. One can prove an analog of Corollary 36 for the left decompositions, giving the second statement. □

Recall that the inverse of an affine permutation x is obtained by reversing any reduced word for x.

Proposition 46. *Let* x *be an affine permutation. Then* $\mathrm{RD}(x^{-1}) = \mathrm{LI}(x)$ *and* $\mathrm{LD}(x^{-1}) = \mathrm{RI}(x)$*.*

Proof. The element x^{-1} is obtained by reversing a reduced expression for x. The reversal of a cyclically decreasing element is a cyclically increasing element, and vice versa. Thus, reversing the cyclic decomposition immediately converts the maximal decreasing right decomposition for x into the maximal increasing left decomposition for x (which coincides with the maximal right increasing decomposition of x^{-1}). □

3.6 Affine Codes and k-Codes

The various k-codes associated to an affine permutation relate directly to the affine code derived from considering x as a permutation of the integers. There are various ways to construct the code of a permutation in the finite case; we directly generalize four methods and place them in correspondence with the k-codes. Two of these methods correspond to the affine code of the permutation, and two correspond to the inversion vector. We unify the two concepts by referring to both as simply the code of the permutation.

Definition 47. An *affine code* is given by a vector with $k+1$ entries, $L = \{L_1, \ldots, L_{k+1}\}$. Four different affine codes are described below, by providing an algorithm for finding the ith entry of the code for an affine permutation x.

- CRD: The *right decreasing code* is given by the number of $j < i$ with $x(j) > x(i)$.
- CRI: The *right increasing code* is given by the number of $j > (i+1)$ with $x(j) < x(i+1)$.
- CLD: The *left decreasing code* is given by the number of $j < x^{-1}(i+1)$ with $x(j) > i+1$.
- CLI: The *left increasing code* is given by the number of $j > x^{-1}(i)$ with $x(j) < i$.

Example 48. Consider the affine permutation x with $k = 3$ and window notation $[1, -6, 0, 15]$. Then the CRD is $(3, 8, 4, 0)$: for example, 11, 7 and 3 all appear to the left of $1 = x(1)$, so that the first entry of $\mathrm{CRD}(x)$ is 3. This matches the k-code for this element, described in Example 42.

Proposition 49. *The affine codes described above are equal to the respective k-codes for an affine permutation.*

Proof. We show that $\mathrm{CRD}(x) = \mathrm{RD}(x)$; the other three equalities follow similar logic. Let $L := \mathrm{CRD}(x) = (L_1, \ldots, L_{k+1})$, and let $K := \mathrm{RD}(x) = (K_1, \ldots, K_{k+1})$. We show that $L = K$.

When $x = d_A$ for some A, the statement is clear: For any $i \in I$, if $i \notin A$ then there are no larger elements to the left of position i, so $L_i = 0$, as will K_i. If $i \in A$, the transposition at positions $i, i+1$ moves exactly one large element to the left of position i, so that $L_i = 1$. Likewise, $K_1 = 1$ in this case, so the base case holds.

For the induction step, let $x = yd_A$ be a reduced product, with A maximal. By induction, we have $\mathrm{CRD}(y) = \mathrm{RD}(y)$. By inspection, applying d_A to y either increases by one or stabilizes each entry L_i in $\mathrm{CRD}(y)$, according to whether $i \in A$. Then the proposition holds. □

3.7 Grassmannian and i-Dominant Elements.

A special case, important in the study of k-Schur functions, occurs when an affine permutation x has $D_R(x) \subset \{i\}$ for some $i \in I$. When $i = 0$, x is called a *Grassmannian element*, and otherwise it is known as an i-dominant element. By Corollary 36 these are given by the k-codes with (at most) a single decent at position i; a flattening of such a k-code is a k-bounded partition.

The following result is known within the community (in particular to the authors of [BBTZ11]), but the author has been unable to find a reference. We state the result here as a corollary of the k-code construction.

Corollary 50. *Let w_λ be the 0-dominant element in the expansion of $s_\lambda^{(k)}$. Then w_λ has a unique reduced decomposition as a maximal cyclically decreasing product, where the i-th cyclically decreasing element has length λ_i. This word is obtained by writing the diagram of the k-bounded partition of λ and marking the $k+1$-residues in each box, and then reading the rows of the resulting tableau right-to-left. In other words, if λ has n parts,*

$$w_\lambda = \prod_{i=1}^{n} d_{[-n+i,-n+i+\lambda_{n-i}-1]},$$

where the subscripts are considered modulo $k+1$.

An identical argument allows one to find a reduced word for w_λ as a maximal cyclically increasing product. To find this reading word, consider the bijection between k-bounded partitions and $k+1$-cores. The k-bounded partition is obtained by removing all boxes with hook $> k+1$ from the $k+1$-core, and then "left-justifying" the resulting skew shape (called the k-boundary of the core). To obtain a cyclically increasing word for w_λ, one instead "down-justfies" the k-boundary to obtain a partition whose columns are all k-bounded. Fill the boxes of this partition with $k+1$ residues, and then read the columns top-to-bottom, right-to-left.

Let λ be a k-bounded partition. Then the bijection between k-bounded partitions and $k+1$ cores yields a core μ. The bijection between $k+1$-cores and k-column bounded partitions gives us a k-column bounded partition ν. To all of these things, there is a 0-dominant element $w \in \mathbb{A}_k$. We can read off the maximal cyclically decreasing product for w from λ, and the maximal cyclically increasing product for w from ν.

In particular, we can convert very quickly between maximal cyclically increasing and decreasing expressions for w.

If we wish to find an i-dominant maximal cyclically decreasing (resp. increasing) word, we can simply add i to all the residues in λ (resp. ν); this is equivalent to applying the Dynkin diagram automorphism i times to the word for the 0-dominant element w.

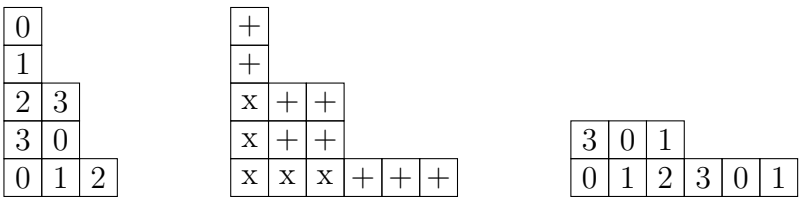
Suppose x is an i-dominant affine permutation with k-code $\alpha = \mathrm{RD}(x)$ which flattens to the k-bounded partition λ. Then x also has a k-column castle $\beta = \mathrm{RI}(x)$, which also has (at most) one descent, and is thus also associated to a k-bounded partition μ. These two partitions are related by an operation called the k-conjugate. There is a bijection $\mathfrak{c}$ from k-bounded partitions to $k+1$ cores, which are partitions containing no hooks of

length $k+1$. Denote the core associated to a partition ν by $\mathfrak{c}(\nu)$, and the conjugate of a partition by ν^t. Then the k-conjugate of λ is defined to be $\lambda^{(k)} := \mathfrak{c}^{-1}((\mathfrak{c}(\lambda))^t) = \mu$.

We summarize this discussion in the following proposition:

Proposition 51. *Let x be a 0-dominant affine permutation, associated to k-bounded partition λ with column heights given by $\lambda' = (\lambda'_1, \ldots, \lambda'_k)$, some of which may be zero. Then the k-code $\mathrm{RD}(x) = (\lambda'_1, \ldots, \lambda'_k, 0)$. Furthermore, if ν is the k-conjugate of λ, with column heights $(\nu'_1, \ldots, \nu'_k)$, we have $\mathrm{RI}(x) = (\nu'_1, \ldots, \nu'_k, 0)$.*

Example 52. Consider the 3-bounded partition $\lambda = (3, 2, 2, 1, 1)$. Below we see λ, the associated $(3+1)$-core, and 3-column bounded partition:



Then the maximal cyclically decreasing decomposition for w_λ is $a_0a_1a_{3,2}a_{3,0}a_{2,1,0}$, obtained by reading the residues in the rows from right to left, top to bottom. The maximal cyclically increasing decomposition is $a_1a_0a_3a_{1,2}a_{0,1}a_{3,0}$.

The constructions of this section provide a natural generalization of the k-conjugate to arbitrary affine permutations.

Definition 53. If x has $\mathrm{RD}(x) = \alpha$ and $\mathrm{RI}(x) = \beta$, then we say that α and β are *k-conjugates*, and write $\alpha^{(k)} = \beta$. Additionally, we define the k-conjugate of x to be the affine permutation $x^{(k)}$ with $\mathrm{RD}(x^{(k)}) = \beta$ and $\mathrm{RI}(x^{(k)}) = \alpha$.

The following proposition is then immediate.

Proposition 54. *The k-conjugate induces an involution on the affine symmetric group. This involution preserves length and right descent sets of affine permutations.*

3.8 Generalized Pieri Rule.

There is a combinatorial Pieri Rule on k-bounded partitions which corresponds to the Pieri Rule for k-Schur functions.

In this subsection, we generalize the combinatorial Pieri Rule to k-codes, and general affine permutations. First, we establish the notion of skew k-codes.

Definition 55. Let α and β be k-codes. We say that β *contains* α, $\alpha \subset \beta$, if $\alpha_i \leqslant \beta_i$ for every $i \in I$. We define a *skew k-code* to be a pair (β, α) where $\alpha \subset \beta$. The *tableau* of a skew k-code is the k-code filling of β with all boxes from α removed.

We say that a skew-k-code is a *horizontal strip* if the tableau of (β, α) contains no more than one box in each column. Likewise, (β, α) is a *vertical strip* if its tableau contains no more than one box in each row.

Proposition 56. *Let x and y be affine permutations with $xy \neq 0$. Then* $\mathrm{RD}(y) \subset \mathrm{RD}(xy)$.

Proof. Given the k-codes $\mathrm{RD}(x), \mathrm{RD}(y)$ we may obtain $\mathrm{RD}(xy)$ by stacking the two castle tableaux appropriately and applying a sequence of two-row moves to obtain a maximal decomposition of xy. In the application of two-row moves, the lower row is always preserved as columns of type TOP are eliminated. Since $\mathrm{RD}(y)$ contains no pairs of adjacent rows of type TOP, we then observe that the k-code of y is preserved as we maximize the product xy to obtain $\mathrm{RD}(xy)$. □

Theorem 57 (Generalized Pieri Rule)**.** *Let x be an affine permutation with maximal right decomposition $x = d_{\vec{A}}$ and k-code $\alpha = \mathrm{RD}(x)$. Let $B \subsetneq I$. Suppose the product $d_B x \neq 0$. Then the skew composition $(\mathrm{RD}(d_B x), \mathrm{RD}(x))$ is a horizontal strip and the skew composition $(\mathrm{RI}(d_B x), \mathrm{RI}(x))$ is a vertical strip.*

Proof. We see that $\mathrm{RD}(x) \subset \mathrm{RD}(d_B x)$ by Proposition 56.

It is easier to show that the skew composition $(\mathrm{LD}(d_B x), \mathrm{LD}(x))$ has no more than one box in each column. The result then follows from Proposition 43, which states that the columns of $\mathrm{LD}(d_B x)$ are a permutation of the columns of $\mathrm{RD}(d_B x)$. Furthermore, showing that $(\mathrm{LD}(d_B x), \mathrm{LD}(x))$ has no more than one box in each column is equivalent to showing that $(\mathrm{RD}(x d_B), \mathrm{RD}(x))$ has no more than one box in each column. Thus, we will focus on proving this statement.

To prove this statement, we stack $\mathrm{RD}(x)$ on $\mathrm{RD}(d_B)$ to form a skew k-code (β, α), and maximize the product using two-row moves. We note that by Lemma 24 there must be an empty column in (β, α) or else the product would not be reduced. One may then use an algorithm similar to the algorithm in the proof of Lemma 25 to maximize the product and obtain $\mathrm{RD}(x d_B)$. When $i \in B$ but $i - 1 \notin A_1$, (so we have a TOP state), we can use a sequence of two-row moves to move an entire column of d_A downward. The two types of move needed - iterated commutation moves and iterated chute moves, are illustrated in Figure 9. Otherwise, the algorithm is exactly as in Lemma 25.

The proof that $(\mathrm{RI}(d_B x), \mathrm{RI}(x))$ is a vertical strip is similar. □

4 Multiplication of Cyclically Increasing Elements by Cyclically Decreasing Elements.

In this section we investigate products of cyclically increasing with cyclically decreasing elements. We focus in particular on product $u_B d_A$ with connected $A, B \subsetneq I$, since non-connected cyclic elements are commutative products of connected elements.

Lemma 58. *Let $A, B \subsetneq I$ with $|B| + |A| \geqslant k + 1$. Then $u_B d_A$ is i-dominant if and only if A is i-dominant and B is $(i-1)$-dominant.*

Proof. Assume $u_B d_A$ is i-dominant. Then it is clear that A must be i-dominant (and thus connected).

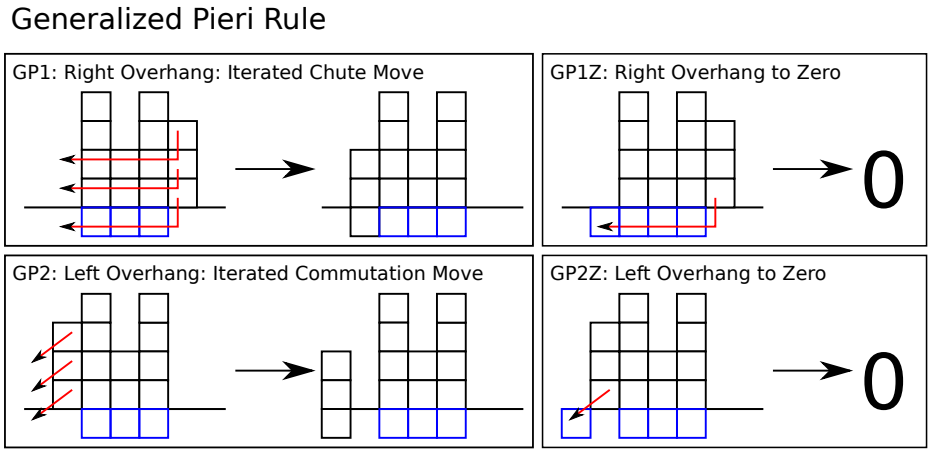


Figure 9: Multi-row moves used in the proof of the Generalized Pieri Rule, for finding the k-code for the product xd_B. The blue boxes represent residues in d_B, and the black boxes represent the k-code $\mathrm{RD}(x)$. The empty column appears on the left; reading to the right, we may move down columns appearing over an empty space by applying iterated commutation or chute moves.

Let $A = \{i, i+1, \ldots, l\}$. In order for the product $u_B d_A$ to be i-dominant, we must have $D_R(B) \subseteq \{i-1, l+1\}$, and thus B has at most two connected components. However, since u_B is cyclically increasing, if there is a component with right descent $l+1$, that component must have cardinality one, or else there will be a braid relation in the product $u_B d_A$ creating a right descent at $l+1 \neq i$.

Now if $|B| + |A| > k+1$, we have $B \cap A \neq \emptyset$. We have $|A| < k+1$, so $|B| \geqslant 2$. In this case, B must have a single connected component because the component with right descent $i-1$ must be large enough to overlap with A, since the component with right descent $l+1$ has cardinality 1. But then $l+1$ is in the component with right descent $i-1$, implying that there was only one component to begin with.

If $|B| + |A| = k+1$, by similar reasoning, we have $B = I \setminus A$, and is thus connected with right descent $i-1$.

For the reverse direction, we associate with B the k-bounded partition $(1^{|B|})$, and we associate with A the k-bounded partition $(|A|)$. Then we consider the product $e_{|B|}h_{|A|} = s^{(k)}_{(1^{|B|})} s^{(k)}_{(|A|)}$. By the forward direction, any 0-dominant element in this product is of the form $u_{B'} d_{A'}$ for some $A', B' \subsetneq I$, where A' is 0-dominant, B' is k-dominant, $|A'| = |A|$, and $|B'| = |B|$. (If these last two conditions did not hold, we could find such an expression for the element because $u_{B'} d_{A'}$ is a summand in the product $e_{|B|}h_{|A|}$.) But this specifies A' and B' completely. Thus, there is only one summmand in $e_{|B|}h_{|A|}$ when expressed as a sum of k-Schur functions. As such, there is only one i-dominant term, and it may be obtained by applying the Dynkin diagram automorphism i times. This exactly recovers the sets A and B, and implies that the product $u_B d_A$ is reduced and i-dominant. □

In proving this Lemma, we have also proved the following Corollary:

Corollary 59. *Suppose λ splits into two partitions (1^p) and (q), with $p+q \geqslant k+1$. Then*

$$s_\lambda^{(k)} = s_{(1^p)}^{(k)} s_{(q)}^{(k)}.$$

Corollary 60. *If $u_B d_A$ is i-dominant with $|B| + |A| \geqslant k+1$, we have:*

$$u_B d_A = d_{A+1} u_{B+1},$$

where $S+1$ is the set obtained by adding 1 to each element of $S \subset I$. In particular, u_B is $i-1$-dominant.

Example 61. This calculation is easiest to see with a particular example; the general case is identical. Suppose $I = \{0,1,2,3,4,5,6\}$ and $A = \{0,1,2,3,4\}, B = \{3,4,5,6\}$. Then:

$$\begin{aligned} u_B d_A &= a_{3456} a_{43210} \\ &= a_{345643210} \\ &= a_{345432160} \\ &= a_{543452160} \\ &= a_{543214560} \\ &= a_{54321} a_{4560} \\ &= d_{A+1} u_{B+1}. \end{aligned}$$

Proof of Corollary. This follows directly from the Lemma and a simple computation. By the Lemma, $A = \{i, i+1, \ldots, l\}$ and $B = \{j, j+1, \ldots, i-1,\}$ are connected, and so $D_R(u_B) = \{i-1\}$. Then:

$$u_B d_A = (a_j a_{j+1} \cdots a_{i-2} a_{i-1})(a_l a_{l-1} \cdots a_{i+1} a_i).$$

If $|B| + |A| = k+1$, we have $j = l+1$. Then using the commutation relations:

$$\begin{aligned} u_B d_A &= (a_{l+1} a_{l+2} \cdots a_{i-2} a_{i-1})(a_l a_{l-1} \cdots a_{i+1} a_i) \\ &= (a_{l+1} a_l a_{l-1} \cdots a_{i+1})(a_{l+2} \cdots a_{i-2} a_{i-1} a_i) \\ &= d_{A+1} u_{B+1}. \end{aligned}$$

If $|B| + |A| > k+1$, then B and A must overlap. Thus we have $l \in B$. In the following computation, we use a sequence of subscripts to indicate the product of a_i's. (So, for example, $a_{1,2,3} = a_1 a_2 a_3$.) (The computation uses the extended braid relation, Lemma 5.)

Then:

$$\begin{aligned}
u_B d_A &= a_{j,j+1,\cdots,l,l+1,l+2,\cdots,i-2,i-1} a_{l,l-1,\cdots,i+1,i} \\
&= a_{j,j+1,\cdots,l,l+1} a_{l+2,\cdots,i-2,i-1} a_{l,l-1,\cdots,i+1} a_i \\
&= a_{j,j+1,\cdots,l,l+1} a_{l,l-1,\cdots,i+1} a_{l+2,\cdots,i-2,i-1} a_i \\
&= a_{j,j+1,\cdots,l,l+1} a_{l,l-1,\cdots,j+1,j,j-1,\cdots,i+1} a_{l+2,\cdots,i-2,i-1} a_i \\
&= a_{j,j+1,\cdots,l,l+1,l,\cdots,j+1,j} a_{j-1,\cdots,i+1} a_{l+2,\cdots,i-2,i-1} a_i \\
&= a_{l+1,l,\cdots,j+1,j,j+1,\cdots,l,l+1} a_{j-1,\cdots,i+1} a_{l+2,\cdots,i-2,i-1} a_i \\
&= a_{l+1,l,\cdots,j+1,j} a_{j+1,\cdots,l,l+1} a_{j-1,\cdots,i+1} a_{l+2,\cdots,i-2,i-1} a_i \\
&= a_{l+1,l,\cdots,j+1,j} a_{j-1,\cdots,i+1} a_{j+1,\cdots,l,l+1} a_{l+2,\cdots,i-2,i-1,i} \\
&= d_{A+1} u_{B+1}.
\end{aligned}$$

This completes the proof. □

4.1 Products of Cyclically Increasing and Decreasing Elements

We catalog the result of multiplying $u_B d_A$ for any connected $A, B \subsetneq I$. First we fix some notation.

Definition 62. Let $B \subsetneq I$ be connected, with $B = \{i, i+1, \ldots, j-1, j\}$. Set:

$$\begin{aligned}
B^+ &= B \cup \{j+1\} \\
B^- &= B \setminus \{j\} \\
B_+ &= B \cup \{i-1\} \\
B_- &= B \setminus \{i\}
\end{aligned}$$

Additionally, let the sets with both subscripts and superscripts be defined in the obvious way. (So that $B^+_- = (B \cup \{j+1\}) \setminus \{i\}$, for example.)

Lemma 63. *Let $B \subsetneq I$ be connected, with $B = \{i, i+1, \ldots, l, l+1, \ldots, j-1, j\}$. Set $B_1 = \{i, i+1, \ldots, l\}$ and $B_2 = \{l, l+1, \ldots, j\}$. Then:*

$$\begin{aligned}
u_B &= u_{B_1} u_{B_2} \\
d_B &= d_{B_2} d_{B_1}.
\end{aligned}$$

Proof. This is follows immediately from the definitions of u_B and d_B. □

Proposition 64. *Let $B = \{i, i+1, \ldots, j-1, j\}$ and $A = \{p, p+1, \ldots, q-1, q\}$. Then*

we have the following:

$$u_B d_A = \begin{cases} 0 & j = q \\ d_A u_{B^+_-} & \text{if } B \subset A \text{ and } j \neq q \\ d_{A^+_-} u_B & \text{if } A \subset B \text{ and } j \neq q \\ d_{A_-} u_{B^+} & \text{if } A \cap B = \emptyset, i = p-1, j \neq q+1 \\ d_{A^+} u_{B_-} & \text{if } A \cap B = \emptyset, i \neq p-1, j = q+1 \\ d_{A^+_-} u_{B^+_-} & \text{if } A \cap B = \emptyset, i = p-1, j = q+1 \\ d_A u_B & \text{if } A \cap B = \emptyset, i \neq p-1, j \neq q+1 \end{cases} \tag{1}$$

Proof. These all follow from straight-forward computations and the extended braid relation, Lemma 5.

These computations are nearly identical to the computation in the proof of Corollary 60, and are thus omitted here. □

Proposition 65. *Let $B, A \subsetneq I$, with both B and A connected. Then there exist connected sets $B', A' \subsetneq I$ with $|A' + B'| = |A + B|$ such that*

$$u_B d_A = d_{A'} u_{B'}.$$

Furthermore, the pair (A', B') is one of $(A, B), (A, B^+_-), (A^+_-, B), (A^+, B_-)$, or (A_-, B^+).

Proof. One may use Lemma 63 and Proposition 64 to derive arbitrary products $u_B d_A$ by taking $B = (B \cap A) \cup (B \setminus (B \cap A))$. Then the proof comes down to checking six additional cases, which all work out. These additional cases are the 'overlapping' cases where $A \cap B \neq \emptyset$, but B not contained in A and vice versa. □

In particular, consider the product w for $A = \{0, 1, \ldots, 1\}$ and B_i connected with $|B_i| > |B_{i+1}|$ for each i given by:

$$w := u_{B_l} u_{B_{l-1}} \ldots u_{B_1} d_A = d_{A'} u_{B'_l} u_{B'_{l-1}} \ldots u_{B'_1}$$

Then w is 0-dominant only if $u_{B'_1} u_{B'_2} \ldots u_{B'_l}$ is 0-dominant.

5 The k-Littlewood-Richardson Rule for Split k-Schur Functions

Our goal in this section is to prove a special case of the Littlewood-Richardson rule for k-Schur functions, as described in the introduction. The proof will rely heavily on the maximal decomposition of affine permutations as well as multiplication of cyclically increasing and decreasing elements.

First, we reformulate the splitting condition for cores in terms of the sizes of rows and columns of the associated bounded partitions.

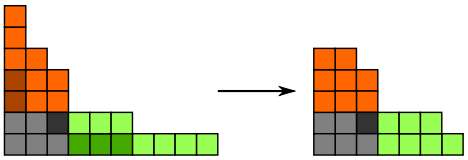


Figure 10: A split 5-core. The core splits into the orange core $\mathfrak{c}(\mu)$ and the green core $\mathfrak{c}(\nu)$. The 5-boundary is given by the lighter-colored boxes. Down-justifying the orange core gives a 4-column bounded partition, and left-justifying the green partition gives a 4-bounded partition. The sum of any light orange column and light green row is $\geqslant 5$, because the dark grey box has hook > 5.

Lemma 66. *Let λ be a k-bounded partition whose associated $k+1$-core $\mathfrak{c}(\lambda)$ splits into $k+1$-cores $\mathfrak{c}(\mu)$ and $\mathfrak{c}(\nu)$. Then for any i, j, we have $\mu_i^{(k)} + \nu_j \geqslant k+1$.*

Proof. Suppose $\mu^{(k)}$ has m parts and ν has n parts. We show that $\mu_m^{(k)} + \nu_n \geqslant k+1$; the statement then holds for arbitrary i, j since $\mu^{(k)}$ and ν are partitions, so that:

$$\mu_i^{(k)} + \nu_j \geqslant \mu_m^{(k)} + \nu_n \geqslant k+1.$$

Diagonally stacking the cores $\mathfrak{c}(\mu)$ and $\mathfrak{c}(\nu)$ yields the core $\mathfrak{c}(\lambda)$. By pushing the k-boundary of $\mathfrak{c}(\mu)$ down, we obtain the k-column bounded partition whose transpose is the k-bounded partition $\mu^{(k)}$. Pushing the k-boundary of $\mathfrak{c}(\nu)$ to the left, we obtain ν. (See Figure 10 for an example.) All of the boxes in the last column of $\mathfrak{c}(\mu)$ have hook $\leqslant k$, and are thus in the boundary $\partial_k(\mathfrak{c}(\mu))$. Likewise for the boxes in the top row of $\mathfrak{c}(\nu)$. But $\mathfrak{c}(\lambda)$ splits at a box with hook $\geqslant k+1$, so we have $\mu_m^{(k)} + \nu_n \geqslant k+1$. □

Suppose λ splits into factors μ and ν. We will express summands in $s_\mu^{(k)}$ as products of cyclically increasing elements and summands in $s_\nu^{(k)}$ as a product of cyclically decreasing elements. To find k-Littlewood-Richardson coefficients, we need to identify 0-dominant terms in the product $s_\mu^{(k)} s_\nu^{(k)}$. In any product wv for $w, v \in \hat{\mathbb{A}}$, we have $D_R(v) \subset D_R(wv)$. Thus, if v is not 0-dominant then the product wv cannot be 0-dominant. Since $s_\nu^{(k)}$ has a unique 0-dominant summand u_ν, we consider products $u_{\vec{A}} u_\nu$, where $u_{\vec{A}}$ appears in $s_\mu^{(k)}$. We then need to answer two questions:

- For which $\vec{A}$ is the product $u_{\vec{A}} u_\nu$ 0-dominant?
- Which of these $u_{\vec{A}}$ appear as summands in $s_\mu^{(k)}$?

Definition 67. Let $x, y \in \hat{\mathbb{A}}_k$. Then x is *left-compatible* with y, which we denote $x \vdash y$, if $xy \neq 0$ and $D_R(xy) = D_R(y)$.

Then, when $\nu \neq \emptyset$, the k-Littlewood-Richardson coefficients may be expressed as:

$$c_{\mu,\nu}^{\lambda} = \sum [x] s_\mu^{(k)},$$

where the sum is over $x \vdash u_\nu$ such that $x u_\nu = u_\lambda$.

Lemma 68. *Suppose $u_{\vec{B}} = u_{B_m} \cdots u_{B_1}$ is a maximal cyclically increasing product of shape $\lambda = (|B_1|, \ldots, |B_m|)$ and $d_{\vec{A}} = d_{A_n} \cdots d_{A_1}$ satisfies $|A_j| + |B_i| \geqslant k+1$ for all i, j. Then $u_{\vec{B}} d_{\vec{A}}$ is i-dominant if and only if $d_{\vec{A}}$ is i-dominant and $u_{\vec{B}}$ is $(i-n)$-dominant.*

In this case, we also have:

$$u_{B_m} \cdots u_{B_1} d_{\vec{A}} = d_{\vec{A}+m} u_{B_m+1} \cdots u_{B_1+1}.$$

Proof. For the reverse direction, let $\lambda = \mathrm{sh}(\vec{B})$ and $\mu = \mathrm{sh}(\vec{A})$ be the bounded partitions associated to the elements $u_{\vec{B}}$ and $d_{\vec{A}}$. Then diagonally stacking the $k+1$-cores $\mathfrak{c}(\lambda^{(k)})$ and $\mathfrak{c}(\mu)$ yields a $k+1$-core ν that splits into $\lambda^{(k)}$ and μ. The 0-dominant element associated to ν is equal to $\Psi^{(-i)}(u_{\vec{B}} d_{\vec{A}})$, and so we see that the product is i-dominant.

The forward direction is more complicated. We see immediately that $d_{\vec{A}}$ must be i-dominant in order for the product to be i-dominant. Thus, we induct on m, the number of parts of $\vec{B}$.

For the base case, $\vec{B} = B_1$, and our assumption is $u_{B_1} \vdash d_{\vec{A}}$. Since $d_{\vec{A}}$ is i-dominant, we have A_n connected, so let $A_n = [i-n+1, j]$. (The features of the base case are illustrated in Figure 11.)

Let $L = [p, q]$ be any connected component of B_1 with $D_R(L) \neq \{i-n\}$. (We will argue by contradiction to show that no such L can exist.) We observe that $D_R(u_{B_1}) \cap \{i-n+1, \ldots, j\} = \emptyset$, or else the product is zero or non-dominant. Thus, $q \notin \{i-n+1, \ldots, j\}$, so that $q \in [j+1, i-n-1]$.

In fact, we claim that $L \cap \{i-n, \ldots, j\} = \emptyset$. Otherwise, L must contain j, as u_L is an increasing product. If $j \in L$, set $d_{\vec{A}'}$ be the maximal decomposition of $a_{j+1} \cdots a_q d_{\vec{A}}$. Since $[j+1, q] \cap \{i-n, \ldots, j\} = \emptyset$, $\vec{A}'$ has the same number of parts as $\vec{A}$. Then we consider $a_j d_{\vec{A}'}$. If $j+1 \in A'_m$, multiplication by a_j starts a new row in the k-code $\mathrm{RD}(d_{\vec{A}'})$, which creates a new right descent. If $j+1 \notin A'_m$, then $a_j d_{\vec{A}'} = 0$, since $j \in D_R(d_{A_m})$. Thus, $j \notin L$, and so $L \cap \{i-n, \ldots, j\} = \emptyset$.

If there is no connected component of B_1 with right descent $i-n$, we then have $B_1 \subset I \setminus \{i-n, \ldots, j\}$. But there are $|A_n|+1$ elements in $\{i-n, \ldots, j\}$, so that $|B_1| \leqslant k+1-(|A_n|+1)$, so that $|B_1| + |A_n| \leqslant k$, contradicting the assumption that $|B_1| + |A_n| \geqslant k+1$.

On the other hand, suppose B_1 has a connected component C with right descent $i-n$. Then $C = \{r, \ldots i-n-1, i-n\}$. If there are other connected components, we know that they are contained in $[j+1, i-n-1]$. Since C contains $i-n$, no other connected component may contain any elements in $[r-1, i-n+1]$, or else that component would be connected to C. As a result, if $C \cap [i-n+1, j] \neq \emptyset$ there can be no other connected components. Thus, all other components are subsets of $[j+1, r-2]$. But if there are other connected components, we then have $|C \cup [j+1, r-2] \cup A_n| = k$, so that $|B| + |A_n| < k+1$, contrary to assumption.

Thus, there are no connected components of B_1 with right descent other than $(i-n)$. As a result, B_1 is $(i-n)$-dominant, as desired. (For an example, see Figure 11.)

For the inductive step, we assume that $x = u_{B_{m-1}} \cdots u_{B_1}$ is $(i-m)$-dominant and $u_{\vec{B}}$ maximal. Set $\vec{B}^- = \{B_1, \ldots, B_{m-1}\}$. We consider $D_R(u_{B_m})$. Maximality of $\vec{B}$ implies that each B_n is a subset of $B_{n-1}+1$.

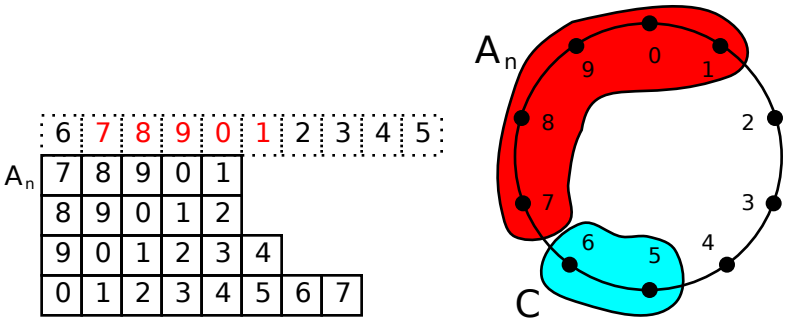


Figure 11: An example with $k = 9$ for the base case of Lemma 68. On the left is the k-code is for a 0-dominant element $d_{\vec{A}}$ of length 24; we consider a 0-dominant product $u_B d_{\vec{A}}$ where $|B| \geqslant 5$. Left-multiplying by any of the generators in $A_n = \{a_7, a_8, a_9, a_0, a_1\}$ will either create a descent or kill the element. Thus, elements in A_n cannot be right descents of u_B, and by the argument in the proof, cannot be in any connected component of B whose right descent is not 6.
On the right is the Dynkin diagram with $k = 9$. The set A_n is highlighted in red, and an increasing subset C with right descent 6 is highlighted in cyan. The proof for the base case argues that any other connected component of B would then have to be contained in the set $\{2, 3\}$, but then $|B| + |A_n| \leqslant 9$, contradicting the assumption that $|B| + |A_n| \geqslant 10 = k + 1$. We may then observe that B must be connected with right descent 6.

We observe that if u_{B_m} is not connected with right descent $i - n + m$, there exists an alternate factorization $x = u_{\vec{B'}}$ with $B'_1 = \{j - (m-1) \mid j \in B_m\}$. This occurs because each connected component of B_m contributes a right descent to $u_{\vec{B}}$. Factoring out this right descent on the right leaves the next element in the connected component, and so on. This B'_1 must have $u_{B'_1} d_{\vec{A}}$ 0-dominant, or else $u_{\vec{B}} d_{\vec{A}}$ will not be zero-dominant. Then by the base case, this B'_1 must be connected with right descent $i - n$. But this means that B_m was connected with right descent $i - n + m$.

□

Lemma 69. *Suppose a k-bounded partition λ splits into two components, μ and ν. Then $s_\mu^{(k)} s_\nu^{(k)} = s_\lambda^{(k)}$.*

Proof. We induct on the number of parts in μ, considering elements of $\hat{\mathbb{A}}_k$ appearing with non-zero coefficient in $s_\mu^{(k)}$ as products of cyclically increasing elements, and those appearing in $s_\nu^{(k)}$ as decreasing elements. Then any x with $[x]s_\mu^{(k)} \neq 0$ has a cyclically increasing expansion $x = u_{B_m} \cdots u_{B_1}$ where $|B_i| = \mu_i^{(k)}$.

Let a_ν denote the unique 0-dominant term in $s_\nu^{(k)}$, with maximal cyclically decreasing product $a_\nu = d_{A_n} \cdots d_{A_1}$. Then we claim that there exists a unique summand x in $s_\mu^{(k)}$ such that $x \vdash a_\nu$, and furthermore that x is $(k + 1 - n)$-dominant. By the splitting condition we have $|B_i| + |A_j| \geqslant k+1$ for all i, j. To do this, we will induct on the number of parts of $\mu^{(k)}$.

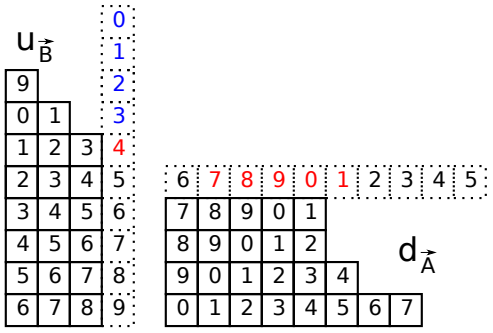


Figure 12: An example with $k = 9$ for the induction step of Lemma 68. The k-code on the left represents the element $u_{\vec{B}}$ and the k-code on the right is for an element $d_{\vec{A}}$; consider a 0-dominant product $u_B u_{\vec{B}} d_{\vec{A}}$ where $|B| \geqslant 5$. Left-multiplying d_A by any of the generators in $S = \{a_7, a_8, a_9, a_0, a_1\}$ will either create a descent or kill the element. B cannot contain any of the blue residues or else maximality of $u_B u_{\vec{B}}$ will be violated. If a_5 is a right descent of u_B then a_2 will be a right descent of $u_B u_{\vec{B}}$, and is thus disallowed (and colored red in the diagram). If B_m is disconnected, (for example, set $B = \{4, 5, 6, 8, 9\}$), we can re-express the element as $u_{\vec{B}} = x u_{1,2,3,5,6}$. But this violates the base case. Then since $|B| \geqslant 5$, B must be connected with $D_R(u_B) = \{9\}$.

For the base case, $\mu^{(k)}$ has a single part, so that $s^{(k)}_{(l)} = e_l = \sum_{|A|=l} u_A$. In this case, every summand is maximal. Then by Lemma 68, if $x \vdash a_\nu$, then x is $(k+1-n)$-dominant. There is a unique such element in $s^{(k)}_{(l)}$, so the product $s^{(k)}_{(l)} s^{(k)}_{\lambda}$ has a single 0-dominant summand, as desired.

For the induction step, we suppose the statement holds for any k-bounded partition ρ with $\leqslant m$ parts. Let $\mu^{(k)} = \rho \cup (l)$ be a k-bounded partition with $m+1$ parts. We consider the product:

$$e_l s^{(k)}_{\rho} = s^{(k)}_{\mu} + \sum_{\kappa} s^{(k)}_{\kappa},$$

according to the Pieri rule. We recall that each $\kappa \succ \mu$. Finally, let a_ρ be the unique $(k+1-n)$-dominant summand in $s^{(k)}_{\rho}$.

Claim 1: For any x a summand in $e_l s^{(k)}_{\rho}$, if $x \vdash a_\nu$ then $x = u_C a_\rho$ with $|C| = l$.

We are interested in 0-dominant elements in the product $e_l s^{(k)}_{\rho} s^{(k)}_{\nu}$. Recall that for any elements p, q in a Coxeter group, we have $D_R(q) \subset D_R(pq)$. Then any 0-dominant term in $s^{(k)}_{\rho} s^{(k)}_{\nu}$ is of the form $p a_\nu$ for some p. But by the inductive hypothesis, we see that the only non-zero summand p in $s^{(k)}_{\rho}$ with $p \vdash a_\nu$ is a_ρ. The claim then follows immediately.

Claim 2: For any $x \vdash a_\nu$, we have

$$[x] e_l s^{(k)}_{\rho} = 1.$$

By first claim, we have $x = u_C a_\rho$. If the coefficient were greater than 1, we would have a second decomposition $u_D a_\rho$. But then $u_C = u_D$.

Claim 3: Let $x = u_C a_\rho \vdash a_\nu$, and $a_\nu := u_{\vec{B}}$. Then if $u_C u_{\vec{B}}$ is not maximal, then $[x]s_\mu^{(k)} = 0$.

By the results of Section 3, x has a unique maximal decomposition $x = u_{C'} u_{\vec{B}'}$, where $\vec{B}'$ has the same number of parts as $\vec{B}$ and $|C'| \leqslant |C|$. By the inductive hypothesis, $u_{\vec{B}'}$ is then $(k+1-n)$-dominant; let it be of shape γ. By Proposition 56, we have $\rho \subset \gamma$. Set $\gamma^+ = \gamma \cup (|C'|)$, which is $\mathrm{sh}(x)$.

By Theorem 57, $\gamma^+ \setminus \rho$ is a weak strip, so γ^+ appears in the Pieri rule expansion of $e_l s_\rho^{(k)}$. Furthermore, $[x]s_{\gamma^+}^{(k)} = 1$, by Theorem 40. Then we observe that:

$$\begin{aligned} 1 &= [x]e_l s_\rho^{(k)} \\ &= [x](s_\mu^{(k)} + s_{\kappa^+}^{(k)} + \sum_\kappa s_\kappa^{(k)}) \\ &= [x]s_\mu^{(k)} + [x]s_{\kappa^+}^{(k)} + \sum_\kappa [x]s_\kappa^{(k)}. \end{aligned}$$

All of these coefficients are $\geqslant 0$, and $[x]s_{\kappa^+}^{(k)} = 1$, so $[x]s_\mu^{(k)} = 0$.

Thus, when $x \vdash a_\nu$ and $[x]s_\mu^{(k)} > 0$, we have $x = u_C u_{\vec{B}}$ is maximal. Then by Lemma 68, x is $(k+1-n)$-dominant. There is a unique such element in $s_\mu^{(k)}$, which completes the proof. □

Theorem 70. *Suppose λ splits into components μ_i. Then*

$$s_\lambda^{(k)} = \prod s_{\mu_i}^{(k)}.$$

Proof. This follows from successive application of Lemma 69. □

References

[BB93] Nantel Bergeron and Sara Billey, *Rc-graphs and schubert polynomials*, Experiment. Math. **2** (1993), no. 4, 257–269.

[BB05] Anders Björner and Francesco Brenti, *Combinatorics of Coxeter groups*, Graduate Texts in Mathematics, vol. 231, Springer, New York, 2005. MR MR2133266 (2006d:05001)

[BBTZ11] Chris Berg, Nantel Bergeron, Hugh Thomas, and Mike Zabrocki, *Expansion of k-schur functions for maximal k-rectangles within the affine nilcoxeter algebra*, Preprint: http://arxiv.org/abs/1107.3610.

[FG98] Sergey Fomin and Curtis Greene, *Noncommutative schur functions and their applications*, Discrete Mathematics **193** (1998), no. 13, 179 – 200.

[Hum90] James E. Humphreys, *Reflection groups and Coxeter groups*, Cambridge Studies in Advanced Mathematics, vol. 29, Cambridge University Press, Cambridge, 1990. MR MR1066460 (92h:20002)

[Lam06] Thomas Lam, *Affine stanley symmetric functions*, Amer. J. Math **128** (2006), 1553–1586.

[LLM03] Luc Lapointe, Alain Lascoux, and Jennifer Morse, *Tableau atoms and a new macdonald positivity conjecture*, Duke Math. J. **116** (2003), 103–146.

[LLM+12] Thomas Lam, Luc Lapointe, Jennifer Morse, Anne Schilling Mark Shimozono, and Mike Zabrocki, *A primer on k-schur functions*, 2012, In preparation.

[LM03] Luc Lapointe and Jennifer Morse, *Tableaux on $k+1$-cores, reduced words for affine permutations, and k-schur expansions*, J. Combinatorial Theory, Ser. A **112** (2003), 44–81.

[LM07] ———, *A k-tableau characterization of k-schur functions*, Adv. Math. **213** (2007), 183–204.

[Lus83] G. Lusztig, *Some examples of square integrable representations of semisimple p-adic groups*, Trans. Amer. Math. Soc. (1983), 623–653.

[S+09] W. A. Stein et al., *Sage Mathematics Software (Version 3.3)*, The Sage Development Team, 2009, http://www.sagemath.org.

[SCc09] The Sage-Combinat community, *Sage-Combinat: enhancing sage as a toolbox for computer exploration in algebraic combinatorics*, 2009, http://combinat.sagemath.org.